Direct Integration of Collaboration into Broadcast Content Formats

DISSERTATION
zur Erlangung des akademischen Grades
Doktorin der Technischen Wissenschaften
im Doktoratsstudium der
Technischen Wissenschaften

Eingereicht von:
Dipl.-Ing.\textsuperscript{in} Sabine Lösch

Angefertigt am:
Institut für Telekooperation

Beurteilung:
Univ.Prof.\textsuperscript{in} Dr.\textsuperscript{in} Gabriele Anderst-Kotsis
Associate Professor Dr. Artur Lugmayr

Linz, Oktober 2015
To David, Eva and Jakob.
ACKNOWLEDGMENTS

The first and most sincere acknowledgment is to my husband David. He is my inspiration and my tower of strength. Without him I would not have been able to reach this point in my academic career.
The second acknowledgment is to my beloved kids Jakob and Eva, who always give me a break from research work back to everyday life and help me see things from a child’s perspective.

For substantial professional and mental support I would particularly like to thank my supervisor Professor Gabriele Anderst-Kotsis, who always believed in me and my work. She motivated and encouraged me, even after 3 years of parental leave, to finish this thesis.

The same is of course true for Professor Artur Lugmayr for his comprehensive and most helpful feedback and for having endless and productive discussions in Finland and in the Linz main railway station, which helped me gain a new perspective on topics I was stuck on.
Particularly important was my mentor Professor Susanne Saminger-Platz. She pushed me to keep trying, and aside from academic topics even helped me to balance two toddlers and a PhD.

Furthermore I owe special thanks to my colleagues Professor Wieland Schwinger and Professor Ismail Khalil, and all other colleagues of the Department of Telecooperation whom I had the privilege for cooperating with during the past couple years. I also owe special thanks to our secretary Angela Kohl, who always had a sympathetic ear and was supportive in non-academic issues.
Finally I would like to thank my colleague Dr. Pauliina Tuomi for her straightforward cooperation in 2011 / 2012.

Besides professional support, my parents deserve thanks and honor for supporting me throughout my entire academic career. Without their support, this would barely have been possible. Other very important sources of mental support were my sister Elke, my brother-in-law Clemens, and my beloved godson Thomas who can always make me laugh and take my mind of things.
ABSTRACT

During the past decade broadcasting technology shifted from analog towards digital. The new digital technology enabled new possibilities for collaborative program formats and interactivity. The shift to digital broadcast technology has increased the consumer’s interest in more interactive applications beyond simple teletext solutions. These new technologies invited the viewer to play a more active or participatory role in TV. Simple examples are i.e. discussing broadcast content in forums, gaining additional information about the program content on the Internet, or working collaboratively with other content consumers i.e. via the implicit exchange of consumer behavioral patterns to personalize program content (e.g. Netflix). Broadcasters followed the consumer trends, and started to offer more and more solutions on program format level to offer a more participatory, inviting, and collaborative television experience. Program formats that enabled an increased participation gained popularity, and range far beyond reality, casting, or voting program formats.

Today two different major developments are noticeable: 1) research and development focused mainly on the development of proprietary television platforms that provide social interaction and collaboration between views; 2) industry on the other hand focused on the development of participatory broadcast content formats deployed on television platforms accompanying the main TV platform such as e.g. second screen principles or web based media forms to invite the audience to engage with content. Despite experimental participatory program formats, transmedia television content, and the new trends on Internet or IPTV platforms, participation in program content, engaging with audiences, and collaboration seems to have been a minor trend. In conclusion, today’s TV platforms provide mostly only simple collaborative services attached to broadcast content, which are rather designed to view TV content passively than to actually participate in program formats. This poses a problem and prevents the development of more complex collaborative and social interaction, as on the one hand synchronization is lost, and on the other hand media disruption happens when the audience has to shift to parallel platforms.

This thesis works exactly challenges this fact, and attempts to develop participatory program formats and platforms supporting collaborative and social activity. Linking content formats, platforms, and audience participation, and which technological solutions exist to create this link is major concern within the scope of this thesis. Thus this thesis investigated and provided solutions which 1) enable the development of more complex collaborative and social interaction on the platform TV; 2) synchronization models for content presentation, interaction, collaboration, secondary and tertiary screens, and streaming technologies; 3) includes metadata definitions to encode meta-information about the content structure, it’s collaboration capabilities, and exchange format between consumer, broadcasters and other 3rd parties; 4) defines a Collaborative Broadcast Content Format (CBF) to enable this type of collaboration; 5) defines a generic reference architecture to integrate the CBF into a TV stream; 6) integrated the proposed reference architecture into the development of applications and services, including an editing environment to create collaborative content
and its impact on the narrative; and 7) validates the applicability of the implementations with self-developed benchmarks. In short, the thesis focuses on bridging the gap between the wide variety of broadcast program formats, content formats, and collaborative activity by providing the required technical solutions.

To link those, a metadata format is developed to be included into the broadcast content format and processed on the consumer’s side. The metadata trigger the collaborative activity and act as a coordinating unit between the collaborators. The thesis aimed at the provision of a generic reference architecture to avoid proprietary solutions, and therefore adopted solutions based on existing standardized solutions as proposed by DVB or MPEG. The developed generic reference architecture shall act as framework for the direct integration of collaborative activities into existing broadcast content, technologies, and formats. The architecture identifies and defines the components, interfaces, standards, technologies, design patterns to support the implementation of a collaborative broadcast system. The reference architecture includes a professional broadcast environment, including the needs of content producers as well as consumers. One special feature of the architecture is the tight alignment with the business lifecycle of collaborative broadcast content formats, and integrates related needs and requirements.

Besides the implementation of the generic collaborative broadcast architecture, several applications and services have been developed to demonstrate technical feasibility. The application “CBF Metadata Editor” illustrates how program content tagged with metadata can be created. By utilizing this tool, two broadcast services - namely “Collaborative Voting App” and “Collaborative Gaming App” - have been developed to tie broadcast content with collaborative activity. To demonstrate the practical applicability of the research, the collaborative broadcast format “Against the Others - Interactive” illustrated how audiences can be ‘invited’ to become an interactive and collaborative part of a broadcast show. Several implementations have been tested against a developed benchmark, which has been developed within the scope of the thesis work. The benchmark included criteria such as resource utilization, reliability, performance, real-time metrics, and time requirements. The benchmark allowed to validate the proposed architecture including its applications, and tested the suitability of the application of metadata as solution to tie broadcast content and collaborative activity in real-time critical broadcasting systems. A feature based evaluation allowed the comparison of the proposed concepts to other market available solutions, or research results.

To conclude, this thesis demonstrated the feasibility, as well as it proposed a technical solution to tie broadcast platforms, content, and collaborative activity with the proposed reference architecture together. It clearly demonstrated, that the development of a technical solution fulfilling several broadcast related requirements is possible and feasible. It also demonstrated the real-time deployment of such services on practical examples and applications. As the broadcasting sector currently still undergoes a re-orientation phase in times of IPTV, Internet TV, and other distribution format, the results of the thesis enrich the possible future pathway of the medium TV by other forms of broadcast formats. If these will succeed, will remain with the consumer. Nevertheless, the thesis contributed to the scientific and research community with several publications, as well as the reference architecture is available for further developments.
KURZFASSUNG


Es sind zwei Hauptentwicklungen zu beobachten: 1) die Wissenschaft fokussiert hauptsächlich die Entwicklung proprietärer Plattformen, welche zusätzlich zum Fernsehen interaktive, soziale und kollaborative Anwendungen anbieten. 2) die Industrie hingegen entwickelte partizipative TV Formate begleitet von parallelen Plattformen wie Second Screen Lösungen oder Web basierten Medien Formaten, um Zuseher zur Interaktion einzuladen. Zusammenfassend ist festzustellen, dass partizipative TV Formate und Plattformen, welche soziale und kollaborative Aktivitäten neben den TV Inhalten anbieten, existieren, jedoch sind sie in den meisten Fällen nicht verknüpft. Diese Trennung schafft Probleme und verhindert das Angebot von komplexerer kollaborativer und sozialer Interaktion da die notwendige Synchronisation sowie der gemeinsame Fokus (welcher der ausgestrahlte TV Inhalt ist) verloren gehen. Synchronisation und Fokus sind aber für kollaborative und teilweise auch für die soziale Interaktion von wesentlicher Bedeutung.

Brückenschlag zwischen der Vielzahl an TV Inhalten und Plattformen zur Kollaboration und sozialer Interaktion durch die Entwicklung technischer Lösungen.


Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die vorliegende Dissertation ist mit dem elektronisch übermittelten Texdokument identisch.

Linz, October 2015

______________________________
Sabine Lösch
# CONTENTS

## 1 FUNDAMENTALS

1 INTRODUCTION  
1.1 Motivation and Research Challenges  
1.2 Research Questions and Goals  
1.3 Hypotheses  
1.4 Contribution  

## 2 STATE-OF-THE-ART WORK OF DIRECT INTEGRATION OF COLLABORATION INTO BROADCAST CONTENT FORMATS  
2.1 Methodology for Reviewing and Classifying State-of-the-Art Work  
2.1.1 Selection of Approaches  
2.1.2 Conveyance of Uncatalogued Criteria  
2.1.3 Excluding Non-Relevant Criteria  
2.1.4 Establishing a Schema for Categories and Criteria Definition  
2.2 Categorization of Consumer Collaboration in Broadcast Content Formats  
2.2.1 Narrative Perspective on Broadcast Content  
2.2.2 CMC Link of Collaborative Activity to Broadcast Technology and Program Contents & Formats  
2.2.3 Collaborative Activity  
2.2.4 Technology Features  
2.3 Classification of Collaborative Program Contents & Formats  
2.3.1 Linear Narrative Space  
2.3.2 Branched Narrative Space  
2.4 Conclusions & Implications  
2.4.1 General Observations  
2.4.2 The Content Narrative Space  
2.4.3 CMC Link of Collaborative Activity to Program Contents & Formats  
2.4.4 Scope of Collaboration  
2.4.5 Key-Implications for the Reference Architecture  

## 3 USE CASE SCENARIOS AND REQUIREMENTS DEFINITION  
3.1 Target Group Description  
3.1.1 Evaluation of the Target Audience  
3.1.2 Content Producer Description  
3.2 Consumer Story Scenarios  
3.2.1 Non-Narrative Modification Story Scenarios  
3.2.2 Narrative Modification Story Scenarios  
3.2.3 Classification of Consumer Scenarios  
3.3 Content Production & Broadcast Station Scenarios  
3.3.1 Non-Narrative Modification Scenarios  
3.3.2 Narrative Modification Scenarios  
3.3.3 Classification of Content Production & Broadcast Station Scenarios  
3.4 Requirements Definition
3.4.1 Broadcast Technology as Medium ..................................... 54
3.4.2 Broadcast Content Formats ............................................. 57
3.4.3 Collaborative Activity and Consumer ............................... 58
3.4.4 Linking Collaborative Activity to Broadcast Technology as Medium and Program Content & Formats .......................... 58
3.5 Technical Requirements .................................................. 59
     3.5.1 Consumer Client Module .......................................... 59
     3.5.2 Broadcaster Server Module ........................................ 62
     3.5.3 Broadcaster/Producer Module to Prepare and Enhance Medium and Content Before Broadcast .......................... 63

ii INTEGRATION OF COLLABORATION INTO BROADCAST CONTENT FORMATS 65
4 REFERENCE ARCHITECTURE ............................................. 66
     4.1 Reference Architectures ............................................... 66
     4.2 A Reference Architecture for the Direct Integration ............ 66
         4.2.1 Strategic Purpose ............................................... 68
         4.2.2 System Architecture .......................................... 71
         4.2.3 Standards ....................................................... 76
         4.2.4 Design Patterns ............................................... 76
         4.2.5 Business Value Chain - Lifecycle of Collaborative Broadcast Formats 81
5 IMPLEMENTATION OF COLLABORATION IN BROADCAST CONTENT FORMATS 83
     5.1 Overview & Architectures ............................................. 83
         5.1.1 Architecture .................................................... 84
         5.1.2 Implementation Requirements ................................. 85
     5.2 Implementation of the Reference Architecture .................... 91
         5.2.1 The Broadcaster/Producer Module ............................. 92
         5.2.2 Implementation of the Broadcaster Module (Server) ........ 100
         5.2.3 Implementation of the Consumer Module (Client) .......... 100
     5.3 Error Protection .................................................... 102
         5.3.1 Impact of Erroneous Metadata ................................. 103
         5.3.2 Levels of Criticality .......................................... 103
         5.3.3 Types of Errors ............................................... 104
         5.3.4 Actions to be taken for the different Levels of Criticality ..... 106
     5.4 Scenarios .......................................................... 107
         5.4.1 Voting (“CBF Voting App”) .................................... 107
         5.4.2 Quiz (“CBF Gaming App”) ..................................... 107
     5.5 Procedure Model of the Implementation of a CBF Application Scenario ............................................. 109
         5.5.1 Conception Phase .............................................. 109
         5.5.2 Implementation Phase ........................................ 110
         5.5.3 Merging Phase .................................................. 113
6 PROOF-OF-CONCEPT IMPLEMENTATIONS AND PERFORMANCE EVALUATION 114
     6.1 Case Study-based Evaluation ....................................... 114
         6.1.1 Scenario 1 - Vote for the Top 333 Music Acts .............. 115
         6.1.2 Scenario 2 - Game/Quiz Show ................................. 123
         6.1.3 Result of the Case Study based Evaluation ................. 140
     6.2 Feature-based Evaluation ............................................ 141
         6.2.1 Narration Space ................................................ 141
6.2.2 CMC Link ................................ 141
6.2.3 Collaboration .............................. 142
6.2.4 Others .................................. 142
6.3 Benchmark-based Evaluation ......................... 142
6.3.1 Performance Evaluation ........................ 143
6.3.2 Requirements Evaluation ....................... 154
6.3.3 Reliability ................................ 155

7 CONCLUSION AND OUTLOOK ................................. 157
7.1 Justification of the Hypotheses ........................ 157
7.2 Discussion .................................... 158
7.3 Key-Results and Contributions ........................ 161
7.4 Shortcomings and Prospects .......................... 163
  7.4.1 Including the HbbTV Standard ...................... 163
  7.4.2 Experimental Setup ................................ 163
  7.4.3 Define Further Metadata Patterns .................. 164
  7.4.4 Implement a Second Screen Scenario ................. 164
  7.4.5 Identify, Observe and Measure the Collaborative Activity .... 164
  7.4.6 Implement the Back Channel to Modify the Course and Characteristics of the Broadcast Content Format .... 164
7.5 Future Work ................................... 164

iii APPENDIX .................................................. 166
A COLLABORATIVE BROADCAST CONTENT FORMATS - GLOSSARY ................ 167
B APPROACHES ENABLING COLLABORATION IN BROADCAST CONTENT FORMATS ....... 170
  B.1 CMC Link of Collaborative Activity to the Broadcast Technology as Medium .... 170
  B.2 CMC Link of Collaborative Activity to the Program Contents & Formats .......... 173
C DESIGN PATTERN DESCRIPTION ........................................ 180
  C.1 Broadcast Medium and Program Content Support ............................ 181
    ........................ Access Medium ...................... 181
    ........................ Analyze Medium ............... 182
    ........................ Modify Medium ............. 182
    ........................ Enhance Medium .......... 183
    ........................ Tags to Anchor the Triggers to the Broadcast Medium and Content 183
  C.2 Collaboration - Application Management ............................. 185
    ........................ Session Management .... 185
    ........................ Shared Object Management ... 187
    ........................ Data Consistency Management 191
  C.3 Collaboration - Community Support ............................. 192
    ........................ Community Management Activities 192
    ........................ Security & Privacy ........ 195
  C.4 Collaboration - Group Support ............................. 196
    ........................ Group Management Activities 197
    ........................ Synchronous Group & Inter-group Awareness 200
    ........................ Asynchronous Group & Inter-group Awareness 202
    ........................ Communication .......... 202
    ........................ Collaborative Artifact Management 203
  C.5 Collaboration - Collaboration Support ............................. 204
C.5.1 Collaboration Provision ............................................. 205
C.5.2 Collaboration Analytics ............................................. 205
C.5.3 Collaboration Interaction Observation ......................... 207
C.6 CMC Link .......................................................... 209
    C.6.1 XML Trigger .................................................. 209

D Benchmark Based Evaluation - Metadata and Results ............ 213
    D.1 Metadata ....................................................... 213
    D.2 Metadata Extraction Results .................................... 217

Bibliography ........................................................................ 222
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>TV Content Production Value Chain from Content Creation to Consumption</td>
<td>2</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Extended TV Content Production Value Chain</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Classification of the Identified Research Questions</td>
<td>7</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Technological Outcome of this Thesis</td>
<td>9</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Overview of the Contributions</td>
<td>10</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Overview of the Research Work Resulting in this Thesis</td>
<td>11</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Catalogue of Categories and Criteria to Classify Collaboration in Broadcast Content Formats</td>
<td>16</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Geometric Design Structures for the Content Narrative Space</td>
<td>18</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Overview of the Classification of Approaches in Collaboration in Broadcast Content Formats</td>
<td>25</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Usage of Information and Communication Technology in Austrian Households</td>
<td>34</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Process Illustration of Synchronized Non-Modification Collaboration</td>
<td>43</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Example Trigger Embedded in the Broadcast Medium</td>
<td>43</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Example Trigger Embedded in the Broadcast Content</td>
<td>45</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Process Illustration of Synchronized Modification Collaboration</td>
<td>46</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Example Trigger Embedded in the Broadcast Medium and/or Content</td>
<td>47</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Integrate Collaboration Into the Broadcast Medium and/or Content</td>
<td>51</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Modify Broadcast Medium and/or Content</td>
<td>52</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Reception Allocation of Austrian Households</td>
<td>54</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Technological Outcome of this Thesis</td>
<td>59</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Surveying Model of the Consumer Client Module</td>
<td>60</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Surveying Model of the Broadcaster Server Module</td>
<td>62</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Surveying Model of the Broadcaster/Producer Module</td>
<td>63</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Abstract Representation of a Reference Architecture</td>
<td>67</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Abstract Representation of the CBF Reference Architecture</td>
<td>67</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Component Architecture of the CBF Reference Architecture</td>
<td>69</td>
</tr>
<tr>
<td>Figure 26</td>
<td>System Architecture and Workflow of the CBF Reference Architecture</td>
<td>71</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Initial Components</td>
<td>72</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Use Case Diagram of the Broadcaster/Producer Module</td>
<td>72</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Broadcaster/Producer Module Activity Diagram</td>
<td>73</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Use Case Diagram of the Consumer Client Module</td>
<td>74</td>
</tr>
<tr>
<td>Figure 31</td>
<td>Consumer Client Module Activity Diagram</td>
<td>75</td>
</tr>
<tr>
<td>Figure 32</td>
<td>Use Case Diagram of the Broadcaster Server Module</td>
<td>75</td>
</tr>
<tr>
<td>Figure 33</td>
<td>Broadcaster Server Module Activity Diagram</td>
<td>75</td>
</tr>
<tr>
<td>Figure 34</td>
<td>Structure of the Design Patterns</td>
<td>78</td>
</tr>
<tr>
<td>Figure 35</td>
<td>Broadcast Content Value Chain for Collaborative Broadcast Formats</td>
<td>81</td>
</tr>
<tr>
<td>Figure 36</td>
<td>Technological Outcome of this Thesis</td>
<td>83</td>
</tr>
<tr>
<td>Figure 37</td>
<td>Block Diagram of the Reference Architecture</td>
<td>84</td>
</tr>
</tbody>
</table>
Figure 38 Timeline of the CBF Quiz .................................. 88
Figure 39 Matching of the XML Metadata to the Graphical User Interface of the “CBF Voting App” ........................................... 89
Figure 40 Component Diagram of the Prototypical Implementation of the Broadcaster/Producer Module .................................. 92
Figure 41 MPEG Transport Stream ..................................... 93
Figure 42 Project Folder Structure ..................................... 95
Figure 43 UML Activity Diagram of the Merging Algorithm ......................... 97
Figure 44 Order of Inclusion of Video and Data Frames with the Corresponding Timestamps ......................................................... 98
Figure 45 Screenshot of the Broadcaster/Producer Module ................. 99
Figure 46 Component Diagram of the Prototypical Implementation of the Consumer Client Module ............................................. 100
Figure 47 Types of Frame and Bit Errors .................................. 105
Figure 48 Screenshot of the “CBF Voting App” ................................ 108
Figure 49 Screenshot of the “CBF Gaming App” .............................. 108
Figure 50 Developing a CBF Application Scenario as Part of the Content Value Chain [52] .................................................. 109
Figure 51 Program Progress Including Metadata for a Voting Scenario ....... 121
Figure 52 Screenshot of the CBFPlayer Showing the Viewer Notification .... 122
Figure 53 Screenshot of the CBF Voting Application .......................... 122
Figure 54 Screenshot of the CBF Voting Application .......................... 122
Figure 55 Screenshot of the CBF Voting Application .......................... 123
Figure 56 Substantiation Process of Format Development .......................... 124
Figure 57 Layout of the Setting for “Against the Others - Interactive” ........ 132
Figure 58 Progress of “Against the Others - Interactive” Including the Position and Ids of the Metadata ............................................ 133
Figure 59 Screenshot of the CBFPlayer Showing the Notifier to the Gaming Application .................................................. 137
Figure 60 The Enabler Triggers Game #1 in the Gaming Application .......... 138
Figure 61 The Disabler Concludes and Terminates the Gaming Application .... 138
Figure 62 The Updater Refreshes the Main Area of the Gaming Application Showing the Gaming Rules .............................................. 138
Figure 63 The Coordinator Activates the Input Functionality for Peer 2 ........ 139
Figure 64 The Shortened Registration Process .................................. 139
Figure 65 Timeline of the Shortened Quiz Scenario .............................. 143
Figure 66 Dreambox 7080HD$^2$ ............................................. 150
Figure 67 CPU and Memory Usage of the CBFPlayer when Running the Test Sequence without Metadata Included .............................. 152
Figure 68 CPU and Memory Usage of the VLC Media Player when Running the Test Sequence without Metadata Included .............................. 152
Figure 69 CPU and Memory Usage when Running the Test Sequence on Experiment #1 .............................................. 153
Figure 70 CPU and Memory Usage when Running the Test Sequence on Experiment #2 .............................................. 153
Figure 71 Four Main Building Blocks as Identified in Chapter 1 ................ 158
Figure 72 Overview of the Contributions ..................................... 159
LIST OF TABLES

Table 1  Class Description for Linear Content .......................... 26
Table 2  Class Description for Non Linear Content ................... 28
Table 3  Description of Male Consumer/Viewer Persona ............... 35
Table 4  Description of Female Consumer/Viewer Persona .......... 36
Table 5  Description of Female Broadcaster/Producer Persona ...... 37
Table 6  Description of Male Broadcaster/Producer Persona ....... 38
Table 7  Classification of Presented Consumer Story Scenarios According to the Activity ............................................ 42
Table 8  Classification of Presented Broadcaster/Producer Story Scenarios .......................................................... 50
Table 9  Example for the Link in Modifying Synchronized Collaboration .................................................. 53
Table 10 Technical Guidance and Standards Established for the Reference Architecture .............................................. 76
Table 11 Pattern Description and Structure, Based on the Alexandrian Form .................................................. 77
Table 12 Technical Guidance and Standards Established for the Reference Architecture .......................................... 86
Table 13 Scenario Categorization ............................................. 104
Table 14 Rating of the Categories in Terms of the Level of Criticality .......................................................... 104
Table 15 Schedule of the Designed Collaborative Broadcast Content Format .................................................. 131
Table 16 Schedule of the XML Metadata Merged for this Scenario .......................................................... 134
Table 17 Results Experiment #1 ............................................... 146
Table 18 Results Experiment #1 ............................................... 146
Table 19 Results Experiment #2 ............................................... 147
Table 20 Results Experiment #2 ............................................... 147
Table 21 Requirements for the Consumer Client Module ............... 154
Table 22 Requirements for the Broadcaster/Producer Module ......... 155
Table 23 Synopsis for CiTV .................................................. 170
| Table 24 | Synopsis for LepidaTV / Multipler | 170 |
| Table 25 | Synopsis for GroupTV | 171 |
| Table 26 | Synopsis for CollaboraTV | 171 |
| Table 27 | Synopsis for Zync | 171 |
| Table 28 | Synopsis for CwaCTool | 172 |
| Table 29 | Synopsis for MIST | 172 |
| Table 30 | Synopsis for ResNet.TV | 173 |
| Table 31 | Synopsis for the Swedish Drama “The Truth about Marika” | 173 |
| Table 32 | Synopsis for the Finnish Production “Akvaario” | 173 |
| Table 33 | Synopsis for the British Drama “Sofia’s Diary” | 174 |
| Table 34 | Synopsis for SMS Games | 174 |
| Table 35 | Synopsis for TV Quizzes | 175 |
| Table 36 | Synopsis for the LIVE Project | 175 |
| Table 37 | Synopsis for the Social Quiz Game | 175 |
| Table 38 | Synopsis for the British Drama “BBC’s Spooks Interactive” | 176 |
| Table 39 | Synopsis for ConnecTV | 176 |
| Table 40 | Synopsis for 2BeOn | 176 |
| Table 41 | Synopsis for AmigoTV | 177 |
| Table 42 | Synopsis for Telebuddies | 177 |
| Table 43 | Synopsis for the German Tatort+ Episode ”Blackout“ | 177 |
| Table 44 | Synopsis for the German Crime Series “Schuld” | 178 |
| Table 45 | Synopsis for the German Crime Series “Dina Foxx” | 178 |
| Table 46 | Synopsis for the Belgian Drama “De Ridder” | 179 |
| Table 47 | Schedule of the XML Metadata Merged for this Scenario | 213 |
Part I

FUNDAMENTALS
INTRODUCTION

“Do not wait until the conditions are perfect to begin. Beginning makes the conditions perfect.”
— Alan Cohen

This chapter provides an introduction to the topic of this thesis. First the motivation and research challenges are clarified. In the next step the research questions and goals are proposed and hypotheses are formulated. Finally, the resulting contributions and the structure of the thesis are described.

1.1 MOTIVATION AND RESEARCH CHALLENGES

In the past decade, there has been a distinct shift in broadcasting from mainly passive analog to digital broadcast technology. The change to digital technology was obvious in the distribution and consumer sectors by substituting digital technologies in place of the analog broadcast in Europe. However, it must be stated that it also influences the entire television value chain (cf. Figure 1). The (television) content production value chain defines several steps from the content creation to the consumption of broadcast content via different user platforms. These changes facilitate new possibilities for collaboration which is growing in popularity in each of these steps.

The Content Creation phase includes the three consecutive stages of Pre-production, Production and Post-production [65]. The pre-production consists of the project preparation. This includes proposals, premises, synopses, treatments, scripts, production schedules, budgets and storyboards. An example for a collaborative feature is the collaborative script-writing. In brief and most importantly amongst other production steps, the production

---

1 Abstract model of the substance consumed by the audience [48].

Appendix A summarizes all definition of terms as used in this thesis.
phase includes the recording of the visual images and sounds. One example for collaborative production features is shown with “Wreck A Movie” or \( E = MC^2 + 1 \) by Artur Lugmayr [53]. Post-production starts right after the visual images and sounds have been recorded. In this phase, images and sounds are examined, improved and combined, as audio and visual elements are tied together.

In the Packaging and Aggregation phase, external sources are compiled to coherent content packages and are aggregated at program or schedule time [52].

Scheduling is the minute planning of the transmission; what to broadcast and when, ensuring an adequate or maximum utilization of airtime [52].

Distributors (like TV broadcast provider) transmit the purchased or in-house produced content to their consumers (also termed viewers) [52]. Example for collaborative features are Peer-to-Peer systems, connecting people with similar taste [70], found their way into the distribution of television program content\(^3\).

The distributed content is received and Consumed (viewed) by the consumers (TV viewers) via user (consumer) platforms. Viewers can act collaboratively with the content during the viewing as done with the “Kabel Eins” show “Leuchtturmmbauf auf Rügen”\(^4\) in 2005. Within this show, a lighthouse was built and co-created by the viewers on the holiday island Rügen. Finally it was raffled off to the participants. With special Web 2.0 services, as offered by A³TV [28], viewers can provide real-time feedback for the producers.

A transition has occurred from the typical producer-consumer-chain (cf. Figure 1) into consumer-centered models, where the consumer becomes content producer as well. Several TV channel providers (for instance the Austrian TV channel “Dorf TV”) specialize in broadcasting consumer produced content. Since consumer-centered models are prevalent in the streaming area (e.g., YouTube\(^5\)) but do not, at least not yet, dominate the television sector, this research work does not consider this case. Other streaming services like Netflix\(^6\) and Amazon Prime Instant Video\(^7\) are also not considered.

The main focus of this thesis is on the collaboration between consumers and consumer platforms in the television sector of the Austrian television landscape. Television was selected as the application area because it is economically and technically interesting. It is economically interesting because collaborative participation during air time is a good concept to bind users to the TV program format\(^8\) and by extension also make money (now 50 cent per SMS). It is technically interesting because of the special challenge of collaboration in the time-critical, real-time area that the television sector is. The transition of the consumer from being entirely passive to an active viewer who is participating in broadcast-content-related social-network functionality using mobile terminals, as well as the technological progress, turn television into an interesting application area.

---

3 Segment of broadcast content intended for broadcast on television.
5 http://www.youtube.com
6 https://www.netflix.com
7 http://www.amazon.com
8 Master plan and branding of a copyrighted television program.

Appendix A summarizes all definitions of terms as used in this thesis.
The noticeable development to digital broadcast technology brought along a boom in the focused consumer sector, by developing interactive applications (beyond teletext in TV) and inviting the viewer to participate in the TV program content in a collaborative manner. Several popular participatory reality and casting TV program formats have demonstrated this by inviting their audience - for instance - to affect the outcome of a TV show by voting. Therefore the concept of collaboration in broadcast has tradition and has gained new possibilities due to technological advances and developments. The popularity of such participatory program formats\(^9\) is indicated by their continuing production and broadcast as well as by their steadily rising numbers and wide-spread acceptance by the audience. Nevertheless, television can not keep up with the popularity of collaboration in other disciplines like computer games or Web applications, especially because the concept of participatory program formats barely extend beyond the well-known voting procedure or the viewer to asking questions via social network platforms.

So far, research and industry have perceived the trend of collaborative participation and the author could distinguish two different developments. On the one hand, research has focused on the development of TV platforms which provide social interaction and collaboration between viewers, but hardly any attention was given to TV program formats. This results in platforms providing collaborative services aside from the displayed TV program content and no association exists between both. On the other side, industry has concentrated on the development of participatory program formats that animate the audience to vote for a participant of the show or to ask questions via some social network. In contrast to research, in this context industry hardly focused on platforms. Participation is optimized for parallel platforms like Telephone or Internet. But the industry set the focus on adapting the concepts of Apps and App Stores from mobile phones to television but again, without taking participatory program formats into consideration.

In summary, TV program contents and formats with audience involvement, and TV platforms supporting collaborative and social activity both exist, yet remain unlinked in most cases. This poses a problem since synchronization is lost, and collaboration requires a common focus which, in broadcasting, is the program content. It is accepted for simple collaboration concepts like voting but it prevents the development of collaborative participatory program formats which requires more complex group interaction to solve a common task or a common goal.

The contribution is to extend the content value chain (cf. Figure 1) with aspects of collaboration (cf. Figure 2). Therefore, the topics of this thesis are situated in the content creation phase to firstly conceptualize a collaborative program content format\(^{11}\) (i.e. a collaborative content format); secondly to adapt broadcast technology as medium\(^{12}\) to be applicable as triggering and coordinating unit. And thirdly in the consumption phase to

---

\(^{9}\) The audience can play an active role.

\(^{10}\) The audience can play an active role in the process of manipulating, collecting, reporting and analyzing the program content.

\(^{11}\) Enables collaborative activity between consumers to reach a common goal, by using the program content format as coordinating unit.

\(^{12}\) Denotes the technical realization and representation of the content.

Appendix A summarizes definition of terms as used in this thesis.
trigger and enable collaborative activity via user platforms.

The research challenges are manifold and can be divided into challenges for broadcast and challenges for collaboration. To be able to enable more collaborative program formats, the broadcast industry is faced with many challenges:

1. Technical system solutions to enable the development of collaborative program formats, including synchronization mechanisms, reference architectures, distribution mechanisms and metadata solutions to describe the structure of program context.

2. Synchronization of the various technical components involved in enabling the collaboration between consumers and content as a coordinated matter. This includes various applications, sub-components, tasks, etc.

3. Enabling collaboration through synchronizing consumer interactions and collaboration to content on a thematic level, thus technical solutions to link content to collaborative features.

4. Development of a framework and a reference system including all it’s sub-components to enable collaborative program formats. This includes e.g. a producer module to “make” broadcasted A/V content collaborative and a consumer platform to enable collaboration to consumer.

5. Provisions of technological solutions to enable participants being part of the broadcast content, interaction possibilities, narrative structures\(^\text{13}\), and potential narrative\(^\text{14}\) flows of the program content.

6. Development of program formats that enable the utilization of technical broadcast infrastructure for collaborative purposes in form of a set of structures, patterns, or other possible forms as e.g. metadata.

\(^{13}\) Place for immersing the viewer into the narrative flow.

\(^{14}\) Series of events that are linked together in a number of ways, including cause and effect, time, and place.

Appendix A summarizes all definition of terms as used in this thesis.
Challenges on consumer level in the creation of more engaging and collaborative program formats are as follows:

1. Mapping of real-time program content to possible consumer interactions and collaborative actions.

2. Exploration of existing program formats and their potential to enable collaboration on first, social level; and second, content and program format level. Thus, the expansion of program formats and the potential features enabling collaboration between consumers and the TV content.

3. Collaboration as a social activity between strangers, large groups, or similar, and their social, psychological and behavioral aspects.

4. Incorporation of collaborative aspects on social, psychological, and behavioral level into the development of program formats by the development of technical solutions providing relevant functionality.

1.2 Research Questions and Goals

Based on the research challenges, the research questions and goals of this thesis result in the development of a reference architecture incorporating a set of predefined collaborative services on the one hand and a standard **Collaboration Medium Content Link (CMC Link)**\(^\text{15}\) mechanism on the other hand to realize the integration of collaboration into the **broadcast content format**\(^\text{16}\) and its subsets (cf. Figure 3). The architecture shall act as a tool kit for realizing a prototypical collaborative broadcast scenario. Therefore the research questions are summarized as followed:

**RQ #1: How can the CMC Link between collaborative activity, broadcast technology as medium and program contents and formats be designed?**

This question addresses the challenge of linking collaborative activity with both, the broadcast technology as medium and program contents & formats and addresses the direct integration of collaboration into the broadcast content format and its subsets.

*Link of collaborative activity to technology as medium:* linking collaborative activity to possibilities of technology. For instance the audience is invited to participate in a live chat on a certain topic, for a certain amount of time. The service is triggered (called, activated and deactivated) by metadata embedded in and transmitted via the broadcast technology as medium.

*Link of collaborative activity to the program content & format:* linking collaborative activity to course or characteristics of the program content. For instance a collaborative service is triggered, to help a candidate answer a question in a game show. The link is realized by using the candidate, who is acting in the scene, as a tag\(^\text{17}\). The appearance of this candidate can trigger collaborative activity.

---

\(^{15}\) Link between broadcast content formats (and its subsets) to collaborative activity. Also termed as link.

\(^{16}\) Consists of broadcast technology as medium and program contents and formats.

\(^{17}\) Tag represents a certain timestamp, scene, period of time, or any other defined reference of the broadcast technology as medium or the program content. Its purpose is to anchor trigger for collaborative activity.

*Appendix A summarizes all definition of terms as used in this thesis.*
1.2 Research Questions and Goals

**RQ #1**: How can the CMC Link between collaborative activity, technology as medium and program content & format be designed?

**RQ #2**: How must collaborative program content & formats be structured, and where are the tags and trigger\(^\text{18}\) to link technology as medium and program content & format to collaborative activity?

The structure of the content is related to the content’s storyline. The key is to design and produce participatory content that invites people to collaborate. Which genres are suitable? For the CMC Link, a set of tags and a trigger must be defined to allow linking collaborative activity with certain positions in the medium and in the storyline of the content. For instance, the appearance of a certain person or object in a scene, or the beginning of a sequence of frames, time stamps, or metadata in the medium can trigger collaborative activity.

**RQ #3**: How can collaborative activity be designed on a human / consumer perspective?

Are existing collaborative mechanisms (e.g. communication mechanisms, concurrency control, context awareness) and tools (e.g. computer-supported cooperative work tools, rating and recommendation systems) applicable for this purpose and for the connection to a

\(^{18}\) A trigger is usually anchored to one or several tags and prompts (enable, disable, update) collaborative activity.

---

Figure 3: Classification of the Identified Research Questions to the Main Parts of Broadcast Technology as Medium, Program Content & Formats, Collaborative Activity and CMC Link

Link of technology as medium to the program content & format: provision of technical possibilities (e.g. metadata) to program formats to provide structure for program contents (e.g. “vote” path).

**RQ #4**: How can collaborative activity (actions) and interaction be identified, observed and processed on a system level?

**RQ #5**: What are the technological requests, features, and system architectural aspects?
broadcast content format and its subsets? In addition, tags and triggers in the collaboration that correspond to those in the medium and content, are necessary (e.g. to ascertain the majority, reactions and counter-reactions).

**RQ #4: How can collaborative activity be identified, observed and processed on a system level?**

To use collaborative interaction for further application (e.g. to modify the course and characteristics of content), it is necessary to measure (e.g. the level of activity), analyze (e.g. the outcome of the collaborative activity) and finally quantify (i.e. indicate the outcome of the measurement and analysis as numeric values) the collaborative activity.

**RQ #5: What are the technological requests, features, and system architectural aspects?**

In essence, which technology supports user participation and collaboration best? It is necessary to define technical requirements (e.g. a run-time environment), security requirements (e.g. privacy and resistance against attacks), exception handling (e.g. handling the unexpected drop-out of participants) and the manner of support for this real-time system.

The research goal of this work is a result of the previously defined research questions and is about how to best process and answer the formulated research questions.

### 1.3 Hypotheses

Based on the previously defined questions and goals the following hypotheses are proposed. The undirected hypotheses #1 and #2 pertains to broadcast technology as medium and program contents & formats. The modification hypotheses #3 and #4 concerns the collaboration.

Hypothesis #1: “On technology level, it is feasible to link collaborative activity and broadcasting technology as a medium through advanced technical solutions based on existing broadcasting infrastructure”.

Hypothesis #2: “Additional solutions on content level, as well as on technology level are required to link collaborative activity and program content and formats which also includes a greater understanding of the content production workflow and narrative aspects.”.

Hypothesis #3: “Digital broadcasting requires the exploration of new emerging technologies that enable the consumer to collaborate, in a particular reference implementation thereof to make the consumer aware about the potential.”.

Hypothesis #4: “To test collaborative systems, a new set of benchmark and evaluation criteria have to be developed”.

In order to verify hypothesis #1, a system will be developed that utilizes broadcast technology enabling collaborative activity (cf. Chapter 5). For verifying hypothesis #2, an evaluation of existing program formats will be conducted (cf. Chapter 2). In addition, dedicated case studies will be conducted followed by the conception and the realization of
a non-linear, participatory program format (cf. Chapter 3). It shall enable modifications to the program content based on collaborative activities of the consumer. Hypothesis #3 will be verified by the examination of a reference architecture including existing design patterns for collaborative services as well as the presentation of concepts of how collaborative activity in the television area can be identified, observed and processed to in turn modify the broadcast technology as medium and the program content & format (cf. Chapter 4). Finally, hypothesis #4 will be verified by the development of a set of benchmark and evaluation criteria (cf. Chapter 6).

1.4 CONTRIBUTION

The major contribution of this PhD research is to create the missing CMC Link between broadcast technology as medium, non-linear program contents & formats and collaborative activity from a technical perspective. Therefore, the outcome of this thesis are firstly the conception of a collaborative program content format (cf. (1) in Figure 4), secondly the development of two main modules for producer and broadcaster (cf. (2) in Figure 4) to integrate metadata into currently broadcasted A/V material (cf. (3) in Figure 4). The metadata turns conventional and mostly passive A/V material into a coordinating unit. And thirdly the realization of a consumer module to enable collaborative activity on the consumer side (cf. (4) in Figure 4). And fourthly the observation and processing of the collaborative outcome (cf. (5) in Figure 4).

To realize this, the work will produce eight intermediate results (cf. Figure 5), which are summarized below.

1. Comprehensive analysis of (collaborative) program contents & formats as well as broadcast consumer platforms is performed with regards to the television area (cf. (1) in Figure 5). This results in a systematic classification concerning the Narration Space, Properties of the Collaborative Activity, Link of Collaborative Activity to the Program Contents & Formats, and Hardware Features.
2. Development of usage scenarios resulting in functional and non-functional requirements to a system providing collaboration in a television environment (cf. (2) in Figure 5). This results in identifying initial and theoretical links between the program content & format and the collaboration.

3. Realization of linking technologies between the broadcast technology as medium, program contents & formats and the collaborative activity, regardless of whether or not the collaboration modifies the broadcast medium and program content (cf. (3) in Figure 5). The CMC Link is designed by using metadata which are included into the broadcast medium and interpreted on the consumer end. The metadata trigger the collaborative activity and act as a coordinating unit between collaborators.

4. Development of a software reference architecture for realizing the prototypical collaborative broadcast scenarios defined in step two by the direct integration of collaboration into the broadcast content format and its subsets (cf. Figure 4 and (4) in Figure 5). Apart from goals and objectives of the reference architecture, it describes the full system architecture of a collaborative broadcast system and necessary standards.
5. Design patterns (for collaborative services, the link and the integration of collaborative services to broadcast technology as medium and program contents & formats) are given to enable an easy implementation of the concept (cf. (4) in Figure 5). To consider economic parts, the consumer and producer context and the business lifecycle of a collaborative program content format are outlined.

6. Conception of a non-linear, participatory and collaborative program content format that invites the audience to become active and collaborative participants instead of adding linear and passive program content, as is mostly done so far (cf. (5) in Figure 5).

7. Development of two prototypes to prove the contributions of the main concept (cf. (6) in Figure 5).

8. Finally, both prototypes are tested and evaluated according to the identified requirements (cf. (7) in Figure 5).

<table>
<thead>
<tr>
<th>Research Questions</th>
<th>Hypotheses</th>
<th>The Design Science Research Process</th>
<th>Methods</th>
<th>Outcome</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ #1 RQ #2 RQ #3 RQ #4 RQ #5</td>
<td>H #1 H #2 H #3</td>
<td>Analysis</td>
<td>Literature research. Expert groups. Case studies. Requirement Analysis.</td>
<td>Missing CMC Link Lost Synchronization Media Disruption</td>
<td>2, 3</td>
</tr>
<tr>
<td>RQ #1 RQ #2 RQ #3</td>
<td>H #1 H #2 H #3</td>
<td>Design</td>
<td>Model conception and artifact construction by the conception and design of the reference architecture.</td>
<td>Reference architecture for the development of collaborative broadcast systems including: System Architecture, Design Patterns, CMC Link Mechanism by the inclusion of metadata into the A/V material, Business Architecture.</td>
<td>4</td>
</tr>
<tr>
<td>RQ #5</td>
<td>H #1 H #2</td>
<td>Development</td>
<td>Artifact construction by the prototypical implementation of the system architecture and patterns of the reference architecture.</td>
<td>Development of the three main Modules „Consumer Module“ (incl. „CBF Player“), „Producer Module“ (incl. „CBF Metadata Editor“), „Broadcaster Module“</td>
<td>5</td>
</tr>
<tr>
<td>RQ #1 RQ #2 RQ #5</td>
<td>H #1 H #2 H #3 H #4</td>
<td>Evaluation</td>
<td>Artifact destruction by the prototypical implementation of two previously designed use cases. Artifact destruction by measuring limits of applicability and effectiveness. Verification.</td>
<td>Development of the „CBF Voting App“ and The „CBF Gaming App“. Metadata are feasible to realize the missing CMC Link.</td>
<td>6</td>
</tr>
</tbody>
</table>

Figure 6: Overview of the Research Work Resulting in this Thesis

Figure 6 summarizes the main steps and the used methodology of this work in more detail and presents a schedule of the processing of research questions and hypotheses.

According to these contributions, the thesis is structured as follows:

1. To clarify and provide further motivation for the contributions, the next chapter discusses related work of broadcast content formats and consumer platforms, with a focus on the television area. The analysis is done along an evaluation framework, resulting in a systematic classification concerning the Narration Space, Properties of the Collaborative Activity, Link of Collaborative Activity to the Broadcast Content Format, and Hardware Features.
2. Chapter 3 steps through motivating collaborative broadcast example scenarios, resulting in the deduction of functional and non-functional requirements.

3. Chapter 4 presents the developed reference architecture which bridges the gap between broadcast technology as medium, program contents & formats and collaborative activity.

4. A prototypical implementation of presented example scenarios presented in Chapter 3, by using the software reference architecture presented in Chapter 4 is given in Chapter 5.

5. Chapter 6 presents the proof-of-concept and evaluation of the prototypical implementation by the development and evaluation of the dedicated case studies concerning the requirements deduced in Chapter 3.

6. The thesis concludes in Chapter 7 with the verification of the hypotheses, a consolidation of the research work of the past few years leading to this thesis, shortcomings, open issues, and outlooks on further research directions.
STATE-OF-THE-ART WORK OF DIRECT INTEGRATION OF COLLABORATION INTO BROADCAST CONTENT FORMATS

Broadcasting is really too important to be left to the broadcasters.
— Tony Benn

Since the 1990’s interactivity has been a significant feature of media culture [69]. The digitalization, internationalization and marketing of the media has created new competitive online- and mobile communication forms that have changed both TV and radio. This development can be seen in the increased competition and cooperation with online media [9]. TV has become part of a bigger puzzle of interconnected devices operating on several platforms rather than just one. Multiplatform itself as a term, in the computer industries and occasionally in media research, tends to focus on digital technology providing a technological and design framework (a so-called content management system) for incorporating existing media (e.g. text based or audiovisual), platforms (e.g. chats, blogs, discussion groups), or software systems (e.g. Linux or Windows, GIF or JPEG) [42]. In this chapter, based on Espen Ytreberg’s idea [84], the term is used to include formats that feature a central cluster of television, Web and telephony platforms, as well as various peripheral platforms. Multiplatform designates a particularly complex subspecies of output, variously characterized in more general terms as crossmedia, intermedia, and transmedia.

As of the publication of this thesis (up to 2015), a survey covering the extensive field of collaboration in broadcast content formats, has not (yet) been found. A comprehensive analysis of existing research work, categories, and criteria was created and collected in a catalog. This catalog should cover all extracted features and functionality of existing work as well as cover and constrain the scope of this thesis.

2.1 METHODOLOGY FOR REVIEWING AND CLASSIFYING STATE-OF-THE-ART WORK

Goal of this Chapter is, to define a classification schema that allows us to categorize the inventory of related work in a complete and comprehensive manner. From this schema, multiple evaluation criteria can be extracted. The applied methodology, based on this carefully established catalogue of criteria, classifies and evaluates gaps in approaches providing collaboration in broadcast content formats.
2.1 Methodology for Reviewing and Classifying State-of-the-Art Work

2.1.1 Selection of Approaches

To the best of my knowledge, there has been no categorization of collaboration in broadcast content formats as a whole, although a considerable amount of approaches exist. As defined in the Introduction (cf. Chapter 1), this thesis focuses first on the production phase, to conceptualize a collaborative broadcast content format, second on the post-production phase, to link the broadcast technology as medium, the program content, and the collaborative activity, and third, on the consumption phase, to enable collaborative activity on a consumer platform. The selection of state-of-the-art work focuses on the same areas:

1. Collaborative broadcast content formats.

2. Established broadcast technology data formats (like MPEG-2).

3. Consumer platforms, which could be platforms on a conventional television device at the viewers home, apps on mobile devices or personal computers designed to participate or modify the broadcast content format collaboratively.

No single user (pre-/post-)production software, computer supported cooperative work (CSCW), or peer-to-peer systems for collaboration in the (pre-/post-)production or distribution level are included in the survey, as they go beyond the scope of this thesis. The motivation for choosing the remaining systems is to assort a representative mix of well-published approaches. In this respect, the goal is to maintain the ratio between pure consumer platforms dealing with any kind of broadcast content format and specially designed non-linear participatory broadcast content formats which invites the viewer to play an active role.

2.1.2 Conveyance of Uncatalogued Criteria

In the following, a catalogue of criteria for a structured classification of approaches in collaborative broadcast content formats is proposed. The idea behind designing this was to provide a fine-grained catalogue that constitutes the prerequisite for an in-depth classification of existing approaches and thus allows us to compare multiple higher-level approaches more accurately than previous surveys such as Jensen [43] [44], Ursu [81], Tuomi [79], and Carey [21]. The categories for the classification were first derived in a bottom-up manner, from the content structure to the delivery network. Second, each criterion is again diversified into sub-criteria, in a top-down manner, further leading to other sub-criteria. Criteria and sub-criteria definitions found in other surveys have been partly adopted and if necessary refined, e.g. The Content Narrative Space, which was taken from Lugmayr et al [54], or the Hardware Features, which are well-established classifications taken from standard references in the field of computer networks, digital media, and Internet, like [77], [78], [19], [65], and [60]. In the criteria descriptions, it is indicated if and how the corresponding criterion has been adopted or refined from related work.
2.1.3 Excluding Non-Relevant Criteria

A few criteria had to be excluded from the presented catalog. This is because most approaches can be categorized concerning their performance, implementation, platforms, usability, and user acceptance or provided functionality. Some criteria must be excluded, since they cannot be measured without user studies and/or extensive case studies for which no resources are available.

2.1.4 Establishing a Schema for Categories and Criteria Definition

Furthermore the goal was to conceive unambiguous definitions and quantitative values of the criteria. Therefore each criterion is described by a set of properties:

1. **Name** and **reference** to the source(s) in case a category/criterion has been adopted or refined from another source as well as a description of how it was refined.
2. **Case or example** to point up the description of the category/criterion. Please consider, the mentioned cases and examples are not exhaustive. They do not depict the entire list of existing cases and examples.
3. In case of redefining a category/criterion, a list of **arguments** are given to emphasize the necessity of the corresponding category/criterion for this approach.
4. **Definition** specifying the category/criterion as clearly as possible with the possibility of an optional **discussion** in case of difficulties or ambiguity.

In the following, the catalogue of categories and criteria is presented.

2.2 Categorization of Consumer Collaboration in Broadcast Content Formats

In the following, five categories are presented in order to classify existing approaches of collaboration in broadcast content formats. The categories were chosen to give a high-level review of the functionality and features but not to focus on specific details of the examined approaches. The categories starts with the **Content Narrative Space** of the broadcast content format, for which the examined approach was designed. Next, the **Collaboration** functionality is outlined and the **Link** between the collaborative activity and the broadcast content format is considered. In addition, **Hardware Features** are recorded. An overview is shown in Figure 7.

2.2.1 Narrative Perspective on Broadcast Content

A narrative is commonly referred to as a story. A “story” is a term that is understood intuitively but hard to define. For our purpose, a narrative is defined as a series of events that are linked together in a number of ways, including cause and effect, time, and place. Something that happens in the first scene causes the action in the second scene, and so on (while also moving forward in time) [35]. The content narrative space
2.2 Categorization of Collaboration in Broadcast Content Formats

<table>
<thead>
<tr>
<th>Category</th>
<th>Criteria</th>
<th>Sub-Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Narrative Space</td>
<td>Linear Structure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Branched Structure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-Deterministic Structure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evolutionary Structure</td>
<td></td>
</tr>
<tr>
<td>Link Collaborative Activity To Medium &amp; Content</td>
<td>No Link to M &amp; C</td>
<td>Correlation</td>
</tr>
<tr>
<td></td>
<td>CMC Link to Medium</td>
<td>Modification</td>
</tr>
<tr>
<td></td>
<td>CMC Link to Content</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Collaborative Activity</th>
<th>Scope</th>
<th>Collaborative Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Distribution</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Content Recommendation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Schedule Creation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Content Creation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Social Experience</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gaming</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Medium Modification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Content Modification</td>
<td></td>
</tr>
<tr>
<td>Reach</td>
<td>Closed Group</td>
<td>By Enhancement</td>
</tr>
<tr>
<td></td>
<td>Open Group</td>
<td>From existing Content</td>
</tr>
<tr>
<td>Representation</td>
<td>Buddy List</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avatars</td>
<td></td>
</tr>
<tr>
<td>Interaction Modality</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Video</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Emotions</td>
<td></td>
</tr>
<tr>
<td>Synchronization</td>
<td>Synchronous</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asynchronous</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hardware Features</th>
<th>Consumer Platform</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>PC, Laptop</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mobile Device</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conv. TV Environment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phone</td>
<td></td>
</tr>
<tr>
<td>Second Screen</td>
<td>DVB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPTV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Misc</td>
<td></td>
</tr>
<tr>
<td>Broadcast Network</td>
<td>MPEG-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MPEG-4</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7:** Catalogue of Categories and Criteria to Classify Collaboration in Broadcast Content Formats
(or content structure) is defined as a place for immersing the viewer into the narrative flow, not only explicitly as a kind of co-director, but also by building virtual communities, communicating with other interested viewers and obtaining natural or narrated multimedia assets. Furthermore it is an indicator whether a given broadcast content is applicable for content-related interaction and collaboration. Four narrative spaces, namely linear, branched, non deterministic, and evolutionary are distinguished in literature.

The definition of content narrative space for this classification and evaluation is partly taken from Lugmayr et al [54], Samsel et al [73] and Garrand [35].

Note: Below, The Content Narrative Space is also referred to as Content Structure.

2.2.1.1 Sequential (Linear) Structure (N.LS)

Sequential structure is the basic building block of linear and interactive media projects. Sequential (or linear) structure has no linking or interactivity. It is the structure of most broadcast content formats. The user navigates a strictly defined procedural path (cf. (a) in Figure 8), without the possibility to jump or skip a node. In interactive media projects, the linear structure acts not as a primary course but as a recurring theme that keeps things moving along. Its beginning and ending are known beforehand.

Case N.LS #1: conventional broadcast content formats like movies, news or documentaries. A variant of the linear structure is adding Cul-de-Sacs\(^1\) to single nodes (cf. (b) in Figure 8). This offers the user the choice to step off the procedural path into areas that do not affect the critical objective of the piece. Puzzles or games usually follow this principle.

Case N.LS #2: weather forecast or stock news-ticker application on digital TV.

2.2.1.2 Branched Structure (N.BS)

Branches in narratives enable story evolution by offering the user different alternatives (cf. (c) in Figure 8). Based on which path the user chooses, the program follows a new node of content. This demonstrates the fundamental concept of interactive theory - user choice.

Case N.BS #1: The continuance of the storyline depends on the results of voting procedures. A variant of branching is conditional branching. This subset requires the user to abide by predetermined rules and conditions (e.g. puzzles that must be solved by the user to continue, cf. (d) in Figure 8) along the branch in order to proceed through the program. Subsets of conditional branching are branching with bottleneck, forced paths and optional scenes (cf. (e), (f), (g) in Figure 8). In bottleneck (case N.BS #2), various branching nodes are brought back to a specific story node.

Case N.BS #2: Bottlenecking happens mostly in games, where, for example, the user can solve side stories voluntarily, but at some point in the game, must get back to the main path.

Forced paths (case N.BS #3) offers the user multiple options and paths to choose from but only one solution.

Case N.BS #3: Imagine a game that offers main objectives for the user to solve. Those objectives guide the user through the game until the end (forced path). Besides that, there are additional objectives that can be solved voluntarily, but are not a determining factor to finish the game.

The user must choose between alternative scenes that spin out from and return to the

\(^1\) dead-end street with only one inlet/outlet application [29]
Figure 8: Geometric Design Structures for the Content Narrative Space (a) Linear, (b) Cul-de-Sac, (c) Branched, (d) Conditional Branch, (e) Bottleneck, (f) Forced Paths, (g) Optional Scenes, (h) Parallel, (i) Exploratorium, (j) Hypermedia, (k) Worlds [73]
primary path of the application.

Case N.BS #4: Learning CD for driving license which offers different scenes of the same topic to play for the examinee.

2.2.1.3 Non-deterministic Structure (N.NS)

In non-deterministic content structures, the space (place of action) is controlled by the viewer. The storyline and current status are broadcast to keep viewers up-to-date. Places, guidelines, limitations, etc where the viewers/consumers interact are created by the director. Parallel streaming and the exploratorium are two examples of non-deterministic structures. Parallel streaming describes many states or paths that exist simultaneously at various levels within the same application. This allows the writer to create a linear story, while also allowing the user to switch between perspectives, paths or states (cf. (h) in Figure 8).

Case N.NS #1: Parallel Streams: A game where users can switch between various protagonists.

An exploratorium that allows the users to explore a world within a world (cf. Figure 8 - (i)).

Case N.NS #2: Exploratorium: A second screen app is offered to linear broadcast content providing additional interactive and collaborative functionality.

2.2.1.4 Evolutionary Structure (N.ES)

The evolutionary structure combines hypermedia (cf. (j) in Figure 8) and worlds (cf. (k) in Figure 8). Worlds interconnect two or more environments by a common thread (usually a goal, mission or story).

Case N.ES #1: The video game “World of Warcraft” is one popular example. It combines several worlds into one storyline.

Hypermedia either encompasses every user path imaginable or no path at all. This offers the most freedom by letting communities creating their own story flow, making the end of the story unpredictable.

Case N.ES #2: Internet is the most complex and powerful hypermedia. Incidents in SecondLife depend on actions and behavior of other players.

2.2.2 CMC Link of Collaborative Activity to Broadcast Technology and Program Contents & Formats

This category focuses on the level of connectivity between collaborative activity and the broadcast medium and/or content (as defined in the Introduction, cf. Chapter 1).

2.2.2.1 No link to Broadcast Technology and/or Program Contents & Formats (L.NC)

Collaborative services are offered additionally but independently to the broadcast technology. Former social television services fall into this criteria. A social service (e.g. chat) on the TV environment is one well known example. These services are offered in parallel with and completely unconnected (no correlation, no modification) to the broadcast content

2 https://eu.battle.net/account/creation/wow/signup/
format (in a technical manner and concerning to the storyline).

Case L.NC #1: Social services running on the television set or the set-top-box without any regard to the broadcast content format.

2.2.2.2 CMC Link to the Broadcast Technology as Medium (L.LM)

This category addresses the technical dependency on the broadcast technology as medium (also termed as broadcast medium or medium). A link to the broadcast medium can be realized in two levels: correlation and modification.

Correlation (CO): Offering collaborative functionality and applications in correlation with properties of the broadcast medium, e.g. time dependence, without modifying it at all.
Case L.LM - CO #1: Connecting people watching the same broadcast medium.

Modification (MOD): Offering collaborative functionality and applications to modify the broadcast medium (e.g. control the medium collaboratively, enhance it, and change its properties). Modification implies correlation.
Case L.LM - MOD #2: An electronic program guide (EPG) that is distributed via the broadcast medium, containing recommendations from others.
Case L.LM - MOD #3: In a collaborative broadcast application, each participant is able to change the state of the broadcast medium (play, pause, forward, etc).
Case L.LM - MOD #4: Change metadata, which are embedded into the broadcast medium, collaboratively or use them as anchors to further collaborative actions.

2.2.2.3 CMC Link to the Broadcast Content & Format (L.LC)

The proposition of collaborative services is related to the characteristics of the broadcast content. For example, offering a voting application to a television casting show. The application is only available during the broadcast of the show. The usage of those services does not necessarily modify the broadcast content. A CMC Link to the broadcast medium is not required.

A CMC Link to the broadcast content can be realized in two levels: correlation and modification.

Correlation (CO): Offering collaborative functionality and applications in correlation with broadcast content (e.g. a chat to discuss certain content during the broadcast).
Case L.LC - CO #1: Puzzles, quizzes, etc are solved on a related web page. The results do not modify the course of the broadcast content.
Case L.LC #3: A chat for discussion is offered to a political live discussion show. The postings and discussions in the chat may (modify) or may not (correlation) modify the course of the political discussion show.

Modification (MOD): Offering collaborative functionality and applications to modify broadcast content (e.g. offering a voting application for a music or casting show). Modification implies correlation.
Case L.LC - MOD #2: Content related chat. The mood and opinion of the discusser modifies the course of the broadcast content.
Case L.LC #3: A chat for discussion is offered to a political live discussion show. The postings and discussions in the chat may (modify) or may not (correlation) modify the course of the political discussion show.

2.2.3 Collaborative Activity

The category Collaborative Activity describes the scope, reach, representation, modality, and synchronization of the provided collaboration functionality concerning the consumed broadcast content format. Please note that collaboration is essential in all criteria and actions.

The previously presented category of Link and the subsequently presented Scope of Collaborative Activity constitute a link between broadcast technology as medium, program content & format, and collaborative activity that describes the main issue of this thesis.

2.2.3.1 Defining the Scope of Collaborative Activity

This criterion is about the scope, aim, and objective of the provided collaboration functionality.

Distribution (S.DI) Collaborative distribution of the broadcast content format, as done in peer-to-peer systems. No further connection to the medium or content is given. Case S.DI: Collaborative stream and broadcast - people distributing content or sharing a medium without additional modifications.

Content Recommendation (S.CR) Collaborative content recommendation refers to recommendations of broadcast content to single viewers. The recommendations are based, amongst other things on the overall preferences of a group of viewers. Case S.CR #1: Content currently being watched by the majority of the viewers is recommended.

Schedule Creation (S.SC) This criterion is related to the collaborative creation of a TV channel schedule, and is characterized by sharing uploaded content and live streams. The channel's schedule is determined by the viewers' shared content. Case S.SC #1: A television channel which follows this concept by broadcasting user generated content.

Content Creation (S.CC) This criterion targets the collaborative creation and enhancement of content, e.g., enhancement with annotations. Please note, this criterion does not include CSCW systems to support collaboration in the (pre-/post-) production phase. However, it does include, for example, new and/with existing content or, as previously mentioned, enhancing existing content with new elements at the consumer level. Note: No further distinction is made in this category for broadcast medium and content, because the creation of new content requires an existing medium or the simultaneous creation of new medium. Case S.CC #1: Collaboratively annotate content. Case S.CC #2: Create a new medium realizing and transmitting music clips cut from existing video clips.
Social experience (S.SE)  Social experience includes interaction with other users and assumes awareness of each other, as seen with instant messengers or the attendance in social networks. This criterion resides another growing branch of research: the social television area. Researchers count social television as a segment of collaboration in broadcast content formats.

Case S.SE #1: Analyzing social network activity (e.g. discussions related to the broadcast content format) in order to modify the course of the broadcast content.
Case S.SE #2: Providing Twitter functionality on a TV channel as a platform to share opinions and feelings with the broadcast content.

Gaming (S.GA)  In this case, the scope, or the aim, of the collaboration is gaming.
Case S.GA #1: Collaborative participation on a quiz show.

Interaction with broadcast technology as medium (S.MM)  This criterion deals with the modification in general, not how (e.g. by social activity or annotations) the modification happens. Therefore this criterion is usually merged with others in this category.
Case S.MM: In a collaborative broadcast application, each participant has the right to change the state of the medium (play, pause, forward, etc).

Content modification (S.CM)  This criterion aims to enable collaborative actions in special broadcast content formats. Collaborative actions are initialized by the content's storyline (to keep collaborative activities continuing in a certain direction) and technical structure (e.g. voting issues). This criterion's objective is to collaboratively modify the broadcast content. Modification or alternation is, for example, changing the content's storyline by using and interacting with certain external applications. This criterion deals with the modification in general, not how (e.g. by social activity or annotations) the modification happens. Therefore this criterion is usually merged with others in this category.
Case S.CM: Direct the storyline into one direction via voting procedures.

2.2.3.2 Social Groups of Collaboration
This criterion characterizes whether the reach is limited to closed groups or scaled to open groups.

Closed group (R.CG)  A closed group is defined by its limitations for participation. The most common limitations are restricting access to family/friends/village/country only or limiting the number of group members.
Case R.CG: Closed groups with a maximum of three members compete against each other within a broadcast content format.

Open group (R.OG)  Open groups have no limitations and strangers (S) are able to communicate and collaborate with each other.
Case R.OG: An open group competes against participants in a TV studio.

2.2.3.3 Embodiment of Self and other Users
This criterion is about how users represent themselves to other users. Options include buddy lists (P.BL) and avatars (P.AV). The list of representation options is limited to those
that are already in use in state-of-the-art work and meaningful for the Television sector. 
*Case P.BL:* Collaborators in a closed group with hundreds of members are quoted in a buddy list. 
*Case P.AV:* Collaborators are represented by their avatars.

2.2.3.4 **Interaction Modality**

This criterion is about how collaborators interact with each other. Options include text (M.TE), audio (M.AU), video (M.VI) and emotions (M.EM).

*Case M.TE:* Collaborators communicate via chat. 
*Case M.AU:* Collaborators communicate via audio conference. 
*Case M.VI:* Collaborators communicate via video conference. 
*Case M.EM:* Collaborators communicate implicitly by expressing their emotional response to the actions, content, etc of others.

2.2.3.5 **Synchronization**

This criterion deals with whether the collaboration takes place synchronously (SY.S) or asynchronously (SY.A) to the broadcast and synchronously or asynchronously within the group.

*Case SY.S:* Collaborators can compete with participants in a TV studio in the context of a live TV game show. 
*Case SY.A:* Collaborators can explore an exploratorium of a broadcast content format after its broadcast.

2.2.4 **Technology Features**

The category *Technology Features* describes features and characteristics of the broadcast consumer platform at which the observed approach is working. Please note, only a limited amount of technology features could be ascertained, as far as noticeable in the corresponding reference paper. It includes the following criteria:

2.2.4.1 **Consumer Platform**

The consumer platform refers to the hardware setup on which the considered approach is operating. This could be a personal computer or laptop (CP.PC), a Mobile Device (as first screen) (CP.MD), a conventional television environment (CP.TV), or a phone (CP.PH).

*Case CP.PC:* A laptop is used to explore an exploratorium of a broadcast content format after its broadcast. 
*Case CP.MD:* A tablet is used to watch and participate with the collaborative broadcast content format by using the provided platform. 
*Case CP.TV:* The television set, including a set-top box, is used to watch and participate with the collaborative broadcast content format by using the provided platform. 
*Case CP.PH:* The telephone is used to participate in the voting of a collaborative broadcast content format.
2.3 Classification of Collaborative Program Contents & Formats

2.2.4.2 Second Screen

The second screen criterion answers the question if a second screen (SS) is considered or not.

SS: A mobile device is used as second screen to participate in a collaborative broadcast content format while the user watches simultaneously on a TV.

2.2.4.3 Broadcast Network

This criterion depicts the underlying broadcast network structure.

Case B.CS: A client-server network structure.
Case B.DVB: A digital video broadcast (DVB) architecture.
Case B.IPT: An IPTV architecture.
Case B.DIV: Any other broadcast network architecture.

2.2.4.4 Data Format (HW.DF)

The last criterion depicts the data format of the broadcast medium.

D.M4: MPEG-4.

2.3 Classification of Collaborative Program Contents & Formats

The following section, as well as Appendix B, lists and describes existing solutions for collaboration in broadcast content formats. First, approaches are categorized according to their link level to provide a well-regulated overview. Second, each approach concerning to the Content structure, Collaboration and Hardware Features (cf. Figure 7) is described (see Appendix B), but without going into detail about their system architecture, evaluation, etc. Third all approaches are categorized and evaluated in detail with regards to the aforementioned categorization. The results are presented in Figure 9. Note: The numbering and order of the approaches in Table 9 corresponds to the numbering and order in Appendix B.

Fourth, to demonstrate correlations between the previously designed categories and criteria, mono-hierarchical classes were built, to classify all approaches. Finally, one representative approach per class is described in detail.

Using a top-down approach, the class hierarchy starts on the top layer of the catalog, combining The Narrative Space with the Link. Second it descends to the Scope of Collaboration (cf. Figure 9 from left to the right) which reflects the structure as defined in the Introduction (cf. Chapter 1). Due to its irrelevance to the structure and a lack of information, the lower Hardware Feature layer is not used in the class hierarchy.

2.3.1 Linear Narrative Space

The linear narrative space can be divided into three classes (#I-#III) according to their link to the broadcast medium or content. Below, these classes are described and approaches
Figure 9: Overview of the Classification of Approaches in Collaboration in Broadcast Content Formats (see Appendix B for short description)
Note: For all class #I approaches the type of content and narrative structure is obscure. Linear narrative structure is assumed since it is the most common broadcast content structure.

<table>
<thead>
<tr>
<th>Class Identifier</th>
<th>Class Description</th>
<th>Classification*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class N.LS:</td>
<td>Linear narrative space / Broadcast Content format and narrative space not relevant.</td>
<td></td>
</tr>
<tr>
<td>Class N.LS - #I:</td>
<td>Approaches that are not linked to medium or content.</td>
<td>None</td>
</tr>
<tr>
<td>Class N.LS - #II:</td>
<td>Approaches that are linked to the medium.</td>
<td>1, 2, 3, 4, 5, 6, 7, 17</td>
</tr>
<tr>
<td>Class N.LS - #II.A:</td>
<td>Focus is on common watching and sharing the controls (like play, pause, etc) within a community.</td>
<td>5, 6, 7</td>
</tr>
<tr>
<td>Class N.LS - #II.B:</td>
<td>Focus is on the collaborative annotation of content.</td>
<td>4, 6</td>
</tr>
<tr>
<td>Class N.LS - #II.C:</td>
<td>Focus is on collaborative distribution of linear content and sharing within a community.</td>
<td>1, 2, 3, 17</td>
</tr>
<tr>
<td>Class N.LS - #III:</td>
<td>Approaches that are linked to the content.</td>
<td>2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24</td>
</tr>
<tr>
<td>Class N.LS - #III.A:</td>
<td>Emotions and moods of the viewer correlate (# 4, 19) with the content or modify it (# 10, 14).</td>
<td>4, 10, 14, 19</td>
</tr>
<tr>
<td>Class N.LS - #III.B:</td>
<td>Focus and aim of collaboration is gaming.</td>
<td>9, 11, 12, 13, 15, 16, 20, 21, 22, 23</td>
</tr>
<tr>
<td>Class N.LS - #III.C:</td>
<td>Focus is on social activity, besides linear content, with correlation to the broadcast content.</td>
<td>2, 3, 4, 5, 6, 7, 8, 17, 18, 19, 20, 24</td>
</tr>
</tbody>
</table>

Table 1: Class Description for Linear Content. (* Number # of Approach According to Figure 9 and Appendix B)

Class N.LS - #I - No link to medium or content

No approach was found that didn’t provide any link between the broadcast content format and the collaborative activity. Class N.LS - #I stays empty after the revision and update of this survey chapter.

2.3.1.1 CMC Link of Collaborative Activity to Broadcast Technology as Medium

Class N.LS - #II.A - Common watching and sharing controls (to modify the medium)

Class #II.A summarizes approaches that focus on social experiences by
common watching as well as on sharing its controls. More precisely, the status of the medium, whether it is played or paused can be changed collaboratively within the group. One popular example is Zync. Zync is a plugin for Yahoo! Messenger\(^3\), which provides a social and collaborative experience for online videos. Based on the (Yahoo!) Messenger buddy list, participants can share content to be synchronized with buddies as well as its controlling mechanisms.

**CLASS N.LS - #II.B - COLLABORATIVE ANNOTATION OF CONTENT** The approaches summarized in this class focus on the collaborative annotation of content (like with the CWAC tool). While a selected video frame is annotated, people can simultaneously view each others’ ink annotations and discuss them using the chat tool [62].

**CLASS N.LS - #II.C - DISTRIBUTION AND CONTENT SHARING** Class #II.C summarizes approaches whose focus is on the collaborative distribution and sharing of content within a community. The type of content (e.g. genre) and its narrative structure are not relevant for the collaborative features. One example system is CiTV, which focuses on the collaborative delivery of on-demand IPTV services [59].

### 2.3.1.2 CMC Link of Collaborative Activity to Broadcast Content & Format

**CLASS N.LS - #III.A - CORRELATE OR MODIFY WITH EMOTIONS AND MOODS** The approaches summarized in class #III.A enables the modification to the content depending on the mood of the viewers. A good example is the LIVE project, done within an Austrian field trial during the Olympic Summer games in 2006. The idea was to build a system to help the director produce a digital bouquet of several output streams from a few hundred live streams and archive materials, covering a single event in real time. During the stream, the viewer can rate the content. Depending on these rates, the producer would change the broadcast content [83].

**CLASS N.LS - #III.B - FOCUS AND AIM OF COLLABORATION IS GAMING** The primary focus of approaches classified with N.LS - #III.B is on gaming. Consequently, the purpose of the collaborative and social features offered with these approaches is to play and win the game. For example the content format Tatort+ (# 21) was expanded by a multi-faceted interactive episode. The prologue started online one week before the Tatort episode was broadcast on television. The commissioner’s office was a main focus, and included different multimedia content that could be explored and shared by fans via PC or tablet. The epilogue was the main crime game and started online immediately after the Tatort episode was broadcast. The viewers could, collaboratively with friends or by themselves, follow as the protagonist collected evidence, visited scenes and solved the background story to the broadcasted Tatort episode.

**CLASS N.LS - #III.C - SOCIAL EXPERIENCE** Class #III.C summarizes approaches whose focus is on social experience, and how it correlates with broadcast content. Accordingly, the type of content (e.g. genre) and its narrative structure are barely relevant for the social and collaborative features. Approaches in this class are well known under the term social television. The idea of social television arose from the observation that people

---

\(^3\) https://messenger.yahoo.com/plugins/view/7551/
share, through social media, their experiences and opinions while watching television [11].

The correlation to the content is shown in the ways viewers discuss and exchange views about the broadcast content. One example of a social television system is CollaboraTV, which realized a unique interface for conventional TV content, providing synchronous and asynchronous text communications of people watching the same channel.

As already mentioned, social television is not the main focus of this thesis but classified as a segment of collaboration in broadcast content formats. Therefore, social TV systems are not discussed deeply, but are considered for the sake of completeness. A detailed and fundamental presentation and categorization of social television approaches can be found in [23].

2.3.2 Branched Narrative Space

<table>
<thead>
<tr>
<th>Class Identifier</th>
<th>Class Description</th>
<th>Classification*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class N.BS:</td>
<td>Branched Structure</td>
<td></td>
</tr>
<tr>
<td>Class N.BS - #I:</td>
<td>Approaches are linked to the content.</td>
<td>9, 10, 11, 12, 13, 15</td>
</tr>
<tr>
<td>Class N.BS - #I.A:</td>
<td>The CMC Link is realized by simple voting mechanisms via phone calls and SMS.</td>
<td>10, 12, 13</td>
</tr>
<tr>
<td>Class N.BS - #I.B:</td>
<td>The CMC Link is realized by correlating exploratoriums which are offered in the real world (# 9) and online in addition to the linear broadcast content format.</td>
<td>9, 16, 21, 22, 23, 24</td>
</tr>
</tbody>
</table>

Table 2: Class Description for Non Linear Content. (* Number # of Approach According to Figure 9 and Appendix B)

So far, the examined approaches have been social and collaborative TV systems built around existing broadcast content formats. Approaches that are classified in the following, assigned in Table 2, are particularly designed collaborative broadcast content formats which link collaborative activity to the program content & format.

**Class N.BS - #I.A - CMC Link is realized by simple voting mechanisms**

Approaches classified in this class are particularly designed broadcast content formats that are able to be modified collaboratively by the viewers, and simultaneously with the broadcast, by offering simple voting mechanisms via phone calls or SMS. One example is the broadcast content format Akvaario, broadcast by the Finnish TV channel YLE. It is automatically edited in real time from a large database of clips (approximately 5,000), according to the viewers’ interactions provided via phone [72].

**Class N.BS - #I.B - CMC Link is realized by correlating exploratorium**

Class N.BS - #I.B approaches expand the linear broadcast content format with additional worlds on the web and/or in the real world. More precisely, viewers modify the course of the content until the next broadcasting time. One example is the Swedish participation drama
“The truth about Marika”. The story is about a young woman who invites the viewers to help her search for her lost friend. The search took place all over Sweden in the real world and online as well.

2.4 CONCLUSIONS & IMPLICATIONS

The results of the classification and evaluation focusing on collaboration in broadcast content formats have revealed interesting peculiarities in current approaches. Below, the findings of the classification and evaluation are summarized. The correlations (the classes presented in the last section) between the categories and criteria are shown hierarchically from left to right, in correspondence with Figure 7.

2.4.1 General Observations

The most remarkable observations are the huge amount of proprietary approaches and the missing tool support. The term “huge amount” is not overstated: 24 approaches were analyzed and the list is far from being complete. Many different disciplines, like social television, CSCW, HCI, A/V industry, etc. agree on the topic of “direct integration of collaboration into broadcast content formats”, and of course it might not be possible to incorporate them all. However, it is remarkable that no standard libraries or patterns exist to design and develop intentional collaboration in broadcast content formats (not to mention collaborative distribution of content by object to object interaction of clients) or link this to the medium / course of the content. Even in the genres of social and interactive TV, only the approaches “Shape Shift TV” of Ursu et al. [80] and ”Enhancing Interactive Television News“ of Olsen et al. [67] define standard functionality to make linear content interactive over the full production lifecycle without regards to collaboration. In 2011, a new standard, called HbbTV, arose that defined standard functionality to design interactive TV. Functionality to integrate collaboration is missing as well.

So far, science and industry have perceived the trend to collaborative participation two different ways. On the one hand, science has focused on the development of TV platforms that additionally provide social interaction and collaboration between viewers, but hardly any attention was paid to program contents & formats. This resulted in platforms providing simple collaborative services for existing broadcast content formats, which are designed to view rather than participate in. On the other hand, industry has concentrated on the development of participatory broadcast content formats, which invite the audience to engage with and get part of the content by using parallel platforms like Telephone or Internet. In contrast to science, industry paid hardly any attention platforms. Participation is optimized for parallel platforms like Telephone or Internet, but industry set the focus on adapting the concepts of Apps and App Stores from mobile phones to television without considering participatory program formats.

Other noticeable observations are that correlation, as well as the analysis and application of social activity, is not being utilized to take the link one step further and modify the

---

4 http://www.conspirare.se/ (URL is idle)
broadcast content format by collaborative activity. Most social TV approaches still follow the paradigm of using the broadcast content format as a central element and social media to support communication of the content [11].

2.4.2 The Content Narrative Space

2.4.2.1 Popularity of Linear Narrative Space by Producers and Developers

In the time of interactive TV and collaboration on the web, and in the broadcasting area, linear narrative space is still prevailing. Only 6 of 24 studied approaches created non-linear narrative space, and 4 out of 24 studied approaches offered an exploratorium in addition to linear broadcast content. This is caused by the lack of standards for the design, assembly and usage of non-linear narratives. For instance, there are no standards to analyze activity in social networks directly linked to the course of the narration. Another reason is certainly high production costs (e.g. producing alternative scenes and endings).

2.4.2.2 Popularity of Non-Linear Narrative Space by Viewers / Participants

Contrary to producers and developers, viewers would much prefer non-linear narratives. This is shown by existing non-linear examples (#9 - #13), which were popular in practical experience. For example, the teenager drama “Sofia’s Diary”, where viewers could collaboratively interact via social network functionality, was produced for three seasons and adapted for six languages as of now (state: March 2015).

2.4.2.3 Collaborative Broadcast Content Formats are Missing

A broadcast content format, designed to be modified (maybe synchronized) by collaborative and social activity, is missing.

2.4.3 CMC Link of Collaborative Activity to Program Contents & Formats

2.4.3.1 CMC Link of Broadcast Content Format and Collaborative Activity over Time

In 2008, when the author started researching this topic, a significant occurrence between social and collaborative activity (communication, sharing, etc) and linear content was given (cf. [55], [14], [27], [63]) - except for “The truth about Marika” and “Sofia’s Diary”. The social and collaborative activity was added to existing content without adapting it at all. Over time, the union of the broadcast content format and the offered social and collaborative activity became popular. For instance, exploratoriums were developed in correlation to the broadcast content format (cf. “Tatort+5”, “DinaFoxx5” and “Schuld7”).

5 http://de.wikipedia.org/wiki/Tatort_%28Fernsehreihe%29
6 http://www.zdf.de/dina-foxx/dina-foxx-35058724.html
2.4.3.2 Connection between the categories “Non-linear narrative” and “Link Broadcast Content”

By definition, a connection between non-linear narrative and link to content must exist because it is not possible to design non-linear narrative without link to it. But the reverse is not necessarily valid, as approach #14 shows, because the broadcast of linear content can be adapted by using archived material. In the case of non-linear content, three observations can be seen:

- Simple voting mechanisms, like SMS and phone calls, are used.
- In case of more complex mechanisms (like blog entries), the collaboration is asynchronous between users and asynchronous with the broadcast content.
- Or an exploratorium, correlating to linear broadcast content, can be used.

The sole exception is the “Social Quiz Game”, which allows live participation via a social network app.

2.4.3.3 Missing of Standards to Link Collaboration and the Broadcast Content Format

Missing standards, firstly to link collaboration to the broadcast content format, and secondly, to analyze collaborative activity and connecting results to the medium, are remarkable. For “Sofia’s diary”, for example, social activity was both analyzed and entered into the content manually by the authors.

2.4.4 Scope of Collaboration

2.4.4.1 The Main Scope of Collaborative Activity for Linear Narratives

In linear narratives, the main scope of collaboration focuses first on social experiences, then on distributing and sharing content and its controls.

2.4.4.2 Social Experience as a Correlating and Modification Factor is on the Rise

Notwithstanding non-linear broadcast content formats #7 - #12 (which were developed around 2008 or earlier), social experience is the leading factor in content correlation and modification. Content correlation means that users can discuss it (as mentioned above) and modification means that controls are shared or the medium is annotated collaboratively. It is interesting to note that social experience is rarely used as a modification factor.

2.4.4.3 Limited Scope of Collaboration in General

As mentioned before, the main scope of collaboration is limited to distribution, social activity, enhancement, and voting procedures. More complex collaboration includes awareness of the participants to modify the medium and/or the course of the content. An interesting idea would be, for example, to add or change metadata that is collaboratively attached to the medium. Those metadata would then be interpreted by the collaborators player.
2.4.5 Key-Implications for the Reference Architecture

Therefore, conclusions can be drawn with regards to the necessity to develop a reference architecture and standard libraries to enable the development of collaborative broadcast systems for the consumption phase. This should be accompanied by the development of a social and collaborative broadcast content format. A reference architecture should define the following:

- Mechanisms to link collaborative activity to program contents & formats. For instance by the inclusion of metadata into broadcast technology to trigger collaborative activity.

- A producer and broadcaster platform which allows the implementation of mechanisms to link collaborative activity, broadcast technology and program contents & formats. For instance which supports the inclusion of metadata into A/V content.

- Consumer platforms which support collaborative activity on a television platform or on a second screen device.

- Prototypes of collaborative services which are suitable for television user platforms.

- Capturing and processing the outcome of collaborative activity in real time.

- Mechanisms to map collaborative activity to the course and characteristics of the program content (to enable modification to the program content).
3 USE CASE SCENARIOS AND REQUIREMENTS DEFINITION

“Technology and Social Media Have Brought Power Back to the People.”
— Mark McKinnon

The previous chapter lists findings and gaps identified while studying the field of collaboration in broadcast content formats. In this chapter, those findings and gaps are applied to create and present example scenarios and use cases. To demonstrate the varied scope of this field, story scenarios are developed and built, as described in the works of Cooper [26] and Carroll [22]. To emphasize the need of standards in this area, scenarios are created from the consumption and the production phase. To establish dedicated functional requirements, the story scenarios are then classified and abstracted into elaborate, advisable, and mandatory requirements, as well as general requirements for the four main domains of this thesis, Broadcast Technology as Medium, Program Contents & Formats, Collaborative Activity, and CMC Link.

3.1 TARGET GROUP DESCRIPTION

This section presents the target group description in the form of fictional persona descriptions. The target group consists of the target audience and of the producer.

3.1.1 Evaluation of the Target Audience

The term “target audience” is denoted as the (end-)consumer or the viewer of the broadcast television content format. The definition of the average consumer is deduced by defining requirements and analyzing Austrian statistics regarding these requirements. Generally, it can be said that the average consumer of collaboration in broadcast content formats fulfills the following requirements:

**Home Internet Access** Access to the Internet is crucial, since additional features may be downloaded and collaborators are connected via an underlying network. Plot #1 in Figure 10 shows that more than 80% of Austrian households have access to the Internet.

**Know-how in Computing** Basic know-how in the handling of a computer and its user interfaces is necessary. As plot #2 in Figure 10 shows, 86% - 98% of individuals, 16 to 54 years, old are experienced in the usage of computers.
3.1 Target Group Description

Usage of Mobile Devices In the case of second screen solutions, mobile devices (smartphone or tablet) having Internet access is advantageous. As plot #3 in Figure 10 illustrates, around 70% of individuals, depending on the age band, own and use mobile devices with Internet access. However, second screens are optional, so less weight is given to this requirement.

Experience with Social Networks Users should know how to use different interaction modalities and the basic concept of social networks. This is advantageous because social networks implement basic principles of computer supported collaborative work, like groups, buddy lists, user awareness and others. Plot #4 in Figure 10 shows that 64% of Austrian Internet users have a social network account. For under thirty-year-old, this figure is between 90% to 100%.

Willingness to Accept TV as a Potentially Lean-Forward Medium Television should be accepted as potentially active and lean-forward entertainment system.

Considering the previously presented requirements and statistics, the average consumer of collaboration in broadcast content formats can be defined as an individual, 16 to 54
years old. Other factors like family status, profession, education, housing situation, etc. are not relevant. Based on this, two persona descriptions were made: a female and a male consumer. They are presented in Table 4.

The persona description is based on [50] and consists of the following six elements:

**Outline**: The outline gives of the person’s name, age and occupation.

**General Description**: The general description briefly details the person’s life, including general interests, computer experience and knowledge, family status, and more detailed information about his/her profession.

**Goals**: This element describes the goals the person has achieved concerning to the collaborative activity.

**Features/Needs**: What does the person need to achieve her/his goals?

**User Tasks**: Lists the user tasks that are necessary to achieve his/her goals.

**Scenarios**: Gives information to the story scenarios in which the person is involved.

<table>
<thead>
<tr>
<th>Outline</th>
<th>General Description</th>
</tr>
</thead>
</table>
| **Name**: Thomas Richter  
**Age**: 22  
**Occupation**: Student | Thomas is a business science student in his second year. He is 22 years of age. Thomas likes watching movies, quizzes and sports. |

<table>
<thead>
<tr>
<th>Goals</th>
<th>Features/Needs</th>
</tr>
</thead>
</table>
| • Get connected to other viewers.  
• Use provided collaborative services.  
• Participate in shows anonymously.  
• Decide if he would rather participate or view. | • Media player that allows correlating services.  
• Repertoire of collaborative services including easy instructions.  
• Content formats correlating to the services. |

<table>
<thead>
<tr>
<th>User Tasks</th>
<th>Scenarios</th>
</tr>
</thead>
</table>
| • Watch content.  
• Subscribe.  
• Interact with correlating collaborative services.  
• Collaborate with other viewers. | #I.A: Watch.  
#I.B: Participate.  
#I.C: Switch Channel and Participate.  
#I.D: Participate in Group.  
#I.E: Terminate. |

Table 3: Description of Male Consumer/Viewer Persona
Sarah is a 35 year old social worker in a youth club near Linz (Austria). She passed the ECDL two years ago and likes surfing the web to watch IPTV and other streaming content. She is no expert in handling the computer. She has two children, ages 6 and 10.

**Goals**

- Choice of participating or just watching.
- Having fun.
- Modifying the course of the content.
- Participating (choice of participating anonymously or not.)

**Features/Needs**

- Media player that allows correlating services.
- Collaborative service that allow modifying the content.
- Hidden mapping between service and content.
- Awareness and some cases connection to other participants.
- Simplicity in handling.

**User Tasks**

- Watch content.
- Decide whether or not to participate.
- If applicable, decide if participation is anonymous and private.
- Subscribe.
- Interact with collaborative services.
- Collaborate with other viewers.

**Scenarios**

#II.A: Watching.
#II.B: Participating - Choice.
#II.C: Participating - Voting.
#II.D: Participating - Quiz.
#II.E: Participating - eLearning.
#II.F: Participate and Switch Channel.
#II.G: Terminate.

Table 4: Description of Female Consumer/Viewer Persona

3.1.2 Content Producer Description

These scenarios are situated in the production and post-production phase. Whoever is involved in the integration of collaborative activity into the broadcast content format, whether it is the producer, the director, the broadcaster or any technician /other staff member is at this stage of the thesis not relevant for the scenario description. Therefore at this point, producer or broadcaster involvement is assumed and a producer/broadcaster persona description is done. The producer persona description is straight, and is not based on any statistics. As done with the consumer persona, one female and one male persona are created.
3.2 CONSUMER STORY SCENARIOS

The following story scenarios use the previously created persona descriptions (cf. Table 3 and 4) in concrete, collaborative broadcast scenarios. The scenarios take place in an IPTV environment, broadcasting linear and non-linear television content. The creation of story scenarios helps to distinguish optional and mandatory functional requirements, which are necessary for modeling the reference architecture. Two story scenarios, from the consumer’s point of view, are described, which are divided into sub-scenarios.

3.2.1 Non-Narrative Modification Story Scenarios

**Scenario #1.a: Watching** Thomas Richter is watching a TV program on his PC via IPTV, using the player cTvC (collaborative TV client). Therefore, Thomas can choose between on demand IPTV content, as offered by the Austrian Broadcasting Corporation ORF at [http://iptv.orf.at/](http://iptv.orf.at/), TV channels broadcast via IPTV, like the German public-law channels ARD (Consortium of public-law Broadcasting Corporation of Federal Republic
### General Description

David is a producer at the film studio “Global Inc.”. The studio produces movies, multiple TV shows and program formats for TV channels. The TV program formats are sold as a package, together with collaborative elements that can be added to the content (e.g., a natural scientific quiz show).

### Goals

- Creating collaborative elements easily.
- Having a pool of existing collaborative elements that he can use and modify.
- Connecting the content to returning outcome of the collaborative activity.
- Using mechanisms for analyzing collaborative activity.
- Adding collaborative elements to the content.

### Features/Needs

- A pool of collaborative elements.
- Software providing tools to connect collaborative elements to the content.
- Software providing tools to connect the content to the outcome of the collaborative activity.
- Tools to analyze and quantify collaborative activity.

### User Tasks

- Prepare content for modifying collaborative activity.

### Scenarios

#IV: Adding modification collaborative participation to the TV content.

### Table 6: Description of Male Broadcaster/Producer Persona

of Germany) and ZDF (Second German Television). After zapping through the channels, Thomas decides to watch the TV show “DISCUSS” on channel CTC.

**Scenario #1.8: Participating**  
“DISCUSS” is a discussion show, where five invited guests discuss a current topic from the world of politics, economics, or chronicle. The guests are a combination of experts and aggrieved parties. In addition to passively watching, this program format offers collaborative applications for viewers to talk with each others. Thomas is notified of this additional opportunity by a pop up notification in the upper right-hand corner of his player. The notification indicates for Thomas to check the item “interactivity” in the menu bar of the player. Otherwise this item is inactive or hidden. Thomas becomes curious and clicks on the menu item, which unfolds a sub-menu that includes all available collaborative applications to this program format. He decides to participate in the chat and is asked for a screen name.
After filling in the form, the following text is displayed:

The following functions are offered:
* Open chat (open group, everybody is allowed to participate)
* Closed chat (build private groups)

Note: the chat activity has no modification to the live discussion in “DISCUSS”.
The service is available until 9:50 PM.
Enjoy!

After 45 minutes, the show draws to a close, and Thomas and all other viewers are notified about the end of the chat sessions by another pop up notification in the upper right-hand corner of the player. With the end of the show, its provided collaborative applications are not available anymore.

Scenario #1.c: Switch channel and participate  The topic of “DISCUSS” is about the wrongdoing of a politician, which is not Thomas's favorite topic. Therefore he switches to channel CTS broadcasting his favorite show, “CURIOSITY”, which is a natural scientific entertainment show. Live experiments are shown and three participants make a guess about its outcome.

By switching the channel, the correlating services change immediately. The chat from “DISCUSS” is removed from the sub-menu. Contrary to the discussion show, “CURIOSITY” provides a service for viewers to take part in guessing. The service is synchronized with the show, which means Thomas is asked about the outcome to experiment $\gamma$ when it is broadcast. He has the same amount of time to guess as the participants in the show do. Thomas can participate privately and anonymously, or he can participate in the competition, where his answer and the time he needed to get the right answer are sent to the broadcaster and handled. For this, Thomas has to fill in a form with his personal data before participating. Unfortunately, the show was already running when he switched to it, so he missed half of the experiments and is not allowed to participate in the competition anymore. Therefore he decides to participate anonymously. Experiment number 4, which is about the danger of instant coffee in the microwave, has just begun, so Thomas can choose one of the three given answers. He chooses answer number three, which is "Instant coffee explodes under several conditions". A few seconds later the experiment starts and time to guess is over. As the experiment shows, Thomas was right with his guess. Never put instant coffee immediately into the water after it has been microwaved.

Scenario #1.d: Participate in group  Since Thomas switched the channel early enough, the scientific show has just started. Therefore he can participate in the competition and has to fill in a form with his personal data (full name, screen name, e-mail address and phone number). Then, he chooses between single and multi player mode. In single player mode he can participate on his own. In multi player mode he can join a peer-to-peer network and be connected to two other participants within the network. He can search for people within the network, or get assigned to a partially engaged group randomly. The collaborative mechanisms are different, depending on the service in this case, a basic majority within the group of three defines the answer. In experiment number 4, as described above, Thomas's guess is again answer number three, but participants "Joe" and "Susanna" choose answer number one - therefore the group's answer is number one.
3.2 Consumer Story Scenarios

Each group member can see each other’s answers, and a simple chat functionality is given for discussions. In this case, they are not aware of other participating people or groups. If no bare majority is given, the group has to re-answer the question.

**Scenario #i.e: TERMINATE**  After an evening full of watching and engaging with IPTV, Thomas quits the cTVc player and turns off his computer.

3.2.2 Narrative Modification Story Scenarios

**Scenario #ii.a: watching**  Sarah Gruber is watching a TV program on her PC via IPTV, using the player cTVc (collaborative TV client). Thereby, Sarah can choose between on demand IPTV content, as offered by the Austrian Broadcasting Corporation ORF at [http://iptv.orf.at/](http://iptv.orf.at/) and TV channels broadcast via IPTV, like the German public-law channels ARD (Consortium of public-law Broadcasting Corporation of Federal Republic of Germany) and ZDF (Second German Television). After flipping through the channels, she decides to watch a TV show on channel CTC.

**Scenario #ii.b: participating - choice**  Sarah switches to a common music channel, which mainly broadcasts clip shows (sparse, moderated shows, which present one music clip after another). A notification in the upper right-hand corner tells her the availability of another collaborative service. When she clicks on it in the menu, a short info text tells her the following:

---

This service offers the possibility to design the course of tonight’s show ‘Movie Soundtracks’ from 7 PM to 10 PM. Suggest video-clips you want to see this evening in the show. Just fill in the artist and the title of any clip you want to see (no restrictions in genre), and our service tells you immediately when your suggestion will be aired. This service is available as long we have free slots in the show, or until 9 PM.

Enjoy!

---

In addition, she has the possibility to take a look at the present schedule. Sarah becomes curious and starts the service. It is built very simply, with only two text fields - one for the artists and one for the song name. Since she is a fan of Nick Cave, she tries his song ‘As I Sat Sadly By Her Side’. The service tells her that this song is not tagged as a movie soundtrack and that she has one more trial. She is also a fan of the movie ‘Magnolia’, so she suggested the song ‘Wise Up’, by Aimee Mann. The service then tells her that her suggestion will be aired at 7:21 PM. After this suggestion, the service turns inactive; it is limited to one suggestion per viewer due to the large number of users. So she listens to the music broadcast by CTCMusic and continues channel surfing.

**Scenario #ii.c: participating - voting**  Sarah’s daughter is watching her favorite show “ocidoki”. During the break, a spot is shown offering the possibility for viewers to vote for which children’s program for Christmas morning. At the start of the spot, a notification in the upper right-hand corner tells her, that she could vote for her favorite TV program to be aired on Christmas morning on channel CTCKidz. In the interactive menu in her player she opens the voting service which offers the following three choices: "The last Unicorn", 
"The Polar Express" and "The Grinch". The voting procedure is quite easy for children; she just pushes the radio button of her favorite show and confirms by clicking "OK". The player then transmits her choice to the channel's base. The voting feature is only available for this day.

**Scenario #11.d: Participating - Quiz**  
Channel CTC provides a new show called '1 against the Others - interactive'. It is an adaption of the show '1 against 100', which was broadcast in Austrian television in 2008. In the original program format, one nominee played a quiz against a group of 100 physically present participants. In the adapted show the group of participants is enlarged to an open group of participants who are not physically present. A notification in the upper right corner tells her of the availability of another collaborative service. When she clicks on it in the menu, a short info text tells her the following:

*This service offers the possibility to participate in an open group and play against a nominee who is live in studio. To subscribe for the group, please fill in the presented form, which desires a username, gender, age, and e-Mail address. You can leave the group at any time and sign up lasts until 8:45 PM.*

*Enjoy it!*

Sarah decides to participate and fills in her username, gender, age, and e-Mail address. After confirming, she immediately joins the game with the next question coming up. In the live show, a statistical analysis of the home participated users is shown constantly. Since the groups are open, this statistic changes continually. When the next question comes up in Sarah's player, the already running collaborative service extracts the next trigger from the received content. A window pops up, showing possible answers to the question: 'What is the first name of Nico Rosberg's (Finnish Formula-1 pilot) father? Is it a) Janne, b) Keke, or c) Mika'. Sarah has 20 seconds to decide. After choosing an answer, the service says thanks to Sarah and goes inactive. The nominee can also use his jokers. Depending on the joker, one or two home participants are chosen randomly (or something similar) and can be asked by the nominee. In the case of 'asking all', all active home participants can give their answers. If Sarah doesn't want to participate anymore, she can sign out with another click. If she switches the channel, she is signed out automatically.

**Scenario #11.e: Participating and switch channel**  
Sarah is watching a show offering a collaborative service. Now, she is tired of participating and wants to sign out. In most cases, clicking the sign out button is enough (depending on the show and the commitment agreed to before participating). Also if she switches the channel she is signed out automatically by the collaborative service.

**Scenario #11.f: Terminate**  
After an evening full of watching and engaging with IPTV, Sarah quits the cTVc player and turns off her computer.
3.2.3 Classification of Consumer Scenarios

The aforementioned consumer scenarios can be classified into passive and participatory content as Table 7 shows. The participatory content can then be further divided based on whether the collaborative activity will cause a narrative modification or not and even further divided based on whether the collaborative activity is synchronized and done in a group or privately. Please note that private participation does not exclude collaborative activity. For example, in a voting scenario each participant votes privately, although the act of voting (where the majority counts) is collaborative.

<table>
<thead>
<tr>
<th>Participatory</th>
<th>Asynchronous</th>
<th>Synchronous</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Private</td>
<td>Group</td>
</tr>
<tr>
<td>Non-Narrative Modification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Narrative Modification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Passive</td>
<td>Watching</td>
<td></td>
</tr>
<tr>
<td>Miscellaneous Activity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switch Channel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Classification of Presented Consumer Story Scenarios According to the Activity

The grayed out sections are not relevant and therefore no scenarios have been created for them. They are beyond the focus of this thesis. These include watching TV asynchronously, group watching scenarios (including collaborative control of the contents state e.g., play, pause, switch channel, etc.), or asynchronous collaborative activity as already realized in well-known voting scenarios.

The most striking feature is the difference between participatory modification and non-modification scenarios, regardless of private or group participation. Below, the difference between synchronous non-modification and modification scenarios is highlighted, with scenarios for when the collaborative activity is done both privately or in a group.

3.2.3.1 Non-Narrative Modification Scenarios

Non-narrative modification participatory scenarios describe the proposition of collaborative services correlating with characteristics of the broadcast medium and/or course and characteristics of the program content without modifying it. As mentioned above, the focus is on synchronized scenarios.

Figure 11 shows a conceptual process illustration of an abstract example scenario modeling the previously listed story scenarios #I.B - #I.D. The illustration provides (identify/enable/disable) non-modification collaborative services correlated and synchronized firstly...
to characteristics of the medium (course #(1) in Figure 11) and secondly to the course and characteristics of the content (course #(2) in Figure 11). To enable this, triggers in the medium and/or content must be defined to identify, synchronize, enable and disable provided collaborative services. In the following, possible triggers for course #(1) and #(2) of the process illustration in Figure 11 are demonstrated.

**Course #(1) — Trigger Embedded in the Broadcast Medium**

Course #(1) of the process illustration in Figure 11 illustrates the link of broadcast technology as medium to collaborative activity. Triggers could be timestamps, specific frames (as shown in Figure 11), and packets or metadata.
3.2 Consumer story scenarios

**Timestamp**: As previously defined, timestamps trigger the activation or deactivation of certain collaborative services (cf. Figure 12b)).

**Frame, Packet**: As previously specified, frames or packages of the medium trigger the activation or deactivation of certain collaborative services (cf. Figure 12c)).

**Metadata**: Embedding metadata into the medium (cf. Figure 12a)), which are extracted and analyzed by the consumer’s video players. Metadata include, for instance, the name and id of a collaborative service as well, as its initial properties and contents (e.g., questions and answers, provided features, etc.). The following example demonstrates how metadata could be used as a trigger. It calls a previously installed quiz application and inputs a new question and answer possibilities to it.

```xml
< .... >
<name> Quiz Service </name>
<id> 01_quiz </id>
<question>
  <id> 01 </id>
  <content> What is the first name of Nico Rosberg’s (Finnish Formula-1 pilot) father? </content>
  <answers> a) Janne, b) Keke, c) Mika </answers>
  <solution> b </solution>
</question>
<question>
  <id> 02 </id>
  ...
</question>
<....>
```

The metadata tag *name* refers to the name of the collaborative service, *id* to the unique identifier of the service or the elements of the service, *question* refers to the element of the question, *content* to the content of the element, *answers* refers to given answers, and *solution* to the correct answer.

Timestamps, frames and packets are suitable to trigger certain collaborative activity but they are not suitable to commit any properties (like a unique identifier) or further conditions. Thus, in the following examples metadata are used as the main trigger.

**Course #(2) - trigger embedded in the broadcast content**  
Course #(2) of the process illustration in Figure 11 illustrates the link of broadcast content & format to collaborative activity. The following lists exemplary triggers embedded in the content.

**Scene**: A previously specified collaborative service is enabled by playing a labeled scene (cf. Figure 13a)).

**Object or Group of Objects**: A previously labeled object or group of objects enables or disables collaborative services #2 and #3 (cf. Figure 13b)).

**Genre**: Collaborative services #4 and #5 provide content for the "game show" genre (cf. Figure 13c)).

**Problems & Vision**  
Taking the previously defined correlation as given, the problems with non-modification collaborative activity can be manifold. Providing correlated collaborative services to content is not new, but has been mainly realized in the form of
Web platforms that are thematically related to the content. For example the Web platform http://www.heldenvonmorgen.orf.at\textsuperscript{1} is related to the Austrian singer casting show “Helden von Morgen”, and includes additional videos and a chat-room. These asynchronous and content-correlating scenarios by using parallel platforms are succeeding. The main problem with using parallel platforms is the missing synchronization, which must be handled manually. Take the afore mentioned casting show “Helden von Morgen” for example. In addition to the listed features, the show provides extra information on a Web platform, including a voting application. Typically, the voting phase is open for the last 10 minutes of the show, during which time the voting must be enabled and the results handled (entered into the Web platform, etc.) by hand. Of course, this scenario is simple and changes are not very time consuming but in the case of more complex collaborative scenarios, synchronization is an important factor.

The vision of this thesis is to combine this existing thematic connection of additional platforms into a technical one. To realize this vision, the following elements must be provided:

- A player which receives, analyzes, and processes the broadcast while also supporting collaborative activity.
- Analysis of the broadcast medium, including time stamps, frame types, formats, properties (e.g., genre, language, etc.), and extraction and interpretation of metadata.
- Analysis of the broadcast content for scenes, objects, or genre.
- Triggers for collaborative activity.
- Security and privacy policies.
- Error protection.

\textsuperscript{1} Idle (1-Apr-2015)
3.2.3.2 Content Modification Scenarios

Content modification scenarios describe collaborative services that exert modification to the course and characteristics of the broadcast content format (e.g., offering a voting application for a music or casting show). As mentioned in Chapter 2, modification implies correlation, since modifying the broadcast content format demands some kind of correlation to it - whether the service is designed for a special genre or, more specifically, for certain types of metadata. As mentioned above, the focus here is on synchronized scenarios.

Figure 14 shows a flowchart of an abstract example scenario modeling the previously listed story scenarios #II.B - #II.E. The flowchart provides (identify/enable/disable) modifying collaborative services correlated and synchronized to characteristics of the broadcast medium (course #1 in Figure 14) and/or the course and characteristics of the program content (course #2 in Figure 14). The modification of medium and content happens on the collaborator’s (client) side - see course #3 and #4 in Figure 14. Modifications are made on the medium and/or content received by the involved collaborators. It also happens on the broadcaster’s (server) side - as shown in course #5 and #6 in Figure 14. Modifications are done at the root before the content format is broadcasted, which requires a back channel to the broadcaster. The modifications are apparent to all viewers and participants.

To establish this, three requirements must be given:

1. **Triggers in medium and/or content** must be defined in order to identify, synchronize, enable, and disable provided collaborative services and to access and modify the medium and/or content. See description above.

2. **Triggers in collaborative activity** must be defined corresponding to those in medium and/or content, to identify state, level or content of collaborative activity.

3. **Link** between the triggers on both sides.
Triggers for course #1 and #2 in Figure 14 were already demonstrated previously. In the following, possible triggers for course #3, #4/#6 and #5/#7 as well as the link are presented.

Course #3 - Triggers Included in the Collaboration

Course #3 of the process illustration in Figure 14 shows the modification of the broadcast technology as medium or the broadcast content & format triggered by the collaborative outcome. Following is a list of examples of triggers embedded in the collaborative activity.

**Topic of Collaboration:** Topic and questions rising in the collaboration.

**Scope of Collaboration:** Whether this is entertainment or a simple social experience.

**Output of Collaboration:** Collaborative solution of a task; the overall opinion of a discussion or similar outputs.

Course #4/#6 - Triggers Embedded in the Broadcast Medium

Course #4/#6 of the process illustration in Figure 14 shows the modification of the broadcast technology as medium on the consumer's/client's/viewer's/collaborator's side (course #4) and on the broadcaster's/producer's side (course #6). Following is a list of examples of triggers embedded in the medium.

**Frame, Packet:** Previously specified frames or packages of the medium trigger the activation or deactivation of certain collaborative services (cf. Figure 15a-2)).

**Metadata:** Embedding metadata into the medium (cf. Figure 15a-1)), which are extracted and analyzed by the consumer's video players. Metadata include, for instance, the name and id of a collaborative service, as well as its initial properties and contents (e.g.: questions and answers, provided features, etc.). See previous example (Course #1)) for further information.

Timestamps, frames and packets are suitable to trigger certain collaborative activity but they are not suitable to input any properties (like a unique identifier) or further conditions. Thus, in the following examples metadata are used as the main trigger.
Course #(5)/#(7) - Triggers Embedded in the Broadcast Content

Course #(5)/#(7) of the process illustration in Figure 14 shows the modification of the broadcast content & format on the consumer's/client's/viewer's/collaborator's side (course #5) and on the broadcaster's/producer's side (course #7). Following is a list of examples of triggers embedded in the content.

Object: Replace an object of a scene or change its properties, e.g., size, color, etc. (cf. Figure 15b-2)

Scene: Rearrange the objects in the scene; change properties of the scene (cf. Figure 15b-1)

Link Triggers in Collaborative Activity to the Broadcast Medium and Program Content

To provide interactivity, it is necessary to link the triggers in medium/content with those in collaborative activity. This can happen, for example, by processing received metadata, which are embedded in the broadcast content format, on the consumer side, or by triggering coordinated events on the consumer side when specified frames are decoded.

Problems & Vision

Previous realizations of viewers modifying content resulted in telephone voting scenarios or additional platforms on the Internet. Interactivity, in turn, mostly resulted in adding interactive features to linear content (as the Multimedia Home Platform (MHP)\textsuperscript{2} [33] does). This basically failed, due to the lack of standards covering multiple platforms, unsuitable content (linear content that is made for “watching” does not agree with external interactivity), insufficient user interfaces, and so on.

The vision is to extend the idea of correlation to modification of the content to reach a certain level of collaboration. Certainly, broadcast content formats cannot be modified on the same level as gaming content, where the player can more or less create his/her own story flow. However, it is possible for viewers to get involved into the story flow without traveling into any producers studio, showing their face, and giving up their anonymity. This is the main critique of reality, game, and casting shows, like YouTube and others. For this, it is necessary to make the content suit social and collaborative interactivity.

However, to realize this vision, the following elements must be provided:

- A player who receives, analyzes and processes the broadcast content format while also supporting collaborative activity.
- Analysis of the broadcast medium including time stamps, frame type, format, properties (e.g., genre, language, etc.), extraction and interpretation of metadata, or similar.
- Analysis of the broadcast content including scenes, objects or genre.
- Triggers of collaborative activity.
- A pool of predefined collaborative services that allows for modifying the broadcast content format.
- A non-linear collaborative broadcast content format.

\textsuperscript{2} \url{http://www.mhp.org}
3.3 CONTENT PRODUCTION & BROADCAST STATION SCENARIOS

The following story scenarios use the previously created producer/broadcaster persona (cf. Table 5 and 6) in concrete scenarios. The broadcast proceeds via IPTV, broadcasting linear and non-linear television content. The creation of story scenarios helps by distinguishing between optional and mandatory functional requirements, which are necessary for modeling the implemented reference architecture. Two story scenarios, representing the producer/broadcaster point of view, are described and divided into several sub-scenarios.

3.3.1 Non-Narrative Modification Scenarios

3.3.1.1 Scenario #III: Provide Correlation Collaborative Activity

Before broadcasting the content or selling it to broadcasters, producer Maria Eder decides to create a participatory version of the linear natural scientific entertainment show "CURIOSITY". To do so, she has to link to and adapt the pre-assembled prototype of a collaborative quiz application. This application is located on the consumer side and receives extracted characteristics (like time slots or metadata) from the pertinent content. For example, the question process waits for metadata, including the question, time slot (defines the active period of the question), possible wrong answers, and the correct answer. The link of the broadcast content format "CURIOSITY" to the collaborative application is done by including metadata into the medium. Therefore, Maria has to add the following metadata:

1. Signature of the collaborative application, which is "001_quiz".
2. Questions and answers, added at the beginning of the time slot, including the time stamp for inactivity.

An editor is provided to easily add these metadata to the content.

3.3.2 Narrative Modification Scenarios

3.3.2.1 Scenario #IV: Provide Modifying Collaborative Activity

Before broadcasting the content or selling it to broadcasters, producer David Brunner decides to create a participatory version of the linear show “1 against 100”. To do so, he has to link to and adapt the pre-assembled prototype of a collaborative quiz application. The client part of the application is located on the consumer side, the server part on the broadcaster’s side. Both parts provide necessary collaborative mechanisms for the participants to collaborate. In addition, it receives extracted characteristics (like time
3.3 Content Production & Broadcast Station Scenarios

slots or metadata) from the pertinent content. For example, the question process receives metadata, including the question, time slot (defines the active period of the question), possible wrong answers, and the correct answer. The link of the broadcast content format “1 against the Others - interactive” to the collaborative application is done by including metadata into the medium. Therefore, David has to add the following metadata to the content:

1. Signature of the collaborative application, which is ”002_show“.
2. Questions and answers, added at the beginning of the time slot, including the time stamp for inactivity.
3. Identification of the server part of the application.

An editor is provided to easily add these metadata to the content. To provide real participation, with feedback, the client communicates directly with the server, and the server sends the id of the chosen answer back. The software automatically synchronizes the adaptation of the medium and/or content and the collaborative activity, so there is no need for the broadcaster to interfere.

3.3.3 Classification of Content Production & Broadcast Station Scenarios

The mentioned producer/broadcaster scenarios can be classified with regards to how the broadcast medium and/or content are prepared, whether they are enhanced by trigger precedent or their characteristics are modified during the broadcast. They can also be classified with regards to the broadcast medium’s and/or content’s purpose whether they are enhanced for non-modification/modification, asynchronous/synchronous collaborative activity. An overview of this classification is shown in Table 8.

<table>
<thead>
<tr>
<th>Enhance Medium and Content</th>
<th>Asynchronous</th>
<th>Synchronous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modification</td>
<td>#IV</td>
<td>#III</td>
</tr>
<tr>
<td>Non-Modification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modify Medium and Content</th>
<th>Asynchronous</th>
<th>Synchronous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modification</td>
<td>#IV</td>
<td>#III</td>
</tr>
<tr>
<td>Non-Modification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Classification of Presented Broadcaster/Producer Story Scenarios

The grayed out sections are not relevant, as they are beyond the focus of this thesis. Therefore, no scenarios have been created for them. These include providing asynchronous collaborative activity via additional platforms, and changing the content concerning to asynchronous collaborative activity, and synchronized enhancement of the medium and/or content by a third party.
3.3.3.1 Integration of Synchronized, Non-Narrative Modification Collaborative Activity Into Broadcast Content Formats

To integrate synchronized non-narrative modification collaborative activity, certain steps need to be taken, as shown in Figure 16.

![Figure 16: Integrate Collaboration Into the Broadcast Medium (a) and/or Content (b)](image)

1. Select the collaborative service(s) to provide (including unique identifier) (step (1) in Figure 16a and b)).

2. Define conditions, under which collaborative services are triggered → trigger could be frames or metadata (cf. step (2) in Figure 16a)), or objects and scenes (cf. step (2) in Figure 16b))

3. Link the unique identifier of a certain collaborative service to a certain trigger (step (3) in Figure 16a and b)).

3.3.3.2 Integration of Synchronized, Narrative Modification Collaborative Activity Into Broadcast Content Formats

To integrate synchronized, narrative modification collaborative activity, certain steps need to be taken, as shown in Figure 17.

1. Provide collaborative activity:
   a) Select the collaborative service(s) to provide (including unique identifier) (step (1) in Figure 16a and b)).
   b) Define conditions, under which collaborative services are triggered → trigger could be frames or metadata (cf. step (2) in Figure 16a)), or objects and scenes (cf. step (2) in Figure 16b))
   c) Link the unique identifier of a certain collaborative service to a certain trigger (step (3) in Figure 16a and b)).
2. Provide modification:

a) Define and add triggers (as described in the consumer scenario) for the broadcast medium and/or content (step (1) in Figure 17a) and b)).

b) Define and add triggers for the collaborative activity (step (2) in Figure 17a) and b)).

c) Link triggers in collaborative activity to those included in the broadcast medium and/or content (step (3) in Figure 17a) and b)).

Problems & vision  
The enhancement of MPEG with interactive components is not new. For instance, it has already been realized in the Multimedia Home Platform (MHP)\(^3\) \([33]\), MPEG-4\(^4\) \([34]\), and by Kim et al to design an interactive augmented broadcasting system \([46]\). The challenge is in providing large-scale collaborative activity and synchronization. This includes analyzing collaborative activity and changing characteristics of the medium and the content.

The vision of this thesis is to provide a reference architecture that enables the enhancement of medium and content and provides the definition of, and link to, the triggers. In addition, the run-time environments allow synchronized modifications on the medium and content. However, to realize this vision, the following elements must be provided:

- A collection of predefined collaborative services, managed in a database.
- Techniques to integrate collaborative activity into the broadcast content format.
- Techniques to define and add triggers to the broadcast content format.
- Techniques to define and add triggers to collaborative activity.
- Techniques to link triggers of the broadcast content format to those of collaboration.
- Techniques to synchronize the broadcast content format in real time.

---

\(^3\) [http://www.mhp.org](http://www.mhp.org)

\(^4\) [http://mpeg.chiariglione.org/standards/mpeg-4](http://mpeg.chiariglione.org/standards/mpeg-4)
3.3.3.3 Modification of the Broadcast Content Formats

Synchronized modification of the medium and/or the content happens automatically, without user input. Modifications will be done with regard to the previously defined triggers and links, as Table 9 demonstrates.

The field "ID" in Table 9 refers to a unique identification of each tag, whereas the ID of the link is a combination of the involved triggers IDs. The "anchor" identifies the object to which this tag belongs. The object can be a frame, object in a scene, object group, scene, timestamp, etc., as described in the consumer scenarios. The "property" specifies the property that is either modified (mostly in medium or content, in this example the broadcast content format), or measured (mostly in collaboration, in this example the collaborative outcome). Modified properties are for example "obj_color" as the color of object X in scene 21 or "obj_rel_position" as the relative position of object X in scene 21. Measured properties are for instance the collaborative answer of a question or task, or the quantification of the collaborative activity. The properties in the link define the conditions at which a certain property is changed as well as the result to which the certain property is modified if the condition is true or false.

Table 9: Example for the Link in Modifying Synchronized Collaboration

<table>
<thead>
<tr>
<th>ID</th>
<th>Trigger (Collaborative Activity)</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>#823947</td>
<td>#87123</td>
<td>#823947_87123</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Anchor</th>
<th>#823947</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object X in Scene 21</td>
<td>Service 001_quiz_7654</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property #1</th>
<th>obj_color</th>
<th>answer1</th>
<th>IF 87123.answer == 'Keke' 823947.obj_color = '#007A00' ELSE 823947.obj_color = '#7A0000'</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Property #2</th>
<th>obj_rel_position</th>
<th>quantification</th>
<th>IF 87123.quantification &gt; MAX 823947.obj_rel_position = '36,138'</th>
</tr>
</thead>
</table>

3.3.3.3 Modification of the Broadcast Content Formats

The main challenge is the interplay of all components in real-time: the analysis of all given conditions at a certain frame, timestamp, or similar, the parallel analysis of collaborative activity, and the synchronized modification of the medium and content during its broadcast.

To realize this vision, the following components are necessary:

- Access to the broadcast medium during its broadcast.
- Access to the program content during its broadcast.
• Techniques to identify, observe and process collaborative activity.
• Techniques to analyze and execute triggers.

3.4 REQUIREMENTS FOR LINKING BROADCAST TECHNOLOGY AS MEDIUM, PROGRAM CONTENT & FORMATS AND COLLABORATIVE ACTIVITY

Drawing from the previously designed persona and scenarios, general requirements can be made for the four basic building blocks defined in the Introduction: Broadcast Technology as Medium, Program Content, Collaborative Activity, and CMC Link.

3.4.1 Broadcast Technology as Medium

As defined in the Introduction, the broadcast medium acts as a transmitter, transmitting the content to its target audience (viewer). Broadcast Medium (or Medium) denotes the technical realization and representation of the content. Well-known data formats are MPEG-2 and MPEG-4; these are used as the digital video broadcasting (DVB) standard in Europe.

To realize these previously-defined application scenarios, the following requirements need to be fulfilled.

3.4.1.1 Integration into the Austrian Media Technology and Television Landscape

The solution that will be implemented by this thesis should easily assimilate into the Austrian media and television landscape. More precisely, the focus is not on the development of another proprietary solution for another proprietary platform but on a scalable solution that can be integrated into the current Austrian media and television landscape.

As of 2013, 73% of Austrian TV households uses digital satellite or digital cable receivers (cf. Figure 18). Using Internet and mobile devices to watch television is not yet as popular. The Austrian mean-target-audience (as described with the persona descriptions) uses a television set composed of a monitor, a set-top box (maybe integrated into the monitor), and either DVB satellite (DVB-S), cable (DVB-C), or terrestrial (DVB-T) transmission and
receiving. As of 2013, the Austrian television landscape provides 95 German-language (using a DVB-S receiver) television channels, broadcasting mostly linear content. Because of this, the following standards need to be considered: MPEG as a DVB standard combined with a consumer platform running on an up-to-date digital set-top box (android technology). Around 70% of Internet users also use mobile devices, so 'second screen' solutions should also be considered.

3.4.1.2 The Broadcast Technology as Coordinator

To enable collaborative activity, a coordinator is necessary, and the broadcast medium should serve as that coordinator. To do so, triggers must be integrated into the medium (producer platform), and received and executed by the consumer platform.

3.4.1.3 Suitable Broadcast Standards

Regarding with the described Austrian media technology and television landscape, the following broadcast medium standards are suitable:

**MPEG-2** The ITU standard, defines "the generic coding of moving pictures and associated audio information" [41], including techniques for lossy video and audio compression. MPEG-2 is a commonly used and well accepted standard, for instance it is used for the transmission of digital television broadcasts by terrestrial, cable, or satellite in Europe. The MPEG-2 standard (ISO/IEC 13818) consists of 11 parts, describing, amongst others, the program and the transport stream (13818-1), the video compression (13818-2), and the audio compression (13818-3 and 13818-7). Since MPEG-2 is standard for TV in Europe, the criteria of integrability into the Austrian media and television landscape is complied. It complies with the second issue (using the medium as coordinator) as well, since it is possible to include a data stream into the transport stream.

**MPEG-4** The ISO/IEC standard MPEG-4, developed by the Moving Pictures Experts Group (MPEG). The MPEG-4 standard (ISO/IEC 14496) consists of 30 parts (effective 2013).

MPEG-4 combines the fields of production, distribution, and content access and prioritizes [47]

- Digital Television.
- Interactive Graphics Applications (synthetic content).
- Interactive Multimedia (World Wide Web, distribution of and access to content).

The main concept of MPEG-4 is the composition of media objects into audiovisual MPEG-4 scenes. Several audio and visual objects in a scene are compounded into more complex media objects, like the visual object of a barking dog and its corresponding sound. Media objects may have allocated synchronized elementary streams to get access to streaming data and are identified by the media object descriptor. This concept allows the user to interact with the scene or, more precisely, to interact with

---

5 http://mediaresearch.orf.at/index2.htm?fernsehen/fernsehen_ma.htm
6 http://www.statistik.at/web_de/presse/073631
single media objects of a scene.

The MPEG-4 systems, including the MPEG reference software, are not adequate means to realize the idea and the use-case scenarios of this thesis. The reference software allows the development of interactive content as well as the development of applets and programs, which are downloaded with the content. In theory, this concept supports the idea of linking broadcast technology as medium, program content & formats and collaborative activity. MPEG-4 group provided an SVN repository\(^7\), so everybody could have read-only access to the reference software. The author tried to access and run the reference software, including the MPEG-J framework, in 2011. At this time, it was not able to check out, generate, and compile an executable version of the software. Even members of the MPEG-4 group could not confirm if the reference software of the MPEG-J framework would work at this point in time. Although MPEG-4 fulfills the presented requirements in theory, it is for this reason MPEG-4 cannot be a solution for the presented concept.

**DVB-MHP** The multimedia home platform DVB-MHP expired in 2011\(^8\), but was relevant during my research work, so it will be briefly mentioned.

The DVB Multimedia Home Platform (DVB-MHP) is an open middleware system standard designed by the DVB project for interactive digital television. MHP enables the reception and execution of interactive, Java-based applications on a television set. Interactive TV applications are delivered, with audio and video streams, over the broadcasted channel and return via the IP [32].

MHP would fulfill the previously defined general criteria; it is a DVB standard and therefore supports MPEG-2 transports. Field trials in Austria and Finland have also shown that MHP integrates easily into the existing TV landscape. It also it supports the development of interactive applications for TV; from here one can imply that the support and functionality can extend to collaborative applications. One deficit for our purpose is the focus on adding interactivity to linear content. Since DVB-MHP has been defunct in Austria since 2011, this solution had to be dropped.

**HbbTV** The Hybrid Broadcast Broadband TV standard is defined in the ETSI TS 102 796 (currently in version 1.2.1 2012-11).

The standard addresses two types of application: [31]

- Broadcast-independent application, which is not associated with any broadcast service. This type of application is downloaded and accesses all of its data via broadband.
- Broadcast-related application, which is associated with one or more broadcast services or events. This type of application is downloaded and accesses all of its data via broadcast or broadband. The application may be launched automatically or upon user request.

As the name of the standard suggests, the system consists of the broadband and broadcast networks and a hybrid terminal. The hybrid terminal receives standard A/V content, application data, and application signaling information via the broadcast

---

\(^7\) [http://wg11.sc29.org/svn/repos](http://wg11.sc29.org/svn/repos)

network. If a broadband connection also exists, application data and more complex non-linear A/V content can be received via this connection. The terminal receives, via the Broadcast Interface, the application information table, linear A/V content, application data, and stream events [31].

The runtime environment consists of a browser and an application manager and can be seen as a very abstract component in which the interactive application is presented and executed. The application manager evaluates the application information table and controls the lifecycle for an interactive application. The browser is responsible for presenting and executing an interactive applications [31].

It is worth noting that the application range displays more information than the current broadcast A/V content. Another interesting idea would be to provide more complex applications to for example combine (non-)linear A/V video broadcast content [31].

The HbbTV standard includes the tools necessary to realize the presented ideas. It merges broadcast and broadband network. It provides optional execution of interactive applications that are related to the broadcast content format. The current quota of applications is limited in order to provide additional information to the received broadcast content format not sure this part is necessary. When this thesis was started in 2008, the HbbTV standard was not available; the first version was published in June 2010.

The decision was made to build a framework around the MPEG-2 standard, which realizes both the consumer and producer/broadcaster requirements. An open issue and future work is clearly the usage of HbbTV.

### 3.4.2 Broadcast Content Formats

One topic of this thesis is the development of a (socially) collaborative broadcast content format, instead of simply combining collaborative functionality with a linear program format. Therefore, the broadcast content format must satisfy the following conditions:

#### 3.4.2.1 Narration and Storytelling

The term *narration* defines the structure behind the *story* (cf. Chapter 2). For the development of a collaborative broadcast content format, a non-linear narration structure is required. Which type of non-linear narration works best, and if it is feasible (technically and usably) to design broadcast content beyond branched structures, cannot be answered at this time. Collaborative actions and the process of their outcomes must be planned from the very beginning and also considered in the script.

#### 3.4.2.2 Coordination of Storyline and Collaborative Activity

In addition to narrative, a collaborative broadcast content format needs a collaborative storyline and structure, coordinated with the narration and storyline. More precisely, the

---

9 Consists of broadcast technology as medium and program contents and formats.

Appendix A summarizes all definition of terms as used in this thesis.
medium and the content works as coordinators between the collaborators. The collaborative storyline refers to the story, including the collaboration topic and action structure as extensions to the main story. It is also necessary to plan the timing for the collaboration into the storyline, including, for example, buffers to synchronize the collaborators or to bypass broadcast delays.

3.4.2.3 Tagging based Content Trigger
Content must be able to be tagged with triggers, which in turn cause certain events. Possible medium and content tags to anchor triggers on are frames, timestamps, persons, actions, locations, and moments. To enable this, triggers must be integrated into the medium's transport stream.

3.4.3 Collaborative Activity and Consumer
3.4.3.1 Coordination Model
A collaborative broadcast content format also needs a coordination model. Depending on the collaborative activity and narration structure, patterns of coordination models can be defined in advance. The coordination model should define:

1. The collaborating entities, which is the target audience.
2. The coordinating medium, which is the broadcast content format.
3. The coordination laws, which must be defined during the scripting phase.

3.4.3.2 Timing and Complexity
As mentioned, the sequence of the collaborative actions must be planned from the very beginning of the scripting phase, and must fit within the storyline. The most important factors in the sequence are the timing and complexity of the collaborative activity. Because the broadcast, and therefore the collaborative activity as well, is time sensitive, the collaborative activity must be intuitive, and the complexity adjusted. Collaboration in broadcast content formats cannot have the same complexity as a group editor or a decision support system.

3.4.3.3 Consumer Client
A consumer client is needed to provide the target audience with the collaborative features dependent on the trigger.

3.4.4 Linking Collaborative Activity to Broadcast Technology as Medium and Program Content & Formats
The trigger embedded in the medium and content are linked to the collaborative activity. For this, two tools are required:

1. Producer client, which enables the enhancement of triggers anchored to tags into the broadcast medium and content before broadcast.
2. Consumer client, which enables the reception, analysis, and processing of the enhanced broadcast medium and content.

3.5 TECHNICAL REQUIREMENTS TO IMPLEMENT THE DIRECT INTEGRATION OF COLLABORATION INTO BROADCAST CONTENT FORMATS

From the story scenarios and the general requirements, three main modules emerged, the Consumer Client Module, the Broadcaster Server Module, and the Broadcaster/Producer Module, to prepare and enhance the medium and content before it is broadcast. Mandatory and advisable functional requirements were created for each module based on the abstraction and classification of the story scenarios. Mandatory functional requirements describe functionality that must be provided by the reference architecture and therefore define the boundary of the implementation part of this thesis. Advisable functional requirements describe necessary functionality that is off the scope of this thesis.

As illustrated in Chapter 1 and shown in Figure 19, the reference architecture will contain three main modules, namely Consumer Client Module, Broadcaster Server Module and Broadcaster/Producer Module. Below, each module and its requirements are presented.

3.5.1 Consumer Client Module

As its name implies, this module is situated on the consumer side. Its main purposes are handling received video content (receive, decompress, analyze, and display), providing collaborative services, and, if necessary, transmitting the output of collaborative activity to the broadcaster. Figure 20 gives an overview of the mandatory and advisable functional requirements as well as their interaction(s).
3.5 Technical Requirements

3.5.1.1 Mandatory functional requirements

**Player:** Receive, analyze, and display the broadcast technology as medium and the program content.

1. *Receive Broadcast Content Format:* Receive and decode video content.

2. *Analyze Broadcast Content Format:* Analyze the received and decoded medium and content, with regard to enhanced collaborative services.

3. *Play Content:* Display received and decoded content.

4. *Notification for participation:* Notify the consumer about possible collaborative services.

5. *Sub-menu for collaborative services:* Provide an active sub-menu in case of available collaborative services.

**Analysis of the Broadcast Medium and Content:** Provide functionality to analyze the broadcast medium and content concerning the embedded collaborative functionality.

1. *Medium:* Analyze the medium, concerning the metadata, frames, packets, and timestamps.

2. *Content:* Analyze the content, concerning the objects, object groups, scenes, frames, genre and timeline.

**Participation:** Provide volunteer participation and group management.
1. **Participation if desired:** Viewer can decide between passive watching and active participation.

2. **Anonymous participation:** Participation is anonymous.

3. **Participation in private group:** Participants can build private groups. Members need an invitation from the creator to participate in this group. Participation in this group can be anonymous or public (group decision).

4. **Participation in open group:** Any participant can join an open group. These open groups might have some restrictions like number of members, a venue for the participants, or other criteria. Open groups are always public, but participants can be either anonymous or visible.

5. **Functionality to administrate groups:** Mechanisms to administrate groups e.g., invite, decline, and exclude participants, assign permissions, get information of participants, or similar. If a creator and/or administrator of a group is not explicitly chosen, mechanisms to determine a leader are needed.

6. **Connect participants:** Provide automated awareness of other participants and connect group members.

7. **Handling of channel hopping:** Active participants that switch channels must be excluded from collaborative actions and groups.

**Collaborative Services:** Provide a collection of collaborative services.

1. **Conflict resolution mechanisms:** This affects conflict resolution for collaboration in groups.

2. **Pool of collaborative services:** A pool of existing collaborative services provided to the consumers.

3. **Guidance for each collaborative service:** Each collaborative service will start with a wizard to introduce the participant and ask for all necessary information and data.

**Observe Collaborative Activity:** Identify, observe, and process collaborative activity.

1. **Identify collaborative activity:** Automated analysis of collaborative activity concerning its content, topic, answers, etc.

2. **Observe collaborative activity:** Check if collaborative activity is going on.

3. **Process collaborative activity:** Further process of collaborative activity analysis results.

**3.5.1.2 Advisable functional requirements**

**Provide several sources and video formats:** IPTV, DVB, Web TV, and streaming technology, as well as several MPEG versions, avi, or similar.

**Session management:** Establish and save sessions in case of termination.

**Group allocation methods:** Provide functionality and support for group allocation.
**Error handling:** In case of temporary inactivity of a participant, a crash or unexpected termination of the application, a loss of connection, or similar.

### 3.5.2 *Broadcaster Server Module*

The *Broadcaster Server Module* is the immediate counterpart to the *Consumer Client Module*. It broadcasts the video content, receives, and analyzes incoming results and quantification of collaborative activity, and modifies the content if necessary. In the case of non-modification scenarios, this module solely acts as a broadcaster. Figure 21 gives an overview of the mandatory and advisable functional requirements as well as their interaction(s).

### 3.5.2.1 *Mandatory functional requirements*

**Analyze incoming quantification:** With regard to previously defined intervals, extreme values, string matches, or similar.

**Map broadcast technology as medium to the program content:** Provide a mapping mechanism to get access to the program content via the broadcast medium.

**Update broadcast technology as medium and program content:**

1. *Broadcast Medium:* Provide methods and interfaces to access single elements of the medium, like metadata, frames, or packets, and their properties.

2. *Broadcast Content:* Provide a mapping mechanism to get access to the content via the medium.

### 3.5.2.2 *Advisable Functional Requirements*

**Security issues:** Compliance of evaluated security issues, like data protection, proof of identity, malware protection, and handling of personal data of group members.

**Age limit and verification:** Verification of a participant's age in the case of existing movie ratings and competitions.
3.5.3 **Broadcaster/Producer Module** to Prepare and Enhance Medium and Content Before Broadcast

![Diagram of Broadcaster/Producer Module](image)

**Figure 22:** Surveying Model of the **Broadcaster/Producer Module** to Prepare and Enhance the Broadcast Medium and Content

Before the video content is delivered and broadcasted, it must be enhanced with collaborative services and prepared for modification. That are the main tasks of this module. Figure 22 gives an overview of mandatory and advisable functional requirements as well as their interaction(s).

3.5.3.1 **Mandatory functional requirements**

**Collaborative Services:** Provide a pool of collaborative services as well as methods and interfaces to enhance and manage them.

1. **Pool of collaborative services:** Provide a pool of collaborative services, managed in a database.

2. **Administration of collaborative services:** This database includes available collaborative services, their descriptions, possible triggers, etc., as well as the functionality to combine services with new ones, rate, include, or exclude services from this pool.

3. **Guidance for each collaborative service:** Each collaborative service will start guidance a wizard to introduce the participant and ask for all necessary information and data. Provide methods and interfaces to customize this guide.

4. **Enhance collaborative services to the medium/content:** Enhance events to the medium/content which trigger the activation and deactivation of collaborative services.
   a) **Initialize Event:** Provide methods and interfaces to initialize an event (e.g., defining its ID).
   b) **Anchor Event:** Anchor the event to the ID of a collaborative service.
   c) **Positioning in the Broadcast Medium:** Position the event at an element or condition in the medium (e.g., at a certain timestamp, within metadata, to a certain scene, etc.).
3.5 Technical Requirements

Positioning in the Broadcast Content: Position the event at an element or condition in the content by using the mapping mechanism.

Map medium to content: Provide a mapping mechanism to access the content via the medium.

Trigger in the broadcast content format: Initialize, anchor and specialize triggers.

1. Initialize: Provide methods and interfaces to initialize triggers (e.g., defining its ID).

2. Anchor Triggers to Tags: Provide methods and interfaces to anchor the triggers to single tags of the medium or content (e.g., time stamp, object, scene, frame).

3. Define Properties: Provide methods and interfaces to define the properties of the anchored triggers that are updated with regard to the collaborative activity.

Triggers in collaborative activity: Initialize, anchor and specialize triggers.

1. Initialize: Provide methods and interfaces to initialize triggers (e.g., defining its ID).

2. Anchor: Provide methods and interfaces to anchor the triggers to single tags of a collaborative service provided under certain conditions.

3. Define Properties: Provide methods and interfaces to define collaborative activity, output, etc. of the anchored collaborative service, which is then analyzed.

CMC Link: Link of triggers in the broadcast medium/program content to collaborative activity.

1. Initialize the CMC Link: Provide methods and interfaces to initialize the link; more precisely, to define the triggers that are linked.

2. Define conditions and context: Define conditions and context that must be fulfilled by the collaborative activity in order to update the broadcast content format.
Part II

INTEGRATION OF COLLABORATION INTO BROADCAST CONTENT FORMATS
4 INTEGRATION OF COLLABORATIVE ACTIVITY INTO BROADCAST CONTENT FORMATS – A REFERENCE ARCHITECTURE

A goal without a plan is just a wish.
— Antoine de Saint-Exupery

This chapter presents the conceptual design of the reference architecture (RA) for collaboration in broadcast content formats, whose requirements were defined in Chapter 3. It starts with an outline about reference architectures in general, and then describes the specific reference architecture for collaboration in broadcast content formats. Further on, its elements Strategic Purpose, System Architecture, Standards, Design Patterns, Business Architecture, and Consumer Context are presented in detail.

4.1 REFERENCE ARCHITECTURES

The reference architecture of a domain is an architecture template for all software systems in the domain. It defines fundamental components of the domain as well as the relationships between those components. Hence, an architecture of a particular product is an instance of the reference architecture [38]. The major parts of a reference architecture are templates, patterns, and solution architectures, allowing for design and code reuse. Templates and patterns are high-level models of an architecture to provide some degree of reuse [34]. As defined by Gerrit and Cloutier in [24, 64], reference architectures should address the technical architecture, business architecture, and the customer context as well (cf. Figure 23). The technical architecture provides solutions in technology, known as design patterns. The business architecture describes the “What” (Activity), “Who” (Responsibility), “Which” (Information), and “Where” (Locality) of the reference architecture. Its aim is to specify problems, and context. The customer context refers to consumer beliefs, values, practices and demographics (e.g., age).

4.2 A REFERENCE ARCHITECTURE FOR COLLABORATIVE BROADCAST CONTENT FORMATS

Given the three essential building blocks of our architecture, the Broadcaster/Producer Module, the Broadcaster Server Module, and the Consumer Client Module including the Collaborative Broadcast Content Format (CBF) Player, as well as their conceptual buildup,
the next step is to design the reference architecture (also termed CBF reference architecture). This gives a blueprint for the software, implementing the application scenarios as outlined in Chapter 3.

The design for this reference architecture uses the general requirements for usability as well as functionality, performance, and re-usability. Usability refers to the consumer who must be able to use the Consumer Client Module ad hoc on their television sets and to the Broadcaster/Producer Module to allow the link and integration of collaborative activity with the broadcast content format. Functionality and performance refer to the real-time and time-critical application of the Broadcaster Server Module and the Consumer Client Module. Re-usability refers to the development of a reference architecture and, especially, to the problem of TV related architectures suffering from proprietary software on TV devices and set-top-boxes. The presented reference architecture consists of six components.

Figure 24 shows the assignment of the components to Müller’s suggested customer context, technical, and business architecture. The selection of the components is based on example
architectures described in the literature, like those described by Gerrit Müller [64]. The references architecture in this thesis focuses on the technical architecture, as this thesis is situated in the field of computer science and the developed software is only prototypical, not a finished product. The reference architecture for Collaboration in Broadcast Content Formats comprises the following six elements:

**Strategic Purpose** - Identifies the goals and objectives of the reference architecture. It also recaps the specific purpose of and the problem(s) to be addressed by the reference architecture.

**System Architecture** - Provides a high-level overview of the system architecture.

**Standards** - Technical guidance and standards, based on specified principles that will be followed and implemented as part of the solution.

**Design Patterns** - Generalized architecture representations, that show relationships between elements and artifacts [64].

**Business Architecture** - Usually, the business architecture describes the activity, responsibility, information, and locality of the designed reference architecture. However, as mentioned, this is a computer science thesis, so the business architecture is neglected. In this context it describes solely the lifecycle of collaborative broadcast content formats.

**Consumer Context** - Refers to beliefs, values, practices, and demographics of the mean consumer. The consumer context is not specifically listed but described with strategic purpose, as in Chapter 3.

### 4.2.1 Strategic Purpose

The strategic purpose provides the basis for the principles, technical positions, and patterns in the reference architecture. To recap related work from Chapter 2, four acquired problem classes can be summarized and refined.

The first problem class is the one-dimensional narration space. This problem is not sufficiently solved in this thesis, because it would need a creative redesign of program content & formats, not a software architecture, and this thesis is in the field of computer science, not art or content creation.

The second problem class is the media disruption, caused by the separation of the television user platform, its broadcast medium and program content, and social and collaborative activity, sometimes thematically, but mostly technically. The technical separation of the TV user platform and broadcast medium can be attributed to proprietary hardware and software requirements of TV sets and set-top boxes and to lacking collaboration support of established standards on the TV market (e.g., MPEG-2, software on set-top-boxes, etc.). The hardware and software of TV sets and set-top boxes require adaptable and reusable software architectures, which will be made by implementing a reference architecture and its main characteristics.

The third problem is the missing synchronization of the (collaborative) user activity and the broadcast medium and the program content.
The fourth problem is the scope of collaboration, which is mainly limited to voting scenarios, distribution of program content, and unlinked social activity. Most existing solutions specialize in one collaboration scenario, like distribution or social activity, or are customized for one specific program content or format. This issue is addressed by adding modular extensibility to the reference architecture to allow for different collaborative activity in different program content & formats.

The fifth problem deals with interactions between user and application, instead of between user and user. Both cases relate to the medium and content, which refers to problem class two. To deal with this problem, one objective of the reference architecture is to connect collaborators via a peer-to-peer network architecture.

In summary, the goals and objectives of the reference architecture can be deduced from the five problems mentioned above. Its main focus is to integrate and enable real-time and synchronized peer-to-peer collaboration for the TV platform, related to and enabled by the broadcast medium and content. To implement this, a direct integration of collaboration into the broadcast content format and the missing CMC Link between Broadcast Medium as Technology, Program Content & Formats, and Collaborative Activity need to be established.

Figure 25: Component Architecture of the CBF Reference Architecture

Figure 25 represents the system context diagram, showing the three main modules (Broadcaster/Producer Module, Consumer Client Module, and Broadcaster Server Module) of the system and their corresponding field of application in the TV content production value chain, as well as the following system actors:

1. **Producer** - As mentioned in Chapter 3, the term Producer refers to the person in charge of a specific broadcast content that should be made collaborative [65]. Therefore, in this reference architecture, the Producer’s purpose is “making” the
4.2 Reference Architecture

broadcast content format collaborative. This is enabled by the Broadcaster/Producer Module, for which the Producer is the main actor.

2. **Broadcaster** - The Broadcaster transmits the television program to the audience for public or general use [65]. In the reference architecture, the broadcaster’s main function is not the broadcast itself (which is done automatically by broadcast networks like DVB), but rather, similar to the producer, to “make” broadcast content formats collaborative and receive, analyze, and process the collaborative outcome. In essence, the Broadcaster is the main actor of the Broadcaster/Producer Module and the Broadcaster Server Module.

3. **Target Audience** - The Target Audience (or Audience) is defined as the persons reached by the television broadcast. The people in the Audience are the main collaborators in the reference architecture, using the Consumer Client Module. They are also labeled as “consumers” in the text below.

4.2.1.1 A Walk Through the Production Pipeline

Below, a walk through the broadcast medium (audio/video (A/V) stream) provides further insight into the scenarios and problems from the audience’s point of view. In current broadcast scenarios, the broadcast technology as medium, which is encoded into MPEG-2 video and audio [30], is multiplexed with an MPEG Transport-Stream (MPEG-TS) and then broadcast to the consumers. On the consumer side the transport stream is demultiplexed, decoded, and its essence, the content, is displayed.

In the case of collaborative activity, which currently might be voting for something or social activity, the audience has to use another platform and the collaboration might not be related to the content. In the first case, the voting is somehow triggered via the content. For example, the TV host of a show calls people to vote for their favorite candidate on the show. The voting happens by calling a telephone number, sending an SMS, or via a certain Website. In the second case, the social activity happens mostly via Internet platforms like Facebook or Twitter or in some prototypical cases via a social application on the TV set connected to the TV channel.

In the case of voting, the collaboration is triggered, enabled and disabled via the content. As mentioned before the TV host could ask the audience to vote for their favorite candidate (triggered) via telephone. Receiving and analyzing the votes is possible via a certain phone number during a certain time period (enabled) and shut down when the time period expires (disabled). In the case of social activity, fan pages and groups on Facebook or similar are available all the time, with no regard to the content or whether it is broadcast or not. In some new TV program formats, the outcome of the social activity is included in the show (like raising questions), but the social activity still happens on different platforms that causes a media disruption.

To avoid the separation of medium and collaboration it is first necessary to somehow link them. Since the collaborative activity happens on the audience’s side (which is the TV platform, and is called consumer client from now on), one obvious solution is to attach metadata to the A/V stream, which could enable or disable collaborative activity on the consumer client. Metadata can be key-length-value (KLV) pairs or XML structures and
included as a separate data stream into the MPEG-TS, which is then broadcast to the consumer. On the consumer's side, the MPEG-TS stream is decoded and the data stream, including the metadata, is deserialized and parsed by the player. Depending on the metadata, collaborative activity, like a chat on the TV platform or on a second screen, can be enabled, disabled, or analyzed and returned to the broadcaster. To provide different TV program formats with different collaborative scenarios, the metadata can be attached to different collaborative apps.

4.2.2 System Architecture

This section describes the system architecture of the reference architecture. Implementation details are discussed in Chapter 5.

![System Architecture and Workflow of the CBF Reference Architecture](image)

**Figure 26:** System Architecture and Workflow of the CBF Reference Architecture

The main purpose of the reference architecture is to link collaboration and the broadcast content format, where this CMC Link is realized by embedding metadata into the broadcast medium using the **Broadcaster/Producer Module**. More precisely, a data stream of metadata frames will be included into the medium transport stream.
As shown in Figure 26, the architecture consists of three main parts: Broadcaster/Producer Module, Consumer Client Module, and Broadcaster Server Module. These modules can be reduced to the initially defined components Broadcast Technology as Medium, Program Content & Formats, Collaborative Activity, and the CMC Link, as described in Chapter 1 and recapped in Figure 27. The module's components are described in the pattern section below.

4.2.2.1 Broadcaster / Producer Module

The Broadcaster/Producer Module allows for the creation and inclusion of metadata into the broadcast technology as medium (cf. Figure 26). The metadata acts as a trigger, telling the player on the Consumer Client Module at which tag which collaborative application should be enabled or disabled with certain configurations. It also acts as a mine of information, updating the data that are processed by the enabled collaborative application(s). A tag is a term to represent a certain timestamp, scene, period of time, or any other defined reference. The creation and validation of previously defined metadata are prerequisites. The metadata are serialized into a utf-8 coded data package, which is in turn included into a previously created data stream. The data stream is synchronized with the video stream and multiplexed into the MPEG-TS.

The interaction of Producers and Broadcasters with the Broadcaster/Producer Module is illustrated in Figures 28 and 29, as a UML use case and activity diagram. The main purpose of this module is to create and include the metadata at specific units. For this, a user interface should be provided which opens, presents, and summarizes the MPEG-TS file.

Figure 27: Initial Components as Identified in Chapter 1

Figure 28: Use Case Diagram of the Broadcaster/Producer Module
in a way that it is processable by the user. This includes summarizing the MPEG-TS video stream in units more easily processed by a human being; scenes, time periods, rounds in a game, or similar. The included XML metadata should be validated before the MPEG-TS video, audio, and data streams are finally encoded.

4.2.2.2 Consumer Client Module

The Consumer Client Module consists of the CBFPlayer and the on-demand collaborative application. The player is a conventional video player whose main tasks are demultiplexing, decoding, synchronizing, and displaying the A/V streams. In addition, it decodes, synchronizes, and processes the data stream if it exists. In this case, the metadata are deserialized from the data package and passed to the parser. Depending on the metadata, the collaborative application is either enabled with submitted parameters, or disabled. In the case of already enabled collaborative applications, metadata are used to update the status, gui, data, etc. of the enabled collaborative application(s). The collaborative application may be a fixed part of the player, which is launched with extracted parameters, or a player-defined interface that is downloaded from a repository and launched either via the player or a second screen application, depending on the metadata and extracted parameters.

To configure user data, privacy, and security options, the CBFPlayer provides a setup for basic settings (including user profile, privacy, and security options) and wizards for setting up group activities (like automatic group-building assistants or assistants to set up user-defined groups).

Since broadcast applications are highly time critical, usability is the most important design requirement for this module. The audience must be able to understand, set up and use the client within the first 5 - 10 minutes of the broadcast content. To process the outcome of the collaborative activity (optional!), it must be measured, analyzed, and quantified. Measure the collaborative activity, analyze the outcome of the ongoing collaborative activity, and quantify the outcome of the measurement and analysis as numeric values. The results of the analysis, or the quantified value, can be processed by the Broadcaster Server Module. The interaction of the Audience with the Consumer Client Module is illustrated in Figures 30 and 31 as a UML use case and activity diagram. In summary, the main purpose of the Consumer Client Module is to provide a player that not only displays the video and...
audio streams, but decodes, synchronizes, and parses the data stream. The player should also notify the viewer about available collaborative applications, and whether they are available at the TV station or on a second screen. This module also provides collaborative applications which can be accessed via the player. In theory, any collaborative applications that fulfill the interface requirements are possible.

4.2.2.3 Broadcaster Server Module

The Broadcaster Server Module is an optional module that should collect, analyze, and process the collaborative outcome. The collaborative outcome is the result of the measurement, analysis, and quantification of the collaborative activity, as described in the previous module. It should also manipulate the current broadcast medium in real-time, depending on the collaborative outcome. Real-time manipulation could be, for example, the change of existing data packages, or the inclusion/deletion of already embedded data packages that have not been broadcasted. This is an open section and can be realized.

Figure 30: Use Case Diagram of the Consumer Client Module
individually, depending on the application scenario. The interaction of the Audience with

the Broadcaster Server Module is illustrated in Figure 32 and 33 as a UML use case and activity diagram. The main purpose of this module is to process the collaborative outcome received from the consumer client modules.
4.2.3 Standards

This section outlines the technical guidelines and standards established for the modules of the reference architecture. Due to rapidly evolving technology, this specifications must change often to keep themselves current. Some of these specifications include required services, standards, agreements, security models, communication protocols, web services, XML name spaces, data quality, etc. The provided information is divided based on the three modules (cf. Figure 26) and listed in Table 10.

<table>
<thead>
<tr>
<th>Part / Module</th>
<th>Producer / Broadcaster</th>
<th>Consumer Client</th>
<th>Broadcaster Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>gcc 4.4.5 (C, C++), SDL 1.2</td>
<td>gcc 4.4.5 (C, C++), SDL 1.2</td>
<td></td>
</tr>
<tr>
<td>UI</td>
<td>GtK+2.16,</td>
<td>GtK+2.16,</td>
<td>libav 0.7 -</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Glade 3.6.7</td>
<td>“Forbidden Fruit”</td>
</tr>
<tr>
<td>Medium Support</td>
<td>libav 0.7 -</td>
<td>libav 0.7 -</td>
<td>libav 0.7 -</td>
</tr>
<tr>
<td></td>
<td>“Forbidden Fruit”</td>
<td>“Forbidden Fruit”</td>
<td>“Forbidden Fruit”</td>
</tr>
<tr>
<td></td>
<td>FFMPEG v. 0.8.5</td>
<td>FFMPEG v. 0.8.5</td>
<td>FFMPEG v. 0.8.5</td>
</tr>
<tr>
<td>Collaboration</td>
<td>Peer-to-Peer pattern</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support</td>
<td>Boost Interprocess Library</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td>XML 1.0 (5th version)</td>
<td>XML 1.0 (5th version)</td>
<td>Pugi XML Parser</td>
</tr>
<tr>
<td></td>
<td>Pugi XML Parser</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>DVB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broadcast</td>
<td></td>
<td>MPEG-Audio, MP3 / MPEG-2, H.264 / UTF-8</td>
<td></td>
</tr>
<tr>
<td>A/V/D Coding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium Container</td>
<td></td>
<td>MPEG-TS</td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Technical Guidance and Standards Established for the Reference Architecture

4.2.4 Design Patterns

Schümmer et al. say the main purpose of patterns is to capture formal solutions to specific problems while maintaining a level of abstraction above design models and source code. With this concept of abstraction, they provide a high-level form of reuse, which is independent from language, paradigm, and hardware [20, 74].

4.2.4.1 Structure of Patterns

The description and structure of the reference architecture’s patterns are based on the Alexandrian Form, designed by Christoper Alexander in his book “A Pattern Language (APL)” [10]. More precisely, the patterns are natural-language descriptions of seven elements, as shown in Table 11.
4.2 Reference Architecture

<table>
<thead>
<tr>
<th>Name</th>
<th>The name serves as the pattern's identification. Similar to the Alexandrian Form, asterisks were used to mark existing, well-established patterns. Patterns marked with one asterisk represents preliminary solutions that were adapted for the broadcast application. Patterns marked with two asterisks are “nice-to-have”, but not utilized in this thesis because they would be out of scope. Unmarked patterns are developed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>The context describes the state of the system for which the pattern is intended. It does not describe the real world scenario - this is described in the element “Scenario” below. The main purpose of this property is to show how this pattern could be applied. The description of the context is just one example; the pattern may be also suitable for other contexts.</td>
</tr>
<tr>
<td>Problem</td>
<td>This property describes the problems that occur often again and should be solved by the pattern.</td>
</tr>
<tr>
<td>Solution</td>
<td>How the pattern solves the described problems, and how to use the pattern.</td>
</tr>
<tr>
<td>Scenario</td>
<td>The scenario positions the pattern in a real-world scenario as presented in Chapter 3.</td>
</tr>
<tr>
<td>Symptoms</td>
<td>The symptoms list example context design problems that can be solved using the patterns, as well as recommendations for when to use specific patterns.</td>
</tr>
<tr>
<td>Example Code</td>
<td>For patterns related to the broadcast technology as medium and program content, example code is listed in C.</td>
</tr>
</tbody>
</table>

Table 11: Pattern Description and Structure, Based on the Alexandrian Form

4.2.4.2 Overview of the defined Design Patterns

Frank Buschmann [20] defines patterns as design vocabulary and pattern languages as the grammar and style of the defined vocabulary. The defined design patterns demonstrates how to create a particular kind of system - in the case of this thesis, the design patterns would demonstrate how to build up the three modules of the system architecture (Broadcaster/Producer Module, Consumer Client Module, and Broadcaster Server Module).

Note: The pattern collection is one component of the reference architecture, but it does not define the complete system and is therefore by no means complete. Patterns which are already described in literature are listed, and if necessary adapted, for the broadcast sector, but not fully described further. A reference refers to the full pattern description in the literature. The pattern design assumes the usage of the MPEG-2 standards as previously argued. The Application Management, Community Support, Group Support, and Collaboration Support layers are based on [74].

The developed design patterns consists of six main layers, as Figure 34 indicates. Each layer is further organized into topic clusters, with each topic cluster addressing a specific need in the process (similar to the pattern language in [74]). The layers reflect the model of the overall system and its partitioning into broadcast technology as medium (red colored clusters), program content & formats (violet colored clusters), collaborative activity (blue colored clusters), and CMC Link (yellow colored clusters). The patterns are not built into
the modules of the system architecture because, as mentioned, the design patterns give the blueprints for how to build the modules, but leave only enough space for design.

To integrate collaboration into the broadcast content format, and consequently link collaborative activity to the broadcast technology as medium and program content & formats, the trigger(s) must be included in the medium. The MPEG-2 standard does not allow direct access to the content, like, for example, the MPEG-4 does, so the tagging of the content must be done via the broadcast medium, by embedding triggers regarding to the content into the transport layer of the broadcast medium. This is done by the **Medium and Content Support** patterns. Therefore, these patterns are considered if medium and content should be linked to collaborative activities. They describe how to get access to the broadcast medium in order to embed, extract, and process the triggers. The main target groups for this layer are producer, broadcaster, and audience. Producer and broadcaster use the **Broadcaster/Producer Module** to create triggers in the form of metadata and embed them into the existing medium. The broadcaster uses the **Broadcaster Server Module** to include or exclude the triggers during the broadcast if possible. The audience is involved implicitly, by patterns analyzing the medium for embedded triggers and parsing them, through the Consumer Client Module.

1. **Medium Access** - Patterns in this category describe how to access elements of the medium in order to analyze, change, or enhance them. Accessible elements of an MPEG-2 medium include streams and frames.
2. **Medium Analysis** - Patterns in this category describe how the medium or elements of the medium may be analyzed for triggers.

3. **Medium Modifications** - This category is about modifying existing elements of the medium regarding collaborative activity.

4. **Medium Enhancements** - This category is about enhancing the broadcast content format for triggers.

5. **Tags to Anchor the Triggers to the Broadcast Medium and Content** - Patterns in this category define tags to anchor triggers to the broadcast content format.

The **Application Management** layer defines patterns for the basic infrastructure of the system, excluding the network structure. The network structure of the system can be assumed, so it is not necessary to define patterns for it. These patterns handle the session management and manage shared objects and data consistency. These patterns are mainly directed to developers who have to figure out how to manage shared objects and how information exchange is mediated by the computer system.

1. **Session Management** - Patterns to build and manage shared sessions between the collaborators.

2. **Shared Object Management** - This category includes patterns to manage synchronization of shared objects between collaborators.

3. **Data Consistency Management** - Patterns to manage data consistency and avoiding race conditions and inconsistencies between collaborators.

The **Community Support** layer contains patterns for establishing and managing a community, which is necessary for collaboration in a large organizational context. The community is seen as an optional superior structure, which, if necessary, can be split into groups. These patterns mainly describe (social) processes like entering or leaving a community, support, security, and administrative issues. The main target groups are the audience and the broadcaster. The audience uses the **Consumer Client Module** as a community member. The broadcaster uses the **Broadcaster Server Module** as a community founder and, therefore, also administrator. In many cases, these patterns can be implemented by tailoring existing community support technology (such patterns are marked with an asterisk).

1. **Community Management Activities** - Patterns to manage communities.

2. **Privacy** - Patterns to manage security issues within communities.

The **Group Support** provides patterns for supporting groups in their establishment, enhancement, closing, communication, and interaction. These patterns should be considered if groups are provided but not built yet. They mainly describe the (social) processes of building, closing, joining, or leaving a group, and support, security, communication, and administrative issues. The main target group is the audience, using the **Consumer Client Module** in the roles of group member or founder/administrator. Similarly to the community patterns, in many cases these patterns can be implemented by tailoring existing group support technology (such patterns are marked with an asterisk). The critical factor is the
delay in broadcasting scenarios, which means joining/building/leading/interacting in the group is time-critical.

1. **Group Management Activities** - Patterns to manage groups.

2. **Group and Inter-group Awareness** - Patterns to provide awareness between members of a group and between groups.

3. **Communication Activities** - Patterns to provide communication mechanisms between members of a group and between groups.

4. **Collaborative Artefact Management** - Patterns to manage shared artifacts within a group.

**Collaboration Support** contains patterns for providing collaboration and for making the outcome of the collaborative activity available for further processing. It is assumed that users (the audience) have found a collaboration context (the TV content) and that they want to perform tasks that advance the group towards its collective goal. Therefore these patterns should be considered if collaborative activity is linked to the medium/content and if the outcome of the collaboration should be collected, analyzed, and/or further processed somehow. The patterns describe how the collaborative activity is provided, how collaborative activity is measured, and how the outcome of the collaborative activity is analyzed and quantified. In addition they describe how the metadata are linked to the collaboration (mostly by application identifier and parameters). This layer solely defines the basic support for collaboration, not specific collaborative applications that allow the continuous composition and add-on of collaborative applications. The target groups are consumers who will use the collaborative application provided on their TV stations or second screens, and producers, or any technician who has to design the collaborative application correlating with the broadcast content format.

1. **Collaboration Provision** - Patterns which provide collaborative activities between members of a group (e.g., voting).

2. **Collaboration Analytics** - This category includes patterns which describe how collaborative activity can be analyzed and detect if collaborative activity is happening.

3. **Collaboration Interaction Observation** - This category includes patterns which describe how detected and analyzed collaborative activity can be observed and processed.

The **CMC Link** layer contains patterns of how broadcast technology as medium/program content & formats and collaborative activity are linked. More precisely, they define examples of triggers and linking parameters of collaborative applications as XML metadata. Other elements can conceivably trigger collaborative activity, so this layer can be extended accordingly. The target group is the producers, who have to consider, design, and plan the CMC Link during pre-production and production phase and include the triggers (metadata in this case) in the broadcast medium during the post-production phase.

1. **Enable** - XML Patterns to enable collaborative services to the viewer.

2. **Disable** - XML Patterns to disable collaborative services to the viewer.
3. **Update** - XML Patterns to update the data processed by the enabled collaborative services.

4. **Coordinate** - XML Patterns to coordinate collaborative activity between the viewers.

*Note:* A detailed description of the patterns is given in the Appendix C.

### 4.2.5 Business Value Chain - Lifecycle of Collaborative Broadcast Formats

As this thesis is not in the field of economics but in computer sciences, the description of the business architecture is confined to the outline description of the broadcast content lifecycle. The presented broadcast content lifecycle is based on the broadcast content value chain (cf. Figure 35), as discussed in Chapter 1, but has been adapted to the direct integration of collaboration into the broadcast content formats.

*Note:* The value chain presented in Figure 35 and described below is universally it only needs to be adapted and refined to fit different scopes.

![Broadcast Content Value Chain for Collaborative Broadcast Formats](image)

**Figure 35:** Broadcast Content Value Chain for Collaborative Broadcast Formats [52]

#### 4.2.5.1 Content Creation

The Content Creation phase includes the three consecutive stages of Pre-production, Production and Post-production [65]. Everything from the inception of the project idea to setting up for actual recording is part of the pre-production stage. Pre-production must include the planning of the narration structure, which should be non-linear for a collaborative broadcast content format. In addition, it should include the planning, design, and integration of the collaborative story overlay. Among other parts, collaboration has to be considered with the proposal, the treatments, and the scripts. The proposal is a market summary to promote and sell a project. Because of the extra expenses, and for promotion issues, the optional collaborative interactivity of the project must be emphasized in the proposal. Treatments are subject-matter summaries in short-story form that are told in the present tense and describe events as they happen. They are typically the first draft of a screenplay for a television program. In such treatments, the “what” and the “when” of collaborative activity is defined. The “what” refers to the type of collaborative activity, and the “when” refers to at which points and branches in the narration is collaborative activity is integrated. The scripts are almost complete production guides, specifying what will be seen and heard in the finished product. The scripts must also have alternate endings for scenes dependent on collaborative activity In addition, puffers for delays and synchronization have to be considered [65].
The story is produced, possibly with alternating directions and endings depending on the narration structure. For the collaboration, the collaborative applications are synchronized to the visual images and sound [65]. In the scope of a TV format, the pilot is produced. Post-production starts after the visual images and sounds have been recorded. It ties together the audio, visual, and, in this case, the collaborative elements of the production. Tags are defined by generating metadata, and the broadcast medium, program content, and collaborative activity are linked by including the metadata into the data stream of the broadcast medium. A handout describes the structure of the content, the tags, the branches, the collaborative activity, and the preparation for the viewer [65].

4.2.5.2 Packaging and Aggregation

The produced collaborative broadcast content format is promoted and offered in combination with collaborative applications and a handout.

4.2.5.3 Distribution

The MPEG-2 transport stream, including the data stream, is broadcast. Before this, the collaborative applications and a user handout must be provided online, and during TV commercials and previews.

4.2.5.4 Consumption

The consumers/viewers consume the broadcast content format passively or actively (consumer interaction and/or collaboration).

4.2.5.5 Archiving

The broadcast content format is archived for further broadcasts. Since the collaboration is optional, the content can be aired as passive content. In a later post-production phase, the broadcaster itself can summarize and integrate the content as additional information.

A TV format is only allowed to be called a "TV format" if it successfully passes all the steps in the development process.
IMPLEMENTATION OF COLLABORATION IN BROADCAST CONTENT FORMATS

“Alone we can do so little; Together we can do so much.”
— Helen Keller

Given the three essential building blocks of our reference architecture, namely the Broadcaster/Producer Module, Consumer Client Module and Broadcaster Server Module (cf. Figure 36), the next step is to find and build a suitable and reusable software architecture that enables our application scenarios, as outlined in the previous chapters.

Figure 36: Technological Outcome of this Thesis

This chapter starts with an implementation overview, followed by a detailed description of the three modules and, then, possible error protection and a short description of the scenarios which were implemented in the context of the evaluation of the reference architecture. Finally, a procedure model for application developers will provide a manual on how to develop collaborative broadcast scenarios by means of the presented reference architecture.

5.1 OVERVIEW & ARCHITECTURES

The prototypical implementation was done according to the previously presented reference architecture, resulting in the realization of the three main modules Broadcaster/Producer Module, Consumer Client Module and Broadcaster Server Module.
5.1 Overview & Architectures

5.1.1 Architecture

In Figure 37, the package diagram of the prototypical implementation of the proposed approach in the thesis is illustrated. Basically, the package diagram is composed of the three main packages of **Broadcaster/Producer Module**, **Consumer Client Module** and **Broadcaster Server Module**.

The purpose of the **Broadcaster/Producer Module** is mainly to provide a platform for integrating XML metadata to the MPEG-TS container. It possesses functionality to, firstly, multiplex/demultiplex, encode/decode the MPEG-TS container and, finally, present the video and data stream as well as their outlines. Secondly, the package provides services to create, add and delete a data stream into the MPEG-TS container. Moreover, it supports the ability to create, edit and parse XML metadata and, finally, to create, add and delete these XML data frames into the data stream.

The **Consumer Client Module** package holds functionality to build the client on the viewer’s side. A differentiation between TV integrated and second screen software is necessary. TV integrated software runs on the viewer’s set-top box; the collaborators interact with
their TV sets. Second screen software runs on an external device, for example the viewer’s tablet, but is synchronized by the TV set receiving and processing the collaborative broadcast content format. The collaborators interact with their external devices. Therefore, the package constitutes, the first instance, functionality to demultiplex and decode the incoming video, audio and data stream of the MPEG-TS. In the case of a data stream, metadata needs to be deserialized, parsed and finally processed by the player to offer collaborative activity. Secondly, this package holds interfaces for collaborative applications which are offered to the viewer. The collaborative applications are proprietary and need to be implemented with the development of the broadcast content format. Finally, techniques to connect peers and to analyze and measure the collaborative activity are provided.

The *Broadcaster Server Module* package consists of standard mechanisms to collect and process the received collaborative outcome.

The reason to design three separate modules is in the design of the whole framework. One module for each type of user is needed (cf. Chapter 4), which results in a client-server similar architecture.

### 5.1.2 Implementation Requirements

As demonstrated in Chapter 3, the broadcast and especially the television area holds many standards, although only a fraction in commercial usage. The used standards for the ecosystem television were chosen with consideration to their market relevance in Europe. In a nutshell, the architecture is based on a standard television ecosystem used in Europe (like DVB MPEG-2). The focus was on using, at the time of the prototypical implementation, prevalent standards (like XML and libav). Table 12 gives an overview of the used standards outlined at a glance below.

**C++ Standards** For the implementation, a C/C++ core on a Linux system was built by using the Gnu Compiler Collection (GCC)\(^1\), which includes front ends for C and C++. To implement the graphical user interfaces of the modules, the GIMP Toolkit\(^2\) (Gtk+) was consulted.

**Libav** Libav\(^3\) is an open source project to process audio and video data. It includes the leading audio/video codec library *libavcodec* and *libavformat*. The first library provides codecs for encoding and decoding video and audio data [2]. The second one provides a generic framework for multiplexing and demultiplex (muxing and demuxing) audio, video and subtitle streams. It encompasses multiple muxers and demuxers for multimedia container formats [3]. In addition, *avplay* was used as the base for the developed video player.

---

1 http://gcc.gnu.org/
2 http://www.gtk.org/
3 http://www.libav.org/
<table>
<thead>
<tr>
<th>Part / Module</th>
<th>Producer / Broadcaster Module</th>
<th>Consumer Client Module</th>
<th>Broadcaster Server Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>gcc 4.4.5 (C, C++), SDL 1.2</td>
<td>gcc 4.4.5 (C, C++), SDL 1.2</td>
<td></td>
</tr>
<tr>
<td>UI</td>
<td>GtK+2.16,</td>
<td>GtK+2.16, Glade 3.6.7</td>
<td></td>
</tr>
<tr>
<td>Medium Support</td>
<td>libav 0.7 - “Forbidden Fruit”</td>
<td>libav 0.7 - “Forbidden Fruit”</td>
<td>libav 0.7 - “Forbidden Fruit”</td>
</tr>
<tr>
<td></td>
<td>FFMPEG v. 0.8.5</td>
<td>FFMPEG v. 0.8.5</td>
<td>FFMPEG v. 0.8.5</td>
</tr>
<tr>
<td>Collaboration</td>
<td>Peer-to-Peer pattern</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support</td>
<td>Boost Interprocess Library v. 1.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td>XML 1.0 (5th version) Pugi XML Parser</td>
<td>XML 1.0 (5th version) Pugi XML Parser</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>DVB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broadcast</td>
<td>MPEG-Audio, MP3/MPEG-2, H.264/UTF-8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium Container</td>
<td>MPEG-TS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 12: Technical Guidance and Standards Established for the Reference Architecture

FFMPEG 4 is a cross-platform solution to record, convert and stream audio and video data [4] [1]. FFmpeg was used to stream the MPEG-TS file in the experimental setup.

MPEG-TS MPEG transport stream is a standard container format for the transmission and storage of audio, video and program data. It is used in broadcast systems like DVB [41]. The MPEG transport stream is specified in the MPEG-2 part 1 specification, formally known as ISO/IEC standard 13818-1 or ITU-T Rec. H.222.0. The transport stream uses the concept of programs and allows one or more programs to be combined into a single stream. Each single program is described by a Program Map Table (PMT), which has a unique Program ID (PID). The elementary streams associated with that program have PIDs listed in the PMT. For instance, a transport stream used in a digital television environment might contain three programs, to represent three television channels. Suppose each channel consists of one video stream and one or two audio streams. A receiver wishing to decode a particular "channel" merely has to decode the payloads of each PID associated with its program. The content of all other PIDs is discarded. A transport stream with more than one program is referred to as MPTS - Multi Program Transport Stream. A single program transport stream is referred to as SPTS - Single Program Transport Stream. [41]

4 http://www.ffmpeg.org
The video player is implemented by using the simple direct media layer\(^5\) (SDL). Simple DirectMedia Layer is a cross-platform multimedia library designed to provide low level access to audio, keyboard, mouse, joystick, 3D hardware via OpenGL, and 2D video frame buffer. Amongst others, it is used by MPEG playback software [5].

**Boost Interprocess** The Boost Interprocess\(^6\) Library supports the use of common interprocess communication and synchronization mechanisms.

**Metadata** The CMC Link between broadcast technology as medium, program content & formats and collaborative activity is given by the inclusion of metadata into the broadcast MPEG transport stream. The metadata can be defined as key-length-value pairs (KLV), extensible markup language (XML) or any other (proprietary) structure and language. In this reference architecture, metadata are structured by using the extensible markup language and included as data packages within a data stream into the MPEG-TS.

The XML data elements are structured in three main parts *Basic Information*, *Application Specific Information* and *Coordinating Information*: *Basic Information* relates to initial information about the collaborative application itself and its current state in the collaboration. This part of the XML data is obligatory in order that the data packet can be assigned to a collaborative application. If this information is missing, the packet is discarded.

The following elements are provided (* - element is obligatory in each case, ** element is obligatory at the startup):

- **Unique Id**\(^*\) of the collaborative application.
- **Name**\(^**\) of the collaborative application.
- **Parameter** that is needed for the application to start.
- **Repository**\(^**\) where the application can be downloaded.
- **PacketUpdate**\(^*\) to store (1) or update (0) a packet. Packets are stored for applications which are either not started yet, or packets which are used in future for already running applications. Packets are updated if they where already used by the application but include for instance an update for the GUI.
- **GuiUpdate**\(^**\) if GUI is updated (1) or not (0).
- **Round** current round in the game.
- **Gui Enabler** if the application is enabled (1) or disabled (0).

DTD of the basic informations:

```
1 <!ELEMENT basics (id, name, parameter, repository, packetUpdate, 
guiUpdate, round, guiEnabler)>  
2 <!ELEMENT id (#CDATA)> integer of unique identifier  
3 <!ELEMENT name (#CDATA)>  
4 <!ELEMENT parameter (#CDATA)>  
5 <!ELEMENT repository (#CDATA)>  
6 <!ELEMENT packetUpdate (#CDATA)> 0 - update packet, 1 - store packet  
7 <!ELEMENT guiUpdate (#CDATA)> 0 - gui is not updated, 1 - update gui  
8 <!ELEMENT round (#CDATA)> integer of round number
```

5 http://www.libsdl.org/
6 http://www.boost.org/doc/libs/1_55_0/doc/html/interprocess.html
5.1 overview & architectures

Example:

```
<?xml version="1.0"?>
<basics>
  <id>55</id>
  <name>CBF Gaming Application</name>
  <parameter></parameter>
  <repository>ftp://123.45.67.890</repository>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>1</guiUpdate>
  <round>0</round>
  <guiEnabler>1</guiEnabler>
</basics>
```

Explanation:
The CBF Gaming Application (name) with the unique identifier “55” should be offered to the viewer at the specific time stamp at which the packet is included into the MPEG-TS data stream (cf. Figure 38). The application doesn’t need any parameters for its startup and, if it is not already installed at the client side, it can be downloaded at repository ftp://123.45.67.890 in case if it is started by the viewer. Since this is the initial phase of the TV quiz show, the quiz round is 0 at this stage - which means no round has started yet. If no round is played at the moment, the parameter would be -1. The gui_enabler element, which is set to 1, defines that the player enables this application to start, so the availability of the collaborative application is visible to the viewer by an overlay and/or an additional button at the players front-end. The packetUpdate = 1 means that the packet needs to be stored in the queue. This parameter should be 0, in case previous versions of this packet were already broadcast and therefore the packet is already stored at the client side and needs to be updated only. No further parameters are given to the application.

Application Specific Information relate to the collaborative application only. This includes structure of the GUI and content to be displayed in the GUI. Since these metadata are very specific to the respective collaborative application, the following elements represent a basic sample set and should be changed and expanded for the respective applications:

**Text** any text to be shown in a conventional text area.
Description description of the collaborative application, tasks, rules, etc which can be shown in a conventional text area.

rb Values values for a radio button structure.

cb Values values for a check box structure.

Text Entry any text to be shown in a text entry.

The metadata elements should be adapted and associated to specific elements of the application GUI.

DTD of the application specific informations:

```
<!ELEMENT application (text, textentry, description, rb_values, cb_values)>
<!ELEMENT text (#CDATA)>
<!ELEMENT textentry (#CDATA)>
<!ELEMENT description (#CDATA)>
<!ELEMENT rb_values (#CDATA)>
<!ELEMENT cb_values (#CDATA)>
```

Example:
```
<?xml version="1.0"?>
<task>Eurovision Song Contest 2011 Final Voting</task>
<application>
  <check>
    <value> Ireland - Jedward - "Lipstick" </value>
    <value> Sweden - Eric Saade - "Popular" </value>
    <value> Estonia - Getter Jaani - "Rockefeller Street" </value>
    ... </value>
  </check>
</application>
```

Explanation:
This example shows XML metadata for a voting scenario. The viewer should be able to vote for their favorite singer at the Eurovision Song Contest Semi Final. In this case, the task element describes the task to be undertaken by the participants. It is shown in a simple text area of the application’s graphical user interface, (cf. Figure 39). The check element
builds a set of check boxes, with four values.

*Coordinating Information for Collaboration* applies to the function of metadata as coordinator. It includes elements like the common task or question to be solved, peer that is next in line with which task, timing information and so on. Since these metadata are very specific to the respective collaborative application, the following elements represent a basic sample set and should be changed and expanded for the respective applications:

- **Task** which should be performed by the peers. Can be shown in a text area.
- **Question** which should be answered by the peers. Can be shown in a text area.
- **Peer** number that is randomly called into action.

The metadata elements should be adapted and associated to specific elements of the application GUI. DTD of the collaboration specific information:

```xml
<!ELEMENT collaboration (task, question, peer)>
<!ELEMENT task (#CDATA)>
<!ELEMENT question (#CDATA)>
<!ELEMENT peer (#CDATA)>
```

**Example:**

```xml
<?xml version="1.0"?>
<collaboration>
  <question>Any Question</question>
  <task>It's peer 2's turn to guess</task>
  <peer>2</peer>
</collaboration>
```

**Explanation:**

This example shows XML metadata for a quiz scenario to coordinate the peers in the group. The element *question* describes the question for the participants to answer and is shown in a conventional text field. The element *task* describes what to do for the peers acting in a group. In case of acting in a group, each peer needs to know what to do or whose turn it is. This is given by the *peer* element. In case one, the peer element includes the division of tasks, in the second case, this element includes the number of the peer whose turn it is.

The optionality or obligation of the elements depends on the purpose and the status of the application as well as their position in the content.

In addition to the structure of the metadata, they can be categorized in patterns relating to their purpose. Again, the purpose of the metadata is highly dependent to the purpose of the collaborative application. Therefore, in the following, some basic patterns are presented which are “ready to use“ and open to be extended. The detailed pattern description is listed in Section C.6 of Appendix C.

**Enable/Disable** The *Enable/Disable* pattern describes metadata to enable and disable a collaborative application.

The minimum DTD looks like the following:

```xml
<!ELEMENT basics (id, name, parameter, repository, packetUpdate, guiUpdate, round, guiEnabler)>
```
Basically, the element `guiEnabler` needs to be 1 to make the collaborative application available.

The minimum DTD to disable an application looks like the following:

```
<!ELEMENT basics (id, guiEnabler)>
<!ELEMENT id (#CDATA)> integer of unique identifier
<!ELEMENT guiEnabler (#CDATA)> 0 - gui is not shown/offered, 1 - show gui
```

In this case, only the `guiEnabler` needs to be 0 and the application is disabled at the moment the metadata were included.

**Updater** The `Updater` pattern focuses on updating a certain collaborative application at certain conditions. Additionally to the data to be updated, the following elements are required:

```
<!ELEMENT basics (id, packetUpdate)>
<!ELEMENT id (#CDATA)> integer of unique identifier
<!ELEMENT packetUpdate (#CDATA)> 0 - update packet, 1 - store packet
```

The `id` must be set to the application id.

**Coordinator** The `Coordinator` pattern focuses on the coordination of the collaborative activity.

One example for a `coordinator` pattern could be the coordination of the peers to guess in a quiz application.

In this case, the minimum DTD looks like the following:

```
<!ELEMENT collaboration (id, task, peer)>
<!ELEMENT id (#CDATA)> integer of unique identifier
<!ELEMENT task (#CDATA)> which task to solve for the peer
<!ELEMENT peer (#CDATA)> number of the peer in the group
```

## 5.2 Implementation of the Reference Architecture

The prototypical implementation of the proposed reference architecture, laid out in the preceding chapter, is illustrated in the following in order to demonstrate its feasibility. The prototypical implementation is realized based on emerging video processing open source technologies. The resulting implementation covers, on the one hand, modules and libraries for the reference architecture and, on the other hand, two collaborative
scenarios. In this chapter, the focus is on the implementation of the reference architecture. The presentation and implementation of the prototypical scenarios is focused in the next chapter (cf. Chapter 6) to prove the concept of the reference architecture. In the following sections, the main modules Broadcaster/Producer Module, Consumer Client Module and Broadcaster Server Module are explained and laid out by means of more detailed package diagrams. Finally, Error Protection and a procedure model for application developers are provided.

5.2.1 The Broadcaster/Producer Module

The **Broadcaster/Producer Module** provides a graphical user interface including the "CBF Metadata Editor" to, firstly, create XML metadata which are validated at runtime against the XML schema behind. Secondly, it provides an overview of the video stream, which means the video stream is presented in different levels of granularity. For example, every single video frame is shown, which might be overwhelming for the user. Therefore, the user has the possibility to summarize the video stream into scenes, blocks (like advertisement) or periods (arbitrary periods of times regardless of scenes or similar, e.g. 1 minute). Thirdly, it includes the metadata at user defined points in time or periodically, as previously defined.
In the following a walk through the single components of the prototype is provided, according to the component diagram presented in Figure 40.

**FILE TYPES**  
The module accepts two file types to be opened, namely an MPEG-TS file or a project file. The MPEG transport stream combines one or more packetized elementary streams (PES) with one or more independent time bases into a single stream. Elementary streams share a common timebase from a program.

For the prototypical implementation, MPEG-TS files with the ending '.ts' are used. The .ts file is organized by 188 byte transport packets. Each packet starts with a 4 byte header followed by 184 bytes of data payload. Figure 41 provides a schematic overview of an MPEG-TS file. For a detailed description of the MPEG-TS header, please check the corresponding ISO/IEC 13818-1 standard [41].

![MPEG Transport Stream Diagram](image)

**Figure 41: MPEG Transport Stream according to ISO/IEC 13818-1 page 125 [41]**

The header contains [41]

1. Initial synchronization field with the bit pattern 0x47 (0100 0111).

2. Three flag bits are used to indicate the payload processing.
   a) The first flag indicates a transport error.
   b) The second flag indicates the start of a payload.
   c) The third flag indicates a transport priority bit.

3. 13 bit Packet Identifier (PID), which is used to uniquely identify the stream to which the packet belongs.
4. Two scrambling control bits telling whether the payload is not scrambled, scrambled with even or odd keys.

5. Adaption field control indicates if there is additional information following the transport stream header.

6. Continuity counter is a sequence number of payload packets.

A single program stream is used, more exactly one set of video, audio and data stream, which are intended to be played together. The decision for this format was made because it is generally used for traditional broadcasting like DVB or over the air HDTV in Europe. To access the MPEG-TS, previously described video libraries, developed within the scope of the libav project are used. The input MPEG-TS container is demultiplex and decoded by the SimplePlayer component. The SimplePlayer is, as the name implies, is a simple video player which allows playback of the video and audio streams and synchronization of the data stream, if present.

The project file, on the contrary, is a simple XML file format. The DTD of the project file looks like the following:

```xml
<!ELEMENT project (name, path, input, output, metadata, filename, timestamp)>
<!ELEMENT name (#CDATA)> project name
<!ELEMENT path (#CDATA)> path to the project folder
<!ELEMENT input (#CDATA)> input MPEG-TS file (including the path)
<!ELEMENT output (#CDATA)> output MPEG-TS file (including the path)
<!ELEMENT metadata (filename, timestamp)>
<!ELEMENT filename (#CDATA)> metadata filename which includes the timestamp as well
<!ELEMENT timestamp (#CDATA)> timestamp at which the metadata file is included to the MPEG-TS
```

The XML project file is generated automatically by the module after the project was saved the first time and may look like the following example:

```xml
<?xml version="1.0"?>
<project>
  <name>channel_3_vote</name>
  <path>/project/path/</path>
  <input>/any/path/input.ts</input>
  <output>/project/path/channel_3_vote/channel_3_vote_out.ts</output>
  <metadata>
    <filename>channel_3_vote.ts_10.xml</filename>
    <timestamp>10</timestamp>
  </metadata>
  <metadata>
    <filename>channel_3_vote.ts_16.xml</filename>
    <timestamp>16</timestamp>
  </metadata>
</project>
```

7 libav.org
The example XML project file was generated for a project called “channel_3_vote”. The module takes the given input.ts file as input. After the metadata were added, the MPEG-TS file channel_3_vote_out.ts is generated and its path included into the project file. For each XML metadata file that is included into the MPEG-TS, the block metadata is included into the project file with the given parameter. Figure 42 shows the project file and the corresponding data structure, which is created automatically.

Figure 42: Project Folder Structure

**Data Manager** The component Data Manager manages, on the one hand, to demultiplex and decode streams within the opened MPEG-TS container and, on the other hand, to encode and multiplex streams to the output MPEG-TS container. In simple terms, it opens the input MPEG-TS container (input.ts) and writes the output MPEG-TS (output.ts) container. In addition, it provides the video and audio data to the SimplePlayer and the video presentation layer and the metadata to the metadata manager.

The MPEG-TS is loaded, by using features of the libav libraries and FFmpeg.

```
[mpegts@0xab0c9a0] max_analyze_duration 5000000 reached at 5005000
Input #0, mpegts, from ‘../cbf_channel_2.ts’:
  Duration: 00:02:18.13, start: 1.400000, bitrate: 729 kb/s
Program 1
  Metadata:
    service_name  : Service01
    service_provider: FFmpeg
  Stream #0.0[0x100]: Video: mpeg2video (Main), yuv420p, 440x360 [PAR 1:1 DAR 11:9], 104857 kb/s, 29.97 fps, 29.97 tbr, 90k tbn, 59.94 tbc
```
The previous lines show an MPEG-TS file opened with FFmpeg in the console, including a video, audio and data stream. Detailed code information is given in the UML activity diagram (cf. Figure 43).

Existing metadata in the input MPEG-TS container are passed to the metadata manager. They are presented in the timeline and the XML editor of the graphical user interface. If the user creates new metadata with the provided XML editor at a specific timestamp, the XML data packet (as it is called now) is buffered into the metadata queue. The metadata queue is a vector which buffers XMLObjects. XMLObject is a simple data structure which holds the timestamp, the data of an XML data packet.

The Data Manager becomes active again when the created metadata needs to be merged into the MPEG-TS stream - the file channel_3_vote_out.ts is created in the given example, including the data packages.

The inclusion of data packages is done by using the libav libraries. For the sake of clarity, the algorithm is presented in Figure 43 by means of an UML activity diagram. To provide error correction, the data packages are included redundant. More specific information to error correction mechanisms is presented in Section 5.3 and in Chapter 6.

The merging algorithm in Figure 43 uses features of the libav to, firstly, multiplex the existing video, audio and data (if already present) streams into the output MPEG-TS and, secondly, to include the created XML metadata as data packages. In a nutshell, the algorithm checks frame by frame of the input MPEG-TS in relation to its stream affiliation (whether it belongs to the video, audio or data stream). If it is an existing audio or data stream, the frame is inherited into the output MPEG-TS. If it is a video frame, the frame is, firstly, inherited into the output MPEG-TS as well and, secondly, it checks whether the timestamp of a metadata packet in the metadata queue is less than or equal to 0.03 seconds to the PTS (Presentation Time Stamp) of the current video frame. If so, the metadata packet is written into the output MPEG-TS. This is done by using the libav function 'av_write_frame'. The process needs to be done at the very end of the creation and positioning of the metadata, since the frames are written one by one, as Figure 44 shows. In the example of Figure 44, data frames are included at timestamps '1', '4', '7', and '20', but the video frames have the timestamps '1', '2', '3', '4', etc. The data frames are included one after the other. The gaps between, for example, timestamp '1' and '4' for data stream are not filled with empty frames. This has the advantage of a smaller amount of unnecessary data but has the disadvantage of a more sophisticated synchronization between video and data frames on the viewer's side when the MPEG-TS is decoded and played. For more information, please check the Consumer Client Module. The algorithm of including the data packages synchronized to the video frames is quite simple. The program runs through all video frames and checks the timestamp of the current video frame. If it is greater than the timestamp of the data frame to include, the attributes of the data frame are set, and its PTS is set to the PTS of the current video frame. By using the libav function 'av_write_frame', the data frame is written to the output MPEG-TS container. The important issue is setting the PTS to those of the video frame to which the data frame belongs, although the data frames are included into the data stream as described above.
At the clients side, the data frames are demultiplex and decoded at the very beginning and buffered until their PTS corresponds to the those of the current displayed video frame.

The metadata, which are already included in an opened project, are presented in a timeline and in the editor at the given point in time. This metadata can of course be edited.

Figure 43: UML Activity Diagram of the Merging Algorithm

Create the same Streams than in the input MPEG-TS, copy the codec and set their parameter:

```c
avcodec_copy_context(video_codec, AVFormatContext_in->streams[videoStream]->codec);
```

```c
av_interleaved_write_frame(AVFormatContext_out, &AVPacket)
```

```c
av_write_frame(pFormatCtx_out, &pkt)
```

```c
av_write_trailer(AVFormatContext_out)
```
Before metadata are included, they are validated against the XML Schema in the back. Concerning the MPEG-TS specification, one frame has a maximum capacity of 188 bytes. When the XML metadata packet is greater than 184 bytes, it is split into frames of 188 bytes. Four bytes overhead are needed to label the split frames as a series of sub-frames, which need to be assembled at the client side.
For the detailed algorithm, please check the UML activity diagram.

**Presentation of the Video** Generally, this component contains a simple video player which is used to integrate the video into the graphical user interface of the module. The user should be able to play the video and seek the integrated metadata. Moreover, the component provides functionality for the user to structure the opened video. The previously opened, demultiplex and decoded input MPEG-TS is passed to the video player in the form of the libav structure `AVFormatContext`. The `AVFormatContext` data structure contains, amongst others, a list of the streams, start time, duration, bit rate and other parameters of the input MPEG-TS.
Hence, the video stream is presented in the user interface frame by frame, which is a critical point. The video, which was analyzed above with FFMPEG on a Linux console, has a total length of 138 seconds and a framerate of 29.97 fps (frames per second). This results in a total number of 4135.86 frames. Therefore, the user needs the possibility to structure the video frames by the following criteria:

- **User defined blocks** the user can summarize an arbitrary number of frames to one unit.
- **Scenes** the video is summarized by scenes. Change of scenes are marked by the switch of the location, position of the actors or the beginning/end of commercial blocks.
- **Periods** the user can summarize frames by defining a period of, for example, 1 minute. The frames of the whole video stream are automatically summarized into units by the defined period.

Navigation through scenes, user defined units and periods can be done by using the slider below. The number represents the current unit. Only the first frame of a unit is presented in the picture above. The summary of frames into units is done by using a two dimensional vector. The vector holds in its first dimension the name/number of the unit, and in its second dimension two timestamps of the first and the last frame of the unit. If the video stream was not summarized yet, it is displayed in its initial form, frame by frame. The user can navigate through the frames by using the slider below the video player.

**Metadata Manager and Parser** If the user wants to add new metadata, a little editor is provided on the right hand side. New metadata can be created, or existing ones can be
5.2.1 Implementation of the Reference Architecture

displayed and edited. The GUI presents those metadata which are added to the current frame or unit. The metadata can be added to one or more defined units or frames. For this, the first timestamp of the unit or the timestamp of the frame is used and stored as described above. It is possible to add more than one metadata packet to one unit because, in this case, timestamps of several consecutive frames are used. This is no problem since the gap between two frames is within a centisecond (for example 0.0333667 seconds in case of 29.97 fps), but it is not possible to add more than one metadata file to one frame. In a first instance, the XML data and their corresponding timestamps are stored in the project file only as described above. The XML data packets are buffered into the metadata queue. The metadata queue is a vector which buffers XMLObjects. XMLObject is a simple data structure which holds the timestamp and the data of an XML data packet. The metadata are serialized and included into the MPEG-TS file when the user firmly launches this process. When this is done, the metadata are serialized and processed to the data manager, to be included into the data stream synchronized by the video stream, as described in the algorithm above.
The parser’s purpose is to parse the XML data packets in relation to the underlying DTD and for syntax errors.

Graphical User Interface

The graphical user interface is kept very simple and implemented by using the Gimp Toolkit (Gtk+). Figure 45 shows a screenshot of the implemented broadcaster/producer module.

![Screenshot of the Broadcaster/Producer Module](image-url)

Figure 45: Screenshot of the Broadcaster/Producer Module

5.2.1.1 Open Issues

The summarization of the video stream is not implemented yet. The only possibility is to navigate frame by frame by using a slider. In addition, the DTD for the XML metadata
5.2 Implementation of the Reference Architecture

needs to be done before the first metadata are created and included. Any opportunity for creating and validating the DTD by using the Broadcaster/Producer Module instead of any other XML editor would be preferable.

5.2.2 Implementation of the Broadcaster Module (Server)

The Broadcaster Server Module is a simple server application which runs on the broadcaster’s side. Its purpose is to receive and process the data outcome of the collaborative activity of the viewer. Since this is highly dependent on the purpose of the broadcast content format and the purpose of the broadcaster, this module is not further explored in this thesis.

5.2.3 Implementation of the Consumer Module (Client)

This module provides possibilities to build the software client that runs on the consumer’s side. As mentioned above, the software client may run, firstly, directly on the set-top box or, secondly, on a mobile device as second screen solution. For the prototypical implementation, the author assumes that the client is running on the set-top box.
Figure 46 gives an overview of the single components and the interfaces of the Consumer Client Module which are described in the following.

5.2.3.1 **CBFPlayer**

The CBFPlayer (CBF is an acronym for collaboration in broadcast content formats) is a video player which is able to process MPEG-TS files including a metadata stream. It consists of two main parts, namely the simple player and its graphical user interface. The simple player is a very simple video player based on avplay. It is able to demultiplex and decode video and audio streams that are included into an MPEG-TS file. In addition, it is able to process a data stream that is also included in the MPEG-TS file. When a data stream exists, the simple player has three main tasks to perform. Firstly, it must decode the data frames synchronized to the video frames. Since there are usually fewer data frames in the data stream than video frames in the video stream, all data frames are decoded at the very beginning, after the MPEG-TS file was loaded. To prevent them from all being processed at the very beginning, they are queued by the first-in-first-out principle. The timestamp of the first data package in the queue is compared to the timestamp of the video package that is displayed next. If the PTS of the video package is greater than or equal to the PTS of the data package, the metadata data are extracted, then given to the GUI component firstly and, secondly, to the CA Manager. The GUI component processes the metadata in such a way that, firstly, a notifier about possible collaboration is shown to the viewer and, secondly, a button to start the collaborative application is shown in the player. For this, the GUI needs to know which collaborative application to start. At a glance, the GUI component is responsible for notifying the viewer and making the application available to the viewer. This includes adding a corresponding button to the player, checking the availability of the application. If the application is already installed, it provides the path. If it is not installed, it notifies the viewer and provides a download source. Usually the download source is given with the metadata.

5.2.3.2 **CA Manager**

The CAManager is responsible for providing received metadata to the running collaborative application. But, firstly, the metadata needs to be parsed to extract the application id. The id gives information about the application for which the metadata are particular. Secondly, it needs to extract packetUpdate and round id. The packetUpdate id tells if the package needs to be newly stored or if an already stored package needs to be updated. To enable communication with the proprietary collaborative applications, a shared memory is provided. If the CA Manager receives new metadata from the simple player, they are parsed and if metadata for a registered collaborative application were received, they are pushed into the shared memory. The CAManager adds and removes packages to the shared memory. In addition, it updates packages that are already in the shared memory and it stores packages for collaborative applications that are not yet running. The CBFInfo object structure maps the XML metadata structure into a C++ object structure.
5.2.3.3 **Parser Library**

The framework includes one library, namely the Parser. The Parser Library just parses the metadata. The library uses the C++ pugixml XML parser.

5.2.3.4 **Collaborative Application**

The collaborative application is not part of the implementation of the reference architecture. It needs to be designed and implemented during the production phase by using and implementing the defined interfaces of the reference architecture. The implementation of the reference architecture is shown, for example, in Section 5.5.

5.2.3.5 **Open Issues**

The measurement and analysis of the collaborative activity was put in second place during the implementation phase and is, clearly, an open issue. Interesting questions would be the identification of quantification values that are applicable for the television sector (as done for instance in [39]) as well as the real-time processing of collaboration in a time critical setting, as the broadcast sector in general is. Different statistics (cf. Chapter 3) have shown that a significant number of individuals use mobile devices with Internet access. Therefore, the realization of a second screen scenario is the second open issue. Since most TV stations already have Internet access as well, the second screen can be synchronized by the metadata processed on the TV station - similar to the approach presented in [66]. The third open issue is the adaptation of the existing implementation to the HbbTV standard. Obstacles in 2010/2011, when the implementation was done, were missing standards to make the television interactive, which is a precondition for collaboration. The MPEG-4 reference architecture and the DVB-MHP platform were down. Therefore, the researcher was constrained in defining a reference architecture, which took much time and effort. In 2011, the HbbTV standard was published right after the implementation was finished, and the researcher decided to keep it.

5.3 **ERROR PROTECTION**

Errors in video and audio transmission are less critical because of already integrated error mechanisms. If metadata are used to coordinate collaborative activity, the criticality depends on the type of error and the constellation of the collaboration. The error protection in this section refers to loss and errors of metadata, not of video and audio frames. Error protection is important for the metadata since their task is to coordinate the group, as well as the indication for collaborative output. In the following, the impact of erroneous metadata for one individual and for the group, as well as their level of criticality and types of errors are listed. Finally, recommendations for the error protection for the data stream are made.

---

8 http://pugixml.org/
5.3.1 Impact of Erroneous Metadata

Based on the primary goal of the metadata, erroneous or lost metadata would cause the following complications:

1. Loss of synchronization of single group members.
2. Loss of information for single group members.
3. Loss of coordination of the group.

The occurrence of the mentioned complications would have the following impacts:

1. No or reduced collaborative outcome.
2. Impact on the participatory collaborative TV content (to which the collaborative activity and therefore the collaborative outcome is tied).
3. Frustrated user resulting in a possible denial of the CBF system.

From the user’s point of view, of course, all complications are critical, requiring basic error protection. From the collaboration, or more exactly the collaborative outcome’s point of view, the scope of the error protection depends on the scenario. Some scenarios are more critical than others, which allows an application based error protection on different levels but, as mentioned, basic error protection must be provided in any event.

5.3.2 Levels of Criticality

Before defining the levels of criticality, and to find out which scenarios are critical, a simple categorization of scenarios is done. The categorization is done from the collaborative outcome’s point of view, whereas the categories Group Mode and Impact of the Collaborative Outcome where chosen.

1. **Group Mode:** defines the type of group to which the collaborators belong. In the case of this thesis, open and closed groups are defined. Open groups are anonymous, informal groups, which everybody can attend. The group size is undefined and unlimited. In contrast, closed groups are personalized, formal groups, which may have a previously defined size or maximum limit of members. A closed group can be joined only after invitation, acceptance or assignment.

2. **Impact:** defines if the outcome of collaborative activity takes immediate or delayed effect. An immediate effect would be, for example, a battle between the candidate and collaborative groups in a live TV show. In the case of, for instance, a 10 minute voting phase, the impact is delayed after the voting phase has ended.

Table 13 gives an overview of the categorization and one example per category.

For the CBF scenarios, four levels of criticality were defined:

1. **Level 1 - Very Critical:** scenarios where erroneous metadata cause no or reduced collaborative outcome, impact on the TV content and frustrated users.
### 5.3 Error Protection

#### 5.3.3 Types of Errors

To define arrangements for the corresponding level of criticality, it is firstly necessary to define the different error types. The types of errors are based on those which typically occur in the broadcast as defined for instance in [6]. The categories of errors presented below are related to the broadcast of the metadata at frame level and do not refer to the typical broadcast of the video and audio streams (cf. Figure 47).

**Frame Lost**

**Single Loss:** a single loss occurs if one frame of the data stream is lost.

<table>
<thead>
<tr>
<th>Group Mode →</th>
<th>Open Group</th>
<th>Closed Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate</td>
<td>Example: Behavior or tendency of the open group to something.</td>
<td>Example: Quiz battle in a live quiz show between candidate of the show and n collaborative, closed groups of three.</td>
</tr>
<tr>
<td>Delayed</td>
<td>Example: Viewer can vote for their favorite singer.</td>
<td>Example: The group discusses the topic of a discussion show in a social network. Emerging questions and points for discussion are captured later in the show.</td>
</tr>
</tbody>
</table>

**Table 13: Scenario Categorization**

1. **Level 2 - Critical:** scenarios where frustrated user and either reduced collaborative outcome (2a)* or impact on the TV content is caused by erroneous metadata (2b)**.

2. **Level 3 - Low Critical:** scenarios where erroneous metadata cause frustrated user.

3. **Level 4 - Not Critical:** nothing applies with the occurrence of erroneous metadata.

Notes: * impact on TV content is negligible in case of o **

Table 14 shows the rating of the previously defined scenario categories in terms of their level of criticality.

<table>
<thead>
<tr>
<th>Group Mode →</th>
<th>Open Group</th>
<th>Closed Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate</td>
<td>Level 2a</td>
<td>Level 1</td>
</tr>
<tr>
<td>Delayed</td>
<td>Level 3</td>
<td>Level 2b</td>
</tr>
</tbody>
</table>

**Table 14: Rating of the Categories in Terms of the Level of Criticality**
**n-Burst Loss**: a burst is defined by a block of \( n \) consecutive frames that gets lost. This happens, for example, when storms interfere with the broadcast signal for a certain time period.

A single error occurs if one frame of the data stream is lost or one single frame is damaged. A frame is damaged if single or bursts of bits are erroneous.

**Frame Damaged**: In general, a frame is damaged if one or several bits in the stream are erroneous.

**Single Error**: errors occurring singly within a data stream are called single bit errors. More exactly, one single bit tilted from 1 to 0 or reversed [6].

**n-Burst Error**: a burst is defined by a block of the length of \( n \) bits. Within this block, at least the first bit and the last bit are erroneous, although individual bits in between might be correct [6].

**Symbol Error**: a symbol error denotes one damaged symbol. A symbol typically has a length of 8 bits and up to 8 bit errors can occur in a random constellation [6].

*Figure 47: Types of Frame and Bit Errors*
Broadcast Delay  The broadcast delay is, technically speaking, not an error, but it can lead to errors in the collaboration. The following types of delay, have to be differentiated:

**Seven Second Delay**\(^9\): this is an intentionally caused delay of live material. The broadcast is delayed between 5 and 10 seconds, to prevent profanity, bloopers, violence or other undesirable material. This kind of delay is not problematic for the collaboration since it is the same for everybody, well-known and predictable.

**Unpredictable Delays**\(^10\): this refers to effects on the transmission path which can not be predicted or modified. To mention some examples (list is not complete), effects could be:

- **Compression**: Lower compression rate offers higher video quality but causes higher bandwidth and more data to be transmitted. Therefore, receivers of HDTV content may expect greater delay. [58]

- **Distribution**: The different distribution paths (for example satellite vs cable) make up the largest part of unpredictable delays. For instance, the satellite up-link takes longer because the microwave signal is transmitted into space, wrapped into the transponders of the providers and then beamed back to the satellite dishes of the viewers and finally decoded by the receivers. A two way propagation delay of 120ms occurs. This is in contrast to IPTV, where the access to the network takes around 100ms. [58]

- **Provider**: The content providers have their own proprietary methods of encoding and capturing the signals for transmission. This can introduce delays between different TV stations broadcasting the same live content.[58]

- **Customer**: Decoding, buffering and rendering steps introduce delays based on the digital TV and set-top box hardware.[58]

These causes of delays are problematic because they can not be avoided and can not be predicted. The authors of [58] measured delays up to 4000ms of a live transmission between different TV stations. The only way to handle these delays is to plan buffer time of, for example, 5000ms for the collaboration to be sure all collaborators receive the same information. The buffer time might be adapted to experience values.

5.3.4 Actions to be taken for the different Levels of Criticality

1. **Level 1 - Very Critical**

To compensate for lost frames (single and bursts):

- Status messages of each group are sent to the broadcaster automatically (including, for example, an index of received and processed metadata).

- Request for lost or damaged metadata frames to other group members.

- Within one TV show, send running index number to each frame.

- Redundancy.

---


To compensate for damaged frames (single, bursts or symbol errors):

- Build and send a checksum for each frame.
- Use existing and, in the broadcast sector, established forward error correction (FEC) algorithms to correct bit errors (single, burst and symbol errors).
- Redundancy

2. **Level 2 - Critical**

To compensate for lost frames (single and bursts):

- Redundancy.

To compensate for damaged frames (single, bursts or symbol errors):

- Build and send a checksum for each frame.
- Use existing and, in the broadcast sector, established forward error correction (FEC) algorithms to correct bit errors (single, burst and symbol errors).

3. **Level 3 - Low Critical**

To compensate for lost and damaged frames (single, burst and symbol errors):

- Redundancy.

The presented arrangements should constitute only recommendations, no prescriptions. To protect from Level 1 - Level 3 errors, redundancy was implemented for now.

### 5.4 Scenarios

To make a proof of concept, two apps for two scenarios were developed, implemented and tested. The first app “CBF Voting App” realizes a simple voting scenario, the second one “CBF Gaming App” realizes a quiz scenario.

#### 5.4.1 Voting (“CBF Voting App”)

In this scenario, collaborative activity in the form of voting is enabled to the viewer (cf. Figure 48). This scenario is very simple and, more or less, already exists in the form of telephone voting. This scenario serves as a first test case for the reference architecture, where a simple scenario is needed.

The goal is to offer voting possibilities to the viewer. The voting is enabled at a previously defined time, and must not happen during the corresponding show, but also before. Each viewer has the possibility to vote once. In addition, different user interfaces are available for different votes, for example via text field, radio buttons or check boxes. Chapter 6 comprises of a detailed implementation description of the voting scenario according to the procedure model below.

#### 5.4.2 Quiz (“CBF Gaming App”)

The quiz scenario offers several possible constellations of rivals and collaborations (cf. Figure 49). The focus of the scenario is to present the participants with several tasks to
be solved. The tasks could be anything, not only typical quiz questions. The crux of this scenario is the possibility of having participants live in the TV studio as well as at home in front of their TV sets, to compete with each other. In which constellation the participants collaborate or compete against each other is completely open and can be adapted to the underlying show. In our quiz scenario, participants in the studio form a group of three to five people and compete against collaborating teams in front of their TV sets. The collaborating teams also consist of three to five people, who might or might not know each other. In the first case, people build their own group and register. In the second case, registered participants are assigned to groups randomly. The challenge is, firstly, to find questions and tasks which can be solved by all participants,
5.5 PROCEDURE MODEL OF THE IMPLEMENTATION OF A CBF APPLICATION SCENARIO

regardless of whether they play in the TV studio or at home. Secondly, the result of the task must be able to be proven by the teams at home. Thirdly, providing an interface which supports the collaboration of the participants at home, especially if they act in randomly built groups and therefore the team members don’t know each other. Fourthly, the broadcast delay needs to be considered in the script of the show. The delay between the transmission of the TV station and receiving the contents at home is approximately 8 seconds, which needs to be considered. Fifth, the presentation of the teams and their results in the show because the TV viewers at home are, of course, not aware of who they are and what they do.

Chapter 6 comprises a detailed implementation description of the quiz scenario according to the procedure model below.

5.5 PROCEDURE MODEL OF THE IMPLEMENTATION OF A CBF APPLICATION SCENARIO

Figure 50: Developing a CBF Application Scenario as Part of the Content Value Chain [52]

The following procedure model serves as a step-by-step manual for the realization of collaborative broadcast content scenarios. All phases are situated in the production (pre- and post-production) phase of the collaborative broadcast content value chain (cf. Figure 50).

5.5.1 Conception Phase

Before starting the implementation, it is necessary to plan and conceive the broadcast content format, which usually happens in the pre-production phase (cf. Chapter 4). Herein the focus is on the metadata.

1. Firstly, the course of the narration structure and the proposal of the story need to be finished. At this point, a sketchy idea of the collaborative overlay (which collaborative activity, at which point in time, does which modification) exist.

2. Along with the composition of the treatments and the storyboard, the timestamps, impact and duration of effect of the metadata are determined. At this step it is not important to know the exact content of the metadata but it is important to know which type of metadata is included at which timestamp, as well as their range of validity and what they trigger.
3. Determine the trigger and tags to anchor: this phase includes, firstly, the determination of the structure of the metadata (in case of XML, the determination of the DTD) and, secondly, the determination of trigger and anchoring tags, which could be timestamps, interval, persons, games, rounds, or similar.

4. Metadata: design the exact content of the metadata relating to the previously determined structure. Two points need to be considered: first, one data frame can contain only 188 bytes. The XML data might be bigger; in this case the metadata are split to multiple consecutive data frames which takes computation time. Therefore, keep the metadata as simple as possible.

5.5.2 Implementation Phase

The implementation phase focuses on the development of the collaborative application. The look and feel of the collaborative application, as well as its functionality, is dedicated to the producer and strongly depends on the broadcast content format. To allow a fluent integration and cooperation with the reference architecture, the following six steps need to be taken. As mentioned above, the reference architecture is written in the C/C++ language whereas in the following, C/C++ implementation is expected.

1. Include the \texttt{CAManager} by using the following code:

   ```
   #include <CaManager.h>
   ```

   As mentioned, the \texttt{CAManager} manages to get the extracted metadata packages from the \texttt{Simple Player} and puts them into the corresponding queue into the shared memory. The inclusion of the \texttt{CAManager} is necessary to get access to the shared memory which is managed by this.

2. Add and adjust the Parser Library

   The \texttt{Parser} library needs to be available at runtime. The \texttt{Parser}'s main task is to parse the metadata when they are taken from the shared memory and processing them into the \texttt{CBFInfo} object. The \texttt{CBFInfo} object maps the XML metadata into the \texttt{C++} object structure, to simplify the further processing of the metadata. In addition, the objects provide functionality to take the metadata packages out of the shared memory. Further, the library needs to be adjusted to the metadata structure which was determined in the production phase. The adjustment needs to be done in the \texttt{CBFInfo} object and the \texttt{Parser} itself.

3. Implement the main process of checking the shared memory for metadata packages. The metadata packages are addressed with the unique identifier of the collaborative application. From the collaborative application's point of view, the shared memory includes two queues - firstly, a public queue and, secondly, a private queue. The public queue is available all the time and visible and readable for all running collaborative applications. The private queue is created by the collaborative application's startup and, therefore, only visible to and readable by the creator. In both cases (public and private queue acting as data carousel), the collaborative application is not allowed to write into the queues, but only to read and to delete metadata packages. The queues are filled by the \texttt{CAManager}. 
If the collaborative application is not running yet, but the CAManager received packages for it, they are put into the public queue. If the collaborative application is already running, packages are put into the private queue only. This is the reason why, at the program’s startup, the public queue has to be checked once for possible packages. After that, only the private queue needs to be checked periodically. Furthermore, the packages are provided with a version number, which is necessary to prevent from processing obsolete packages which might accidentally not have been deleted.

This can be done by doing the following or a similar code:

```c++
// initialize shared memory
managed_shared_memory sm_segment(open_only, "MySharedMemory");
// create a unique identifier for the collaborative application
const int id = 99;
// create any name for the collaborative application
const char* name = "99";
// initialize the current version counter to zero
int curr_pkt_version = 0;
// example procedure to access the shared memory and create the private queue
int GetSharedMemory(){
    // get public queue
    GenerelPacketList = sm_segment.find<PacketVector>("PacketQueue").first;
    const ShmemAllocator alloc_inst(sm_segment.get_segment_manager());
    // create private queue
    OwnPacketList = sm_segment.find<PacketVector>(name).first;
    if(OwnPacketList == NULL){
        OwnPacketList = sm_segment.construct<PacketVector>(name)(alloc_inst);
    }
    return 1;
}

// example procedure to check the public queue for metadata packages
CBFInfo* CheckGeneralPacketQueue(){
    // initialize the CBFInfo object
    CBFInfo *tmp = NULL;
    // search the public queue for packages marked with the unique identifier
    for(int i = 0; i < GenerelPacketList->size(); i++){
        // found one
        if(GenerelPacketList->at(i).id == id){
            // store metadata into the CBFInfo object structure
            tmp = getLocalCBFInfo(GenerelPacketList->at(i), &sm_segment);
            // push CBFInfo object into the private queue for further processing
        }
    }
    return tmp;
}
```
5.5 Procedure Model of the Implementation of a CBF Application Scenario

```c
33     OwnPacketList->push_back(GenerelPacketList
34          ->at(i));
35          // erase the package from the public queue
36     GenerelPacketList->erase(GenerelPacketList
37          ->begin()+i);
38 }
39
40 // example procedure to check the private queue
41 CBFInfo * CheckOwnPacketQueue(){
42     // initialize the CBFInfo object
43     CBFInfo *tmp = NULL;
44     // check if any package is in the private queue
45     if(OwnPacketList->size() > 0){
46         // if so, process all packages
47         for(int i = 0; i < OwnPacketList->size(); i++){
48             tmp = getLocalCBFInfo(OwnPacketList->at(i)
49                 ,&sm_segment);
50                 // not necessary to delete because if a
51                 // newer version of a package is
52                 // available, it is overwritten
53                 //OwnPacketList->erase(OwnPacketList->
54                 //begin()+i);
55         }
56     }
57     return tmp;
58 }
59
60 // example thread for the main procedure for checking the queues
61 // for packages
62 void *packet_thread(void *args){
63     // create a temporary CBFInfo structure to hold the XML
64     // metadata
65     CBFInfo* tmp;
66     // get the shared memory space from the CAManager
67     GetSharedMemory();
68     // check the general packet queue for possible packages
69     tmp = CheckGeneralPacketQueue();
70     // in case there is a packet:
71     if(tmp != NULL){
72         // check the version number of the packet in the
73         // queue with the current number
74         // to ensure that the packet is not obsolete
75         if(tmp->version > curr_pkt_version){
76             // notify your program about the available
77             // metadata
78             Notify(tmp);
79             // set the packet version
80             curr_pkt_version = tmp->version;
81         }
82     } else{
83 ```
4. Implement the graphical user interface and the collaborative functionality.

5. Implement further steps if necessary, for example steps in a game or similar.

6. Connect processed metadata to the logical procedures and steps of the program by the implementation of the `Notify()` method.

Note: Steps 4, 5 and 6 should be seen as iterative. They strongly depend on the program and are, therefore, not further explained.

5.5.3 Merging Phase

Finally, the created metadata of the production phase needs to be merged with the video source by using the Broadcaster/Producer Module. Afterwards, the sources are compiled to coherent content packages and are aggregated at program or schedule time. Important: the adjustment of the parser library (which is step 2 in the implementation phase) needs to be done before the merging process is possible.
This chapter critically demonstrates the concepts proposed in this thesis. The evaluation is split into four parts:

First (i), dedicated case studies were developed, presented and analyzed to show the applicability of the prototypical implementation of the reference architecture. Second (ii), it is a dedicated benchmark for testing previously defined parameters and questionnaires. Third (iii) a feature-based evaluation is done by a comparison of the proposed concepts to the related work on the basis of the framework discussed in Chapter 2.

6.1 Case Study-Based Evaluation

To evaluate the practicability of the implemented reference architecture, two application scenarios were developed and implemented using the procedure model presented in Chapter 5. The scenarios are presented in Chapter 3. The development of a collaborative broadcast format consists of three steps, namely the conception phase, the implementation phase and the merging phase, whereas the production phase is only necessary if a new broadcast show or broadcast content format is developed, which should be modified by the outcome of the collaborative application.

The first scenario is a simple voting scenario. Its purpose is to test and show the basic functionality of the reference architecture. More exactly, it shows the possibility of including and processing metadata into a broadcast. The focus in this scenario is on the collaborative application, not the broadcast content, since voting scenarios may happen outside of the dedicated content.

For the second scenario, both parts were developed, namely a concept for a collaborative TV format, and its corresponding collaborative application at the viewer’s side. The scenarios purpose is to figure out the upper limit of what is possible with the concept of merging metadata into the MPEG-TS.
6.1.1 Scenario 1 - Vote for the Top 333 Music Acts

The first scenario, realized with the previously described reference architecture, is a simple voting scenario. It serves as a first test environment to test the inclusion/extrusion and processing of metadata. The scenario was developed by following the previously presented procedure model. The conception phase was accomplished only partly and superficially, since the underlying content format supports a simple voting scenario, where the voting is done days before the show is aired. This is why several steps of the conception phase can be skipped.

In general, the voting scenario is a simple radio broadcast scenario where users can determine the top 333 music acts. The user can vote via different receiving units. For example if they are receiving the broadcast via an app on a smartphone, they can vote using the app, or, if they are receiving the broadcast via their television devices by using a DVB-H receiver, they can vote via their television device and so on. For the realization, the television scenario, receiving the radio show via DVB was used. To keep it simple, the time frame of the voting is one week. After the frame ends, the top 333 music acts are broadcast and presented in a radio show.

The goal of this scenario is to demonstrate the applicability of the developed concept and the developed reference architecture within a broadcast scenario.

6.1.1.1 Conception Phase

As defined in the procedure model, the development process starts with the conception phase.

**Plan the broadcast format and its progress:** the scenario is the presentation of the top 333 music acts in the context of radio shows broadcast before midday. Instead of playing any music acts, the shows present the result of voting the top 333 music acts. The time-frame for voting is from Monday to Saturday (six days). The presentation of the voting result starts on the Tuesday after. Mention the average number of aired music acts of a music channel as up to 13 songs per hour and a presentation time frame of 6 am to 12 am (6 hours per day), 78 voted songs can be aired per day. This allows the presentation of 312 voted songs until Thursday and having the grand finale of presenting the top 21 songs on Friday.

The voting happens one week before, where the presenters of the radio shows, where the top 333 music acts will be presented, should plug for the voting. The metadata to enable the voting should be broadcast whenever the presenter does some advertising for these purposes. The presenter does the advertising in addition to their usual program, which follows a previously defined schedule. In this scenario, the presenter should advertise every hour for voting for the top 333 music acts whereas once an hour the XML metadata, which enables the voting via a simple text field, should be broadcast. Further, the audience can vote via conventional ways like telephone or Internet at any time. The current top 10 of the voting are broadcast which should enable simplified voting via check boxes. If the voting is done, the notifier is disabled until the next metadata are broadcast.

*Note:* the checkboxes were chosen to demonstrate different GUI possibilities. For the suspense of the scenario, showing the current top ten will have contra-productive effects.
TAGS TO ANCHOR THE METADATA: the tags are simple timestamps in the schedule of the single radio shows in which advertising for voting of the top 333 music acts is done.

Note: for demonstration and testing purposes, the timeline was changed for the implementation. Metadata are not inserted every hour but at an interval of 10 - 20 Seconds.

METADATA: the voting activity is triggered by two types of metadata. Firstly to enable the textual voting via a simple textbox and secondly to enable the voting via checkbox by broadcasting the top 10 list. The enabler for the textual voting looks as follows:

```xml
<?xml version="1.0"?>
<application>
  <id>99</id>
  <guiEnabler>1</guiEnabler>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>0</guiUpdate>
  <name>CBF Voting Application</name>
  <repository>ftp://123.45.67.890</repository>
  <task>Vote for your Top 333 Music Acts</task>
  <textEntry>Actor - Song Title</textEntry>
</application>
```

This enabler just enables a simple text box where the listener can insert his or her favorite. To prevent from voting consecutively, the interface is disabled after the vote and enabled again when the next metadata are sent. The field id defines the unique application id which is set to '99'. The guiEnabler is set to '1', which means that it enables the notifier and, therefore, also the voting interface. The repository gives a repository where the voting application can be downloaded if it is not installed already. The name defines the name of the collaborative application. The task just includes a short instruction for the user. The textEntry is a keyword for the application to show a text field where the user can enter their vote and in the format they should enter it.

The second type of metadata enables a simpler voting by checkboxes.

```xml
<?xml version="1.0"?>
<application>
  <id>99</id>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>1</guiUpdate>
  <name>CBF Voting Application</name>
  <repository>ftp://123.45.67.890</repository>
  <task>Vote for your Top 333 Music Acts</task>
  <check>
    <value> Victim Of Love  Pospichal, Leonard & Th. Blug </value>
    <value> La La La  Naughty Boy Feat. Sam Smith </value>
    <value> Roar  Perry, Katy </value>
    <value> And We Danced  Macklemore Ft. Ziggy Stardust </value>
    <value> Love Me Again  Newman, John </value>
    <value> Whatever  Cro </value>
    <value> Applause  Lady Gaga </value>
    <value> Hey Now  Solveig, Martin & The Cataracs </value>
    <value> Burn  Goulding, Ellie </value>
  </check>
</application>
```
The description of the XML tags is the same as described before, except for the definition of the check box list. The tag **check** defines a keyword for the collaborative application to show a list of checkboxes using the values that are defined with the tags **value**.

### Implementation Phase

1. Include the CAManager.

2. Adjust the CBF object structure to the previously presented metadata.

   ```cpp
   struct CBFInfo{
       /* Program Parameter */
       int id;
       int guiEnabler;
       int packetUpdate;
       int guiUpdate;
       std::string name;
       std::string repository;

       /* Gui */
       // Textfield to display the task to viewer
       std::string task;
       // to take the input of the textentry
       std::string vote;

       // values for radio buttons
       std::vector<std::string> rb_Values;
       // values for check boxes
       std::vector<std::string> cb_Values;
   };
   ```

Adjust the parser library to the CBF object structure:

```cpp
CBFInfo* walk(xml_node parent){
    CBFInfo *info = new CBFInfo();
    xml_node app = parent.child("application");

    info->id = atoi(app.child_value("id"));
    info->packetUpdate = atoi(app.child_value("packetUpdate"));
    info->guiUpdate = atoi(app.child_value("guiUpdate"));
    info->name = app.child_value("name");
    info->repository = app.child_value("repository");

    info->task = app.child_value("task");
    /* get text entry */
    info->vote = app.child_value("vote");

    /* get values for radio buttons */
    xml_node radio = app.child("radio");
    for(xml_node tmp = radio.child("value"); tmp; tmp = tmp.
        next_sibling()){  
        info->rb_Values.push_back(string(tmp.child_value()));
    }

    /* get values for check boxes */
```
If the packetUpdate value is set to 0, the metadata packet is a newer version of an already sent packet. If it is 1, the packet is stored into the shared memory. This step should prevent from keeping multiple versions of one packet. Note: the presented code is a dump of the whole implementation.

3. The main thread is more or less the same as presented in the procedure model. It leads in a notifier to build up the graphical user interface depending on the metadata.

```c
void *main_thread(void *args){
    // create a temporary CBFInfo structure to hold the XML metadata
    CBFInfo* tmp;
    // get the shared memory space from the CAManager
    GetSharedMemory();
    // check the general packet queue for possible packages
    tmp = CheckGeneralPacketQueue();
    // in case there is a packet:
    if(tmp != NULL){
        // notify your program about the available metadata
        Notify(tmp);
    }
    // Check the programs own packet queue periodically for new packets
    for(;;){
        tmp = CheckOwnPacketQueue();
        if(tmp != NULL){
            Notify(tmp);
        }
        sleep(rand() / (RAND_MAX / 3) + 1);
    }
    return NULL;
}
```

The general packet queue is called once before the collaborative application is started. The CBFPlayer collects extracted metadata packages which corresponds to inactive collaborative applications in this queue. For active applications, the packages are put into the private queue.

4. The graphical user interface (cf. Figures 52, 53, 54 and 55) is built up depending on the metadata as the following code demonstrates. The application is very simple and holds three possibilities for the user to do the voting, namely via a text box, via radio buttons or via check boxes. For the scenario, the voting via text box and
check boxes are used. The implementation of this scenario differs essentially from
the second scenario. Because of the simplicity of the voting procedure, the structure
and the content of the graphical user interface are sent via metadata. But this is only
recommendable for standard scenarios which are as simple as the voting is, otherwise
the metadata become too comprehensive. The advantage is, that the scaffold of the
voting application can be used for different voting scenarios since, as previously
mentioned, the structure and the content is sent via metadata. The user does not
have to install several kinds of voting applications.

```c
void Notify(CBFInfo* info){
    gdk_threads_enter();

    /* Set Iterator */
    gtk_text_buffer_get_start_iter(buffer, &start);
    gtk_text_buffer_get_end_iter(buffer, &end);
    gtk_text_buffer_delete(buffer, &start, &end);
    gtk_text_buffer_get_iter_at_offset(buffer, &iter, 0);

    /* Text View */
    if(!info->text.empty()){
        gtk_text_buffer_insert_with_tags_by_name(buffer, &
    iter, info->text.c_str(), -1, "blue_fg", "centering_text", NULL);
    }

    /* Radio Buttons */
    if(info->rb_Values.size() > 0){
        /* erase previous elements */
        for(int i = 0; i < radioValues.size(); i++){
            gtk_widget_destroy((GtkWidget*)radioValues .at(i));
        }
        radioValues.erase(radioValues.begin(), radioValues .end());

        /* insert new elements */
        radioValues.insert(radioValues.begin(),
            gtk_radio_button_new_with_label(NULL, info->
        rb_Values.at(0).c_str()));
        group = gtk_radio_button_group (GTK_RADIO_BUTTON (radioValues.at(0)));
        gtk_toggle_button_set_state (GTK_TOGGLE_BUTTON ((
            GtkWidget*)radioValues.at(0)), TRUE);
        gtk_table_attach_defaults((GtkTable*)table_layout, (GtkWidget*)radioValues.at(0),1,7,3,4);
        gtk_widget_show((GtkWidget*)radioValues.at(0));
        for(int i = 1; i < info->rb_Values.size(); i++){
            radioValues.push_back(
                gtk_radio_button_new_with_label(group,
                info->rb_Values.at(i).c_str()));
            group = gtk_radio_button_group (GTK_RADIO_BUTTON (radioValues.at(i)));
```
6.1 Case Study-Based Evaluation

gtk_table_attach_defaults((GtkTable*)
table_layout, (GtkWidget*)radioValues.
at(i), 1, 7, i+3, i+4);

gtk_widget_show((GtkWidget*)radioValues.at
(i));

}  
else{
    for(int i = 0; i < radioValues.size(); i++){
        gtk_widget_destroy((GtkWidget*)radioValues.
at(i));
    }
    radioValues.erase(radioValues.begin(), radioValues.
.end());
}

/* Check Boxes */
if(info->cb_Values.size() > 0){
    /* erase previous elements */
    for(int i = 0; i < checkValues.size(); i++){
        gtk_widget_destroy((GtkWidget*)checkValues.
at(i));
    }
    checkValues.erase(checkValues.begin(), checkValues.
.end());

    for(int i = 0; i < info->cb_Values.size(); i++){
        checkValues.insert(checkValues.begin()+i,
gtk_check_button_new_with_label(info->
cb_Values.at(i).c_str()));
        gtk_table_attach_defaults((GtkTable*)
table_layout, (GtkWidget*)checkValues.
at(i), 1, 7, i+3, i+4);
        gtk_widget_show((GtkWidget*)checkValues.at
(i));
    }
}  
else{
    for(int i = 0; i < checkValues.size(); i++){
        gtk_widget_destroy((GtkWidget*)checkValues.
at(i));
    }
    checkValues.erase(checkValues.begin(), checkValues.
.end());
}

/* Text Entry */
if(!info->textEntry.empty()){
    textEntry = (GtkWidget*)gtk_entry_new();
    gtk_entry_set_editable(GTK_ENTRY(textEntry), TRUE)
    ;
    gtk_entry_set_text(GTK_ENTRY(textEntry), info->
textEntry.c_str());
6.1.1.3 Merging Phase

The merging happened by using the Broadcaster/Producer Module, as described in the previous chapter (cf. Chapter 5). The medium is a standard MPEG-TS container. The only exception is the missing video stream, whereas the synchronization happens with the audio stream. The MPEG-TS container included an mp2 audio stream and a data stream with the following parameter:

Stream #0.0[0x100]: Audio: mp2, 44100 Hz, stereo, s16, 128 kb/s
Stream #0.1[0x101]: Data: [6][0][0][0] / 0x0006

As mentioned, for testing issues, the metadata are inserted at an interval of 10 to 20 Seconds. Figure 51 gives an overview of the timeline of the inserted metadata.

![Figure 51: Program Progress Including Metadata for a Voting Scenario](image)

00:10 After the first metadata packet is extracted by the player at timestamp 00:10, it is parsed and the program id is extracted. The CBFManager module checks if the corresponding application is already running, which would mean that a packet queue for this application would already exist in the shared memory space. If not, the packet is stored in the general packet queue, in case the application is started at a later time. If yes, the packet is stored in the applications packet queue, which is named by the unique id of the application. Simultaneously, the packet is checked if its guiEnabler is set to “1”. If yes, the graphical user interface of the player is notified and notifies the viewer. The appearance of the viewer notification is specific to the implementation. In this prototypical implementation, the notification is shown as additional button (cf. Figure 52).
In addition, the metadata contains content for the voting application to build up a text field where people may fill in their choice (cf. Figure 53).

Another enabler for a new voting round is extracted. This enabler also includes content for the collaborative application to build up a multiple choice surface as shown in Figure 54.
If desired, an enabler can be written, which also includes content for the collaborative application to build up a radio button surface as shown in Figure 55.

<?xml version="1.0"?>
<application>
  <id> 99 </id>
  <packetUpdate> 1 </packetUpdate>
  <guiUpdate> 1 </guiUpdate>
  <path> ../../CA__Voting </path>
  <name> CBF Voting Application </name>
  <repository> ftp://123.45.67.890 </repository>
  <task> Eurovision Song Contest 2011 Final Voting </task>
  <radio>
    <value> Ireland - Jedward - "Lipstick" </value>
    <value> Sweden - Eric Saade - "Popular" </value>
    <value> Estonia - Getter Jaani - "Rockefeller Street" </value>
    <value> Ireland - Jedward - "Lipstick" </value>
    <value> Sweden - Eric Saade - "Popular" </value>
    <value> Estonia - Getter Jaani - "Rockefeller Street" </value>
  </radio>
</application>

Figure 55: Screenshot of the CBF Voting Application

In summary, the metadata, inserted in the medium and the program line, includes metadata to change the collaborative activity. The collaborative activity in this case is the voting, therefore the metadata control the appearance, structure and content of the graphical user interface.

The given example is highly compressed to one minute progress of the radio broadcast. This was done for demonstration purposes.

6.1.2 Scenario 2 - Game/Quiz Show

For the second scenario, the concept of a TV content format as well as the corresponding collaborative application were developed, whereas the full development process, as described in the procedure model, was passed.

The second scenario is a TV content format for a game show. Two teams compete against each other in the studio, by solving different tasks from answering questions to games of skill. The highlight of this format is the additional participation of teams, single players and the bulk at home in front of their TV sets who battle against the teams in the studio.

The goal of this scenario is to test the upper limits of the proposed concept. The challenge of this scenario is the synchronization and the coordination of the collaborating teams at home in a live TV scenario.

For the first step of producing the TV content format, eight steps from the producer’s point of view, defined by Boljets [16], were followed.

Note: The production process used in this chapter differs slightly from the one presented with the reference architecture. The process presented in the reference architecture is for universal application but, for the evaluation, a TV format is a concept which demands a variation of the production process.

For the sake of completeness, in the following all steps of the process of TV format development are described in a nutshell [37], although only the first two steps were finished in this thesis.
1. Idea Paper - is the result of the idea generation. This is a very unregulated phase of the format development process. The paper just describes the basic idea of the TV format.

2. Concept Paper - the concept paper is more concrete than the idea paper. It includes explanations of the sequence and production, as well as the description of the concept and the verification of the eligibility on the TV market.

3. Pilot Book - refers to a concrete episode of the TV format; the episode is completely described.

4. Pre-Test - is the first field test to lower the risk of a flop. For example, game principles and tasks of a game show are acted out with subjects.

5. Pre-Pilot - in the pre-pilot the focus is on the layout and the timing. The pilot book episode is played through and recorded, which helps to identify unintended length or other necessary changes.

6. Pilot - is a record of the conceived format which is able to be broadcast.

7. National Broadcast

8. International Application

A TV format is only allowed to be called a "TV format" if all steps of the development process are passed successfully [37]. But, due to the technical focus of this thesis and the lack of knowledge, time and resources, the format development process was done only as far as the concept paper. Therefore, the development is called a conceptual paper format of “Against the Others - Interactive”. In the following, the idea and the concept paper of “Against the Others - Interactive” is presented. Since the focus of this thesis is in computer science, there is no claim of being complete in the description of the concepts of the TV format. There is also no possibility to carry out the further steps of the development process, as it is not possible to test the concept of the TV format. It does, however, allows for the

Figure 56: Substantiation Process of Format Development [37]
inclusion and testing of the possibility of linking the TV format with the viewer at home by putting metadata into the stream. Therefore, it allows an estimation of the feasibility of a collaborative TV scenario and it allows the proving of the technical feasibility.

6.1.2.1 “Against the Others - Interactive” - Idea Paper

The basic idea of “Against the Others - Interactive” is a game show between two teams in the studio, each consisting of three candidates, and the TV viewer at home in front of their TV sets. It is an adaption of the show '1 against 100' (or “Eén tegen 100” created by Endemol), which was broadcast in Austrian television in 2008. In the original program format, one nominee played a quiz against a group of 100 physically present participants. In the adapted show the group of participants is enlarged to an open group of participants who are not physically present. The tasks to solve in the game show range from simple questions to answer or phrases to guess to more complex tasks to solve, like writing a collaborative text or solving game skills. TV viewers at home build small groups of three members as well. The difficulty for the teams at home is that the teams are built automatically, which means team members don't know each other but have possibilities to communicate and collaborate via the collaborative application. In addition, the TV viewers have the possibility to participate anonymously or to register just for fun. In the case of anonymous participation, the participants results are not sent to the TV station. The registered but single participants compete against each other, are finally ranked and have the possibility to win prizes for each game. A notary supervises the happenings in the studio.

Resumed, the following modes exist:

1. **Single Anonymous User**: This mode allows everybody to participate without any collaboration and registration. Example: Voting or single participation in quizzes.

2. **Single Registered User**: In this mode, the participant needs to be registered but is not collaborating actively with others. Implicit collaboration might happen in simple collaboration scenarios like voting, where the opinions of the users are collected, but users do not interact and collaborate actively. Example: Voting or single participation to quizzes.

3. **Groups**: This mode allows incorporation of groups with a defined number of members. The groups can be formed previously by the collaborators, or single registered users can be merged to groups automatically. Example: Small groups participate against a candidate or in serious games.

4. **The Bulk**: acting as loose and temporary limited alliance with limited collaborative interaction because of the mass of participants. An example could be bulk bets against a candidate or a team in the TV studio.

The team who solves the task faster or better, collects points. The team with the most points wins the game show. The challenge is to define the tasks in a way that the results of the collaborators at home are able to be proven because it is, of course, not possible to send a notary to every participant’s home.
6.1.2.2  Concept Paper

The basic idea was already mentioned in the idea paper and is quite simple. Two teams of, respectively, three people in the studio are fighting against each other and against an arbitrary number of teams at home with the following facts:

- Genre: Game Show.
- Frequency: Weekly.
- Duration: 60 minutes gross (including commercials), 51 minutes 40 seconds net.
- Time Slot: Prime Time.
- Production: Block Production.
- Category: Entertainment.
- Target Group: 18 - 49 Years, technically adept.

The show is live, although the data stream including the metadata is prepared previously.

**Title**  In general, the title should suggest what the contents of the show are and, in our case, it should also bring attention to the possibility to participate collaboratively from the home TV sets. The title of the conceived TV format is “Against the Others - Interactive”.

**Short Description**  One team, consisting of three to five members, competes live against N teams of collaborators at home. The home teams consist of three to five members, who are grouped automatically. The teams at home do not know each other but they have, of course, possibilities to communicate via the collaborative application. The candidates in the studio do not know each other. Single people can apply and are chosen by an independent jury.

The teams play against each other and have to solve previously defined tasks. The team that solves the task faster or better collects points and wins the task. The team that collects the most points is the winner. In addition, TV viewers at home can participate as single players. As mentioned, this single participation can be anonymous or registered. Anonymous players participate just for fun because no results are transmitted to the TV station, but they are not listed in the rankings and can not win any prizes. For some tasks, the so-called ‘bulk’ is challenged. Single, registered players can act as a unit - the bulk, for example to bet against the playing teams or similar. This concept is based on the TV format "1 gegen 100" ("1 against 100") which was aired on Austrian, German, Swiss and Dutch TV between 2002 - 2008. The format was originally designed by Endemol, with the original title "Eén tegen 100". In our case the mass of “100” consists of N teams of collaborators in front of their TV sets.

The winning team wins the main prize, with the other participating teams dividing a cash prize or winning a consolation prize.

**Tasks to Solve**  In the following, a list of nine tasks is presented which must be solved within one to two episodes of the game show. Of course, new tasks must be developed for

---

1 Some tasks concern to those presented in the TV show “The Cube” [http://www.itv.com/thecube](http://www.itv.com/thecube)
6.1 Case Study-based Evaluation

Each subsequent episode. The challenge is to find tasks which can be solved by both the team in the studio and the teams at home. For the teams at home, the results need to be verifiable and the tasks need to be easy to understand. Research work in the Internet must not be helpful.

1. “Quiz” - A phrase or a question to guess is presented to the players. In the case of a phrase, the players take turns to guess (the player whose turn it is to guess is decided) letters. They have, in total, eleven attempts to guess all letters or the phrase. In the case of a question, one question after the other is presented to the players. The players guess alternatively (the player whose turn it is to guess is decided). If the question is answered correctly, the team gets one point, if not, the team gets no point.

2. “Write a Story” - a topic or a word/number/picture is given, the teams have to write any text relating to this. The players write alternatively (the player whose turn it is to write is decided). The best story is elected by the viewers.

3. “Sort” - 10 cylinders of varying diameters are given. To win this game, the players must place the cylinders in size order, smallest to largest. They have 12 seconds to complete this game. It is decided which player in a team starts. At intervals of 4 seconds, the token goes to the next player in the team who must continue the task. This game needs to be implemented for the players at home.

4. “Bet against X” - the team in the studio/the teams at home bet against the teams at home/the team in the studio if they can solve a given task within a given time period or if they can solve it in general, how many points they will reach, or similar.

5. “1 against the Bulk” - the team in the studio guesses how many teams/players at home can solve a given task or can answer a given question.

6. “Speed” - On the surface of a given console are 9 buttons. The aim is to extinguish the lights as they appear. When one is extinguished the next appears. 20 lights must be extinguished in 10 seconds to win the game. This game needs to be implemented for the players at home.

7. “Cubes” - 27 cubes must be transferred into a container which is just large enough to fit them all. Only one cube can be transferred at a time. The time-frame is 20 seconds. This game needs to be implemented for the players at home.

8. “Accurate” - A circular button must be pressed on a console exactly 20 times within five seconds. If the button is pressed fewer or more than 20 times, points are lost. A maximum 20 points can be reached. This game needs to be implemented for the players at home.

9. “Recognize” - A geometric form appears. After five seconds it will vanish. Five similarly geometric forms (in case of size, color, form) will then appear. The player must answer which geometric form was the correct one to win the game points. This game needs to be implemented for the players at home.

---

2 Concerns to “Eén tegen 100” created by Endemol [http://www.kro-ncrv.nl/eentegen100/370-35705-de-een-tegen-100-app](http://www.kro-ncrv.nl/eentegen100/370-35705-de-een-tegen-100-app)

3 Concerns to the game presented in the TV show “The Cube” [http://www.itv.com/thecube](http://www.itv.com/thecube)
“Against the Others - Interactive” is an up-to-date entertainment format, which can be classified in the category of game shows. The classification is unambiguous, since the show contains game elements that must be solved by the candidates, which builds the major element of the show.

“Against the Others - Interactive” addresses the mass of viewers during prime time. In addition, the target group of the show is comprised of, firstly, those who want to participate and those who just want to watch. The participants should be between the ages of 18 - 49 and technically adept, so as not to have any problems with the handling of the collaborative application. The age range is chosen because two requirements must be met: firstly, the majority age to be allowed to participate in a gaming show. The majority must be proven during the registration process by any legal id card which can be read over a card reader. In Austria this can be, for example, a citizen identification or the student id card. Secondly, the participants need a basic understanding of the handling of their TV sets which exceeds turning the TV on an off, switching the channel and the volume adjustment. For those who want to watch only, the show addresses participants between the ages of 16 to 49. Any further understanding is not necessary.

The unique selling position should cover the advantages of the format and why it is advantageous to the TV station if they buy the format. The outstanding features of the TV show are clearly definable. The following points bring advantages to the TV station and are therefore obvious sales arguments:

- Game with competition.
- Participation prevents from switching channels.
- The audience is thrilled, can participate if desired and take sides.
- Time pressure, unknown results of the games, especially due to consideration of the teams at home building up tension.
- Game shows are in vogue.
- A variation of the show, with prominent participants in the studio, is possible. Participants at home have the possibility to battle against prominent candidates.

For the developed TV format, four different characters are needed, i.e. the presenter, the jury, the participants in the studio and the participants at home.

- Presenter - the presenter plays an important part in a TV show. He or she is, more or less, the face of the show.
- Jury - the jury observes the gaming in the background and only interferes in the event of any mistake or cheating happening during a game.
- Participants in the studio - it is up to the participants if the show is boring, amusing or entertaining. They decide if the TV viewers at home are thrilled and entertained and if they finally stay on the ball or if they change the channel. In this show, one to
two teams are playing against each other and/or playing against the teams at home. It is important to choose both female and male candidates between the age of 18 and 49.

- Participants at home - addresses TV viewers between the ages of 18 and 49 with technical basic knowledge.

6.1.2.4 Schedule

Usually, the schedule is a written paper including mandatory stage directions for one transmission date for all involved. Order and chronology must be evident [37]. The challenge in the schedule is to plan the teams that are acting at home. Slack time for the delay, as well as for technical problems, needs to be considered. The overall duration of the show is 60 minutes gross (including breaks) which are approximately 51 minutes 40 seconds net + slack time of the TV station for possible timeouts. Below (cf. Table 15, an example schedule is presented (in minutes).

<table>
<thead>
<tr>
<th>Trailer</th>
<th>Duration:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trailer</td>
<td>0’15</td>
</tr>
<tr>
<td>First Moderation</td>
<td>0’45</td>
</tr>
<tr>
<td>Team Presentation</td>
<td>2’00</td>
</tr>
<tr>
<td>Team building at home</td>
<td>0’10</td>
</tr>
<tr>
<td>Rules</td>
<td>2’00</td>
</tr>
<tr>
<td>Countdown expires</td>
<td>1’00</td>
</tr>
</tbody>
</table>

**Duration:** 6’10

<table>
<thead>
<tr>
<th>Lap # 1 - Quiz</th>
<th>Duration:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present the rules of game 1</td>
<td>1’00</td>
</tr>
<tr>
<td>A given phrase needs to be guessed by the teams within 4 minutes. In each round, one letter can be guessed by one candidate. The candidate who guesses is chosen randomly, whereas one candidate may guess several times consecutively. A maximum of eleven errors are allowed. If they where made, the team loses one point. If one team guessed the phrase, the game is over, the team gets the point. All other teams get no point. Game starts.</td>
<td>4’00</td>
</tr>
<tr>
<td>After the game ends, a short debriefing takes place.</td>
<td>2’00</td>
</tr>
</tbody>
</table>

**Duration:** 7’00
### Lap # 2 - Write a story

Present the rules of game 2.

The teams write a text collaboratively within 6 minutes. This means, to a given word or topic, one team member writes text. After 30 seconds, the next team member has 10 seconds to read the predecessor’s text, and he or she gets a new word or topic and must resume the text. The team members take consecutive turns. When the game is over, the participants at home (not the participating teams but the un-/registered single users) evaluate the texts and give between 0 and 10 points. There is no winner in this round, the collected points are added to the total points. The participants at home have time until the end of Lap # 4 to evaluate the texts. To encourage people to evaluate the texts, they can win a prize.

Game starts.

After the game ends, a short debriefing takes place.

**Duration:** 6'00

### Lap # 3 - Sort

Present the rules of game 3.

10 cylinders of varying diameters are given. To win this game, the teams must place the cylinders in size order, smallest to largest. They have 20 seconds to complete this game. It is decided which player in a team starts. After 5 seconds, the player within a team change randomly. The other players who have not had a turn are not allowed to see the cylinders. The team who manages to place the cylinders in order first is the winner and gets the point.

Game starts.

After the game ends, a short debriefing takes place.

**Duration:** 4'20

### Lap # 4 - 1 against the bulk

Present the rules of game 4

10 questions are given, which must be answered by the bulk at home. The bulk represents registered users. To avoid the users from using a search engine to answer the questions, only 5 seconds are left. 4 possible answers are given. The teams in the studio guess what percentage of the users can answer each question. They guess and confer within the 5 seconds, the users have to guess the question. The winner is the team who is closest and gets one point.
Game starts (including 20 seconds of time between the questions).
After the game ends, a short debriefing takes place.
**Duration:** 8'10

**Lap # 5 - Cubes**

Present the rules of game 5.
27 blue cuboids need to be sorted into a box, which is just large enough to fit them all. The players act together and have 2 minutes in which to solve the task. The team who solves the task first is the winner and gets extra points. The team gets one point for each cuboid.
Game starts
After the game ends, a short debriefing takes place.
**Duration:** 6'00

**Winner of Lab #2**

Determine the winning team of Lab #2.
**Duration:** 1'00

**Lap # 6 - Accuracy**

Present the rules of game 6.
This is the final game, where the teams can collect a maximum of 90 points. A rotating button must be pressed 30 times within 30 seconds. Each player of a team gets a random turn. Three rounds are played in a total time of 90 seconds. If the button is pressed fewer or more than 30 times in 30 seconds, points are lost.
Game starts
After the game ends, a short debriefing takes place.
**Duration:** 5'30

**Trailer**

The winner of the teams in the studio and of the teams at home is determined and gets their prizes. The participants at home who participated in Lap # 2 who get the prize are estimated and displayed.
Sneak preview and information on the next show
Last moderation
closing titles
**Duration:** 3'30

**Total Time of “Against the Others - Interactive”** 51'40

Table 15: Schedule of the Designed Collaborative Broadcast Content Format
6.1.2.5 Setting

For the sake of completeness, the design of the setting is described briefly. The show takes place in a television studio, which is helpful for the recognition and orientation of the show because the setting remains constant. The setting impresses with a clear and simple look, and is constructed as in game shows commonly used. As Figure 57 shows, the gaming area, which holds the games, is situated in the center. On the base stations above, the teams get together after each game for conversations and the presentation of the results. The main stage on the top is also used for presentation of prizes and a live act. This is necessary since the gaming area needs to be reconstructed after each game. The seating capacity for the live audience is situated at the bottom, with the giant main screen on top, showing the results of the studio teams. The results of the collaborating teams at home are shown on the secondary screens. The jury follows the games from the backstage area.

6.1.2.6 Progress of the TV Format

The progress of the TV format is about an expansion of the previous presented schedule, including the exact timing position, id and class of the metadata. The progress is presented in Figure 58. Figure 58 sketches the timeline of the TV format at fixed intervals of one minute. The total length of the format is 51’40 minutes, whereby a lead time and a slack time are recommendable (marked by the dashed line). A lead time of a minimum of 10 minutes is recommendable because the registration and team building process takes time, which cannot be fully considered in the show. A slack time of at least 10 minutes is recommended as well in case of delays within the games or technical problems as this is common in live game shows.

Additionally, the laps and trailers are of contrasting colors for a better legibility of the progress diagram. The trailers are marked in gray, the laps are marked alternately in violet and turquoise colors. The colored triangles mark metadata to be included at the
6.1 Case Study Based Evaluation

Figure 58: Progress of “Against the Others - Interactive” Including the Position and Ids of the Metadata

given timestamp. With the id, the metadata can be identified as they are listed in Figure 58. The green triangle marks the enabler, which could enable, firstly, the collaborative application itself or, secondly, the games at the given lap. The red triangle marks the disabler, which could disable the collaborative application itself or the game at the given lap. The blue triangle marks new content for the collaborative application or for the game, as for example rules or a new question for a game. The orange triangle marks metadata to steer the collaboration within the teams, for example to determine who is next in line. The orange dashed arrow, including the given seconds, marks at which intervals coordinating metadata should be included in the stream within the given period. The thicker orange line tells if the notifier in the player is visible and for which user mode it is visible. The group building process, for example, is only available until the first lap starts. After this point, only single users can participate, whether they register or not. In the following, an overview of the included metadata is given.
### Table 16: Schedule of the XML Metadata Merged for this Scenario

<table>
<thead>
<tr>
<th>#</th>
<th>ID</th>
<th>Timestamp</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ca_e</td>
<td>-11'00</td>
<td>Enable the collaborative application</td>
</tr>
<tr>
<td>2</td>
<td>ca_r</td>
<td>-10'00</td>
<td>Send/show the rules of the game show</td>
</tr>
<tr>
<td>3</td>
<td>g_d</td>
<td>5'10</td>
<td>Disable the group building process</td>
</tr>
<tr>
<td>4</td>
<td>01_r</td>
<td>6'00</td>
<td>Send/show the rules for game #1</td>
</tr>
<tr>
<td>5</td>
<td>01_e</td>
<td>7'10</td>
<td>Enable game #1</td>
</tr>
<tr>
<td>6</td>
<td>01_c</td>
<td>7'10</td>
<td>Coordinators for game #1 each 10 seconds for 4'00 minutes</td>
</tr>
<tr>
<td>7</td>
<td>01_d</td>
<td>12'10</td>
<td>Disable game #1</td>
</tr>
<tr>
<td>8</td>
<td>02_r</td>
<td>13'00</td>
<td>Send/show the rules for game #2</td>
</tr>
<tr>
<td>9</td>
<td>02_e</td>
<td>15'10</td>
<td>Enable game #2</td>
</tr>
<tr>
<td>10</td>
<td>02_c</td>
<td>15'10</td>
<td>Coordinators for game #2 each 30 seconds for 6'00 minutes</td>
</tr>
<tr>
<td>11</td>
<td>02_d</td>
<td>21'10</td>
<td>Disable game #2</td>
</tr>
<tr>
<td>12</td>
<td>02a_e</td>
<td>21'11</td>
<td>Enable voting application to vote for the best story</td>
</tr>
<tr>
<td>13</td>
<td>03_r</td>
<td>23'00</td>
<td>Send/show the rules for game #3</td>
</tr>
<tr>
<td>14</td>
<td>03_e</td>
<td>25'10</td>
<td>Enable game #3</td>
</tr>
<tr>
<td>15</td>
<td>03_c</td>
<td>25'10</td>
<td>Coordinators for game #3 each 5 seconds for 0'20 minutes</td>
</tr>
<tr>
<td>16</td>
<td>03_d</td>
<td>25'30</td>
<td>Disable game #3</td>
</tr>
<tr>
<td>17</td>
<td>04_r</td>
<td>27'00</td>
<td>Send/show the rules for game #4</td>
</tr>
<tr>
<td>18</td>
<td>04_e</td>
<td>29'30</td>
<td>Enable game #4</td>
</tr>
<tr>
<td>19</td>
<td>04_c</td>
<td>29'30</td>
<td>Coordinators for game #4 each 5 seconds for 4'10 minutes</td>
</tr>
<tr>
<td>20</td>
<td>04_d</td>
<td>33'40</td>
<td>Disable game #4</td>
</tr>
<tr>
<td>21</td>
<td>05_r</td>
<td>35'00</td>
<td>Send/show the rules for game #5</td>
</tr>
<tr>
<td>22</td>
<td>05_e</td>
<td>37'40</td>
<td>Enable game #5</td>
</tr>
<tr>
<td>23</td>
<td>05_c</td>
<td>37'40</td>
<td>Send one coordinator for game #5</td>
</tr>
<tr>
<td>24</td>
<td>05_d</td>
<td>39'40</td>
<td>Disable game #5</td>
</tr>
<tr>
<td>25</td>
<td>02a_d</td>
<td>41'40</td>
<td>Disable voting application and present results</td>
</tr>
<tr>
<td>26</td>
<td>06_r</td>
<td>42'30</td>
<td>Send/show rules for game #6</td>
</tr>
<tr>
<td>27</td>
<td>06_e</td>
<td>44'40</td>
<td>Enable game #6</td>
</tr>
<tr>
<td>28</td>
<td>06_c</td>
<td>44'40</td>
<td>Coordinators for game #6 each 30 seconds for 1'30 minutes</td>
</tr>
<tr>
<td>29</td>
<td>06_d</td>
<td>46'10</td>
<td>Disable game #6</td>
</tr>
<tr>
<td>30</td>
<td>ca_i</td>
<td>48'10</td>
<td>Send concluding information</td>
</tr>
<tr>
<td>31</td>
<td>inf01_e</td>
<td>50'40</td>
<td>Send/show information for the upcoming show</td>
</tr>
<tr>
<td>32</td>
<td>inf01_d</td>
<td>51'00</td>
<td>Disable information for the upcoming show</td>
</tr>
<tr>
<td>33</td>
<td>ca_d</td>
<td>52'00</td>
<td>Disable notifier for all user modes</td>
</tr>
</tbody>
</table>

*Note: if additional or new information is sent or shown depends on the implementation. Sending means that the information itself is included in the metadata. Showing them*
means, that only an event is sent by metadata to the collaborative application to know that it has to display the information to the users.

6.1.2.7 Tags to Anchor Metadata

Tags for this scenario are timestamps as listed in Table 16.

6.1.2.8 Metadata

In contrast to the voting scenario, the triggering metadata for this scenario will not contain any content for the collaborative application but events to display the corresponding content. This is done to broadcast only a minimum of metadata and to keep the number of split packages as small as possible. Therefore, only the following patterns are used:

- Enabler - this pattern serves as an enabling event for the collaborative application to enable the application itself or to enable different contents to the user, like rules or games.

- Disabler - this pattern serves as a disabling event for the collaborative application to disable the application itself or to disable different contents to the user, like rules or games.

- Coordinator - this pattern supports the coordination of the team members who act at home. In this scenario, the coordinating information is very simple and mostly includes which team member has a turn in a game.

- Updater - this pattern supports sending content for the enabled application.

All metadata are fully listed and sorted in relation to their timestamp and id, as listed in Table 16.

6.1.2.9 Implementation Phase

The implementation phase was done according to the presented procedure model.

1. Include the CAManager

   ```cpp
   #include <CaManager.h>
   ```

2. Adjust the Parser Library

   First, the CBFInfo object structure of the parser library was designed concerning to the defined metadata.

   ```cpp
   struct CBFInfo{
   /* Program Parameter */
   int id;
   int guiEnabler;
   int packetUpdate;
   int guiUpdate;
   int round;
   std::string name;
   std::string repository;
   ```
The parser library was adjusted in relation to the defined metadata.

```c
void parseCBFInfo(CBFInfo *info){
  if(info->id == 1){ // Introduction
    // showIntro();
  }
  else if(info->id == "01_e") { // Round #1
    // showQuestion Round 1
    // check for steps
  }
  else if(info->id == "02_e") { // Round #2
    // ...
  }
  else if(info->id == "03_e") { // Round #3
    // ...
  }
  else if(info->id == ca_i){ // Conclusion
    // showConclusio()
  }
}
```

Note: the presented code is just a dump of the whole implementation.

3. Implement the main thread
The main thread is, more or less, the same as presented in the procedure model in Chapter 5.

```c
void *main_thread(void *args){
  // create a temporary CBFInfo structure to hold the XML metadata
  CBFInfo* tmp;
  // get the shared memory space from the CAManager
  GetSharedMemory();
  // check the general packet queue for possible packages
  tmp = CheckGeneralPacketQueue();
}
6.1 Case study-based evaluation

The general packet queue is called once before the collaborative application is started. The CBFPlayer collects extracted metadata packages which correspond to inactive collaborative applications in this queue. For active applications, the packages are put into the private queue.

4. Implement the graphical user interface

The following Figures (cf. Figure 59 - Figure 63) sketch the graphical user interface of the gaming scenario. Basically, the same methodology as in the voting application was used, which split the window into several parts, managed by the processed metadata. For this, several notifiers which are triggered by the metadata were implemented. Because of its scope, it was decided not to list any implementation code. Enabler

![Image](image_url)

**Figure 59:** Screenshot of the CBFPlayer Showing the Notifier to the Gaming Application

and disabler patterns activate (cf. Figure 59) or deactivate (cf. Figure 61) the whole application or single areas of the application (cf. Figure 60). The updater pattern updates the content of single areas of the application, for instance the main area showing the gaming rules (cf. Figure 62). The coordinator pattern activates or deactivates single areas or functionality of the application for single peers (cf. Figure 63 and Patterns “Floor Control” and “Schedule Collaborative Activity”). Note: the player to watch the broadcast content format was scaled-down since the gaming has
6.1 Case Study-Based Evaluation

Figure 60: The Enabler Triggers Game #1 in the Gaming Application

Figure 61: The Disabler Concludes and Terminates the Gaming Application

Figure 62: The Updater Refreshes the Main Area of the Gaming Application Showing the Gaming Rules

priority. If the gaming application is closed, the player scales to its original size. For demonstration, any video material was taken.

5. Further steps
People can decide at the beginning whether they want to participate anonymously, as single player or within a group. As single or group players, registration to the community (cf. Patterns “Create Community” and “Registration”) of the broadcast content format is required (cf. Figure 64). In both cases, the full name, screen name,
score and ranking of the player or the group is sent and further processed by the broadcaster. It is the broadcaster’s decision if the screen name and the results are published, stored or further processed or not but the player must comply. As anonymous players, no identities, inputs, scores, results or similar are sent to the broadcaster or published (cf. Pattern “Masquerade”).

In case of group players, the group building phase happens next (cf. Patterns “Create Group”). The group building can be done manually or automatically (cf. Patterns “Build Group Manually” and “Build Group Automatically”). In the former, the group building works similarly to social networks. The player sends invitations (cf. Pattern “Invitation”) to others or chooses them from a list of other registered players who are searching for a group (cf. Pattern “Quick Join”). In the latter, the player is attached to a partial group automatically (cf. Pattern “Quick Join”). For this, players and partial groups are managed in a database. If players allow tracking via GPS or tell their location otherwise, one criteria for group building can be the neighborhood, but this is not implemented. The group members are aware of each other by a buddy list (cf. Pattern “Buddy List”), the groups are aware of each other by a group list which is projected by one of the secondary screens in the studio (cf. Pattern “Group List”). Since no TV studio was available, this feature only exists in theory but is not implemented.

Furthermore, it is necessary to implement communication functionality. In this scenario a simple chat is provided (cf. Patterns “Embedded Chat”).

For consistency, the pattern of “Replicated Objects” connected with “Decentralized Updates” was realized. Each peer holds his/her own copy of the object (the object is, for instance, the phrase which must be guessed). Only one peer is allowed to guess at one time; whose turn it is is coordinated by the metadata. The guess is sent to all
other peers by a short message.

**Note:** not all games were implemented. If so, more complex solutions for collaboration, coordination and communication have to be realized.

### 6.1.2.10 Merge

The merging process is quite simple, as described above, and is done by using the *Broadcaster/Producer Module*. The metadata are merged into the MPEG-TS and synchronized with the video stream in relation to the timestamps given in the schedule above.

### 6.1.3 Result of the Case Study based Evaluation

Two scenarios were realized to test the feasibility of the concept and the realized reference architecture. Both scenarios were able to be realized with the reference architecture and the procedure model.

The purpose of the simple voting scenario was to test the applicability of the concept of including metadata into the MPEG-TS and their processing in real-time at the viewer’s side. The interoperability of the developed concept is clearly shown, although several deficiencies are conceivable. In general, the merging, analysis and the processing of the metadata works in more adequate duration. The exact duration is determined with the benchmark tests. In addition, for this simple scenario, simple metadata are sufficient. Since synchronization of the viewers and synchronization to the broadcast content is not necessary for this scenario, no attention is paid to delays.

A critical point on this scenario is the insufficient consideration of the collaboration because, in the voting scenario, the viewers/collaborators act autonomously; the sum of their decisions gives some indication of the result. This is the reason for reading a little into the applicability of the developed concept concerning collaborative purposes. Furthermore, the advantage of this voting scenario towards the conventional telephone and Internet voting is doubtful. One advantage is surely the independence of the broadcast content at the point of voting. For example, for an arbitrary broadcast content in the future, different voting can be done and it is not necessary for the voters to change to another platform. It is also not necessary for the broadcasters to give information about the voting during different content. This can be done with the metadata and the collaborative application. The dynamic and time dependent features of this kind of voting apply properly also in the telephone and Internet voting.

The gaming scenario is far more complex because the concept TV format is designed for a live broadcast. In this case, the big problems of the synchronization and the real-time error treatment need to be faced. Another problem of a complex scenario is the rising complexity of the metadata. Because of the rising complexity of the topic, the focus was set on proving the functionality, more exactly proving the technical feasibility of linking broadcast content and collaborative activity. Requirements to quality of service issues are future prospects.
6.2 **FEATURE-BASED EVALUATION**

In this section, a comparison of the proposed concepts to dedicated related work is done. The comparison is based on the framework developed and presented in Chapter 2, which was used to classify related work. The framework is used to classify and hereafter to evaluate the presented work. In the following, the criteria of the framework are recapped shortly. A detailed description of the criteria is presented in Chapter 2.

The classification and evaluation of the developed scenarios is outlined in the following:

### 6.2.1 Narration Space

For this purpose, a narrative is defined as a series of events that are linked together in a number of ways, including cause and effect, time and place. Something that happens in the first event causes the action in the second event, and so on (usually moving forward in time) [35]. The Narrative Space may be *linear*, *branched*, *non-deterministic* or *evolutionary*.

The narrative space of the developed scenarios is not beyond the scope of branched structure, because it was not in focus and beyond the resources of this thesis to develop revolutionary content. But, with the scope of the reference architecture, the use of all narrative structures is possible. As the voting scenario demonstrates, a collaborative application for linear narrative structure is as possible as the use of more interactive structures, as the second scenario demonstrates.

### 6.2.2 CMC Link

The CMC Link is about the level of connectivity between the collaborative activity and the broadcast medium respective to the content. There may exist *No CMC Link*, *CMC Link to the Broadcast Medium* or *CMC Link to the Program Content*.

In this thesis, the link is realized with the reference architecture by the usage of XML metadata which are merged to the MPEG-TS. The functionality of the concept is proven by the development of two working scenarios. The critical point in case of synchronized and live collaboration within a television area is the broadcast delay (beyond the seven second delay), which is hardly predictable but can be solved by planning delay buffers in the program structure. The CMC Link to content is possible and given as well which is shown by scenario #2 with the prototypical TV content format. The CMC Link of collaborative activity to the program content combined with the link to the broadcast medium allows for modifying the broadcast content in manifold possibilities. The CMC Link to the broadcast medium was barely focused on in this thesis, which is a point of critics. The possibility of modifying the broadcast medium by collaboration was proven by the scenarios.
6.2.3 Collaboration

6.2.3.1 Scope of Collaboration

The category Scope of Collaboration describes the aim of the provided collaboration functionality in relation to the consumed (play back) broadcast medium. Whether it is just modified collaboratively by changing its state commonly, or it is created collaboratively, like annotating existing content commonly, or the broadcast medium is used to provide and initiate interactive collaborative actions, the connectivity may be to Distribute, Create new Content, for Social Activity or to Modification of the Broadcast Medium/Content.

The scope of collaboration is not limited to one feature, for example either social activity or interaction, as scenario #2 demonstrates. Each kind of collaborative activity (beyond pure interactivity) can be linked with the broadcast content format, as long as the already mentioned synchronization demand is complied with. Collaboration to modify the broadcast medium is not given in existing solutions but is possible as the developed scenarios prove.

6.2.3.2 Synchronization

The critical point of existing solutions was the lack of synchronous collaborative activity. Synchronous collaboration is mostly limited to simple mechanisms like voting procedures. In case of more complex mechanisms (like using blog entries) the collaboration is asynchronous between the collaborators and asynchronous to the broadcast content. Scenario #2 has demonstrated the technical feasibility of using synchronized complex collaboration mechanisms. A missing point is doing a real life broadcast scenario in this point to check the practicability as well.

6.2.4 Others

The development of a reference architecture allows for the realization of highly diverse scenarios. Technically, a broadcast content format with all types of narrative spaces can be combined with all kinds of collaborative activity, modality and synchronization. The technical feasibility of integrating synchronized collaboration into any kind of broadcast content format is shown. One major shortcoming of this evaluation is the missing field trial in real life broadcast scenario. The field trial was missing because of the time and resource issues.

6.3 Benchmark-based Evaluation

One important part of the evaluation of the developed reference architecture, as well as the implemented scenarios, is the benchmark based evaluation. Thus, in the following, benchmark criteria and parameters that depict the scopes of performance, functional requirements and robustness in case of errors are presented.
6.3 Benchmark-Based Evaluation

6.3.1 Performance Evaluation

In the performance evaluation, the system performance is evaluated in relation to conventional non-functional requirements which are significant for the broadcast and collaboration sectors. As performance requirements, real-time capability and resource utilization were determined.

The experiments were done on the previously implemented gaming scenario.

*Figure 65: Timeline of the Shortened Quiz Scenario*

The scenario was shortened for the measurements to a movie sequence of 2 minutes 34 seconds. Two experiments were conducted. Firstly, the metadata were included constantly every 15 seconds (Experiment #1) and, secondly, the metadata were included irregularly partly with a minimal gap of one second to check if the system can deal with that (Experiment #2). Testing the inclusion of metadata at an interval of one second is important to test the possibility of redundancy as an error protection mechanism. For both experiments, the first seven metadata packets of the game show timetable presented in Table 16, were included into the shortened movie sequence, plus the final disabler for the collaborative application at the end. Figure 65 presents the timeline of the experiments and the ids of the included metadata. Some packets need to be split since their memory size exceeds the maximum capacity of one MPEG-TS frame which is 188 bytes (cf. Grey triangles in Figure 65). The included metadata are listed in Appendix D.

For the voting and the gaming scenario, it applies that the measurement results from the shortened test sequence can be implied to the further process of the scenarios broadcast content format and metadata.
6.3.1.1  **Real-Time Capability**

This is one of the most important requirements because of the time critical broadcast and the time critical collaboration. The real-time capability is validated according to the following parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration of Metadata Extraction</td>
<td>This parameter relates to the reception and extraction of the metadata. It starts with the reception of the broadcast stream, detecting the metadata stream and extracting and synchronizing the metadata packages for the player.</td>
</tr>
<tr>
<td>Synchronization of the Teams</td>
<td>This concerns the synchronization of the collaborative teams at home and in the studio. It includes the handling of the broadcast delays and the consistency between the teams at home and in the studio.</td>
</tr>
<tr>
<td>Preparation of the clients</td>
<td>The duration of the viewer (clients) to get ready to attend the broadcast content is one important factor and includes the preparation of the receiving unit (turn on, receive the broadcast), the installation of the collaborative application, reading and understanding the purpose, rules and the goal of the broadcast content (“What do I have to do?”) and, if applicable, the group building process.</td>
</tr>
<tr>
<td>Network delay</td>
<td>Average network delay when collaborators communicate.</td>
</tr>
</tbody>
</table>

**Duration of Metadata Extraction**  The metadata extraction happens and is fully measured at the client side. The metadata extraction consists of the following four steps:

1. Extraction of the MPEG-TS data frames (which include the metadata) from the MPEG-TS data stream into a data picture queue. The data picture queue holds data frames using the first-in-first-out principle. If a metadata packet was split into several MPEG-TS frames, the frames are assembled to one data frame in the queue.

2. If a video stream exists, check with each displayed video frame if the next element in the data picture queue is a matching data frame. A data frame matches if its pts >= pts of the current video frame and if its pts < pts of the next video frame. If no video stream exists, the synchronization is done with the audio stream.

3. If a data frame matches and is synchronized to the current video frame, it is extracted from the queue and committed to the collaborative application.

4. The data frame enables or disables a collaborative application or it is processed by the corresponding collaborative application.

This overall extraction process, from steps 1 to 3, is the same for all metadata patterns, whether they are, for instance, enablers, disablers or updater. This extends to the measurement as well. Step 4, which is the individual processing phase of the collaborative application certainly differs subject to the pattern.
The measurements were done by using a simple measurement algorithm:

```c
startTime = CurrentTime()
Do_some_process()
endTime = CurrentTime()
TimeDifference = endTime-startTime
```

More exactly, the Linux standard function gettimeofday is used.

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <unistd.h>
int main()
{
    struct timeval st , et;
    gettimeofday(&st , NULL);
    sleep(2);
    gettimeofday(&et , NULL);
    printf("Total time taken is : %lu seconds and %lu microseconds\n", (et.tv_sec - st.tv_sec),(et.tv_usec - st.tv_usec));
    return 0;
}
```

The test setup was the following:

<table>
<thead>
<tr>
<th>CPU</th>
<th>Intel®Core™2 Duo CPU T9300 @ 2.50GHz x 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics</td>
<td>GeForce 8400M GS/PCIe/SSE2</td>
</tr>
<tr>
<td>RAM</td>
<td>3.9 GiB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Ubuntu 12.02 32 bit</td>
</tr>
</tbody>
</table>

The metadata and the detailed results on the measurement can be found in Appendix D. Measurements of each step were done separately. 5 runs were done for each step for a test period of 2 minute 34 seconds, whereas the measurements stop after the last metadata package. In total, the percentage of time which is needed to do the metadata extraction (including Steps 1 - 3) and processing (including Step 4) was calculated to evaluate the real-time capability of the reference architecture and the implementation.

Tables 17 - 20 show the overall results, including (cf. Table 18 and 20) and excluding Step 2 (cf. Table 17 and 19). The differentiation was done since Step 2 is about the permanent checking of the PTS of the next data frame in the queue to the PTS of the current displayed video frame. In general, this is a good solution because it allows a fine grained inclusion of data frames with the frame rate of the video frame. But checking the PTS causes a permanent overhead as long as data frames are in the queue, whereas it is important to check this overhead separately. If this overhead passes a critical threshold, there is the potential for optimization.

The time exposure for the metadata extraction and processing was calculated in percentage over the period of time until the next metadata package was scheduled. For instance, in the case of the first metadata package, the average time exposure for extraction and
<table>
<thead>
<tr>
<th>Timestamp</th>
<th>1st Run</th>
<th>2nd Run</th>
<th>3rd Run</th>
<th>4th Run</th>
<th>5th Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.04</td>
<td>326.30</td>
<td>188.20</td>
<td>363.20</td>
<td>131.30</td>
<td>142.00</td>
</tr>
</tbody>
</table>
|           | 0.054023| 0.031159| 0.060132| 0.021738| 0.023510| ms
| 21.04     | 178.40  | 265.10  | 272.60  | 232.30  | 536.20  |
|           | 0.011893| 0.017673| 0.018173| 0.015487| 0.035747| %
| 36.04     | 55.10   | 71.00   | 92.20   | 95.00   | 102.00  |
|           | 0.003673| 0.004733| 0.006147| 0.006333| 0.006800| %
| 51.04     | 167.50  | 119.50  | 76.60   | 107.90  | 194.30  |
|           | 0.003673| 0.004733| 0.006147| 0.006333| 0.006800| %
| 67.04     | 131.20  | 56.70   | 145.20  | 153.10  | 185.60  |
|           | 0.008200| 0.003544| 0.009075| 0.009569| 0.011600| %
| 81.04     | 135.10  | 94.87   | 161.40  | 139.50  | 128.50  |
|           | 0.009650| 0.067764| 0.011529| 0.009964| 0.009179| %
| 96.04     | 68.00   | 94.10   | 60.70   | 101.20  | 91.10   |
|           | 0.004533| 0.006273| 0.004047| 0.006747| 0.006073| %
| 111.04    | 131.00  | 89.90   | 132.80  | 146.70  | 151.90  |
|           | 0.008733| 0.005993| 0.008853| 0.009780| 0.010127| %

Table 17: Metadata Extraction Experiment #1 (Total Duration Excluding Step 2)

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>1st Run</th>
<th>2nd Run</th>
<th>3rd Run</th>
<th>4th Run</th>
<th>5th Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.04</td>
<td>348.3</td>
<td>210.4</td>
<td>384.8</td>
<td>154.5</td>
<td>164.00</td>
</tr>
</tbody>
</table>
|           | 0.057666| 0.034834| 0.063709| 0.025579| 0.027152| %
| 21.04     | 247.9   | 336.3   | 352.6   | 302.9   | 612.70  |
|           | 0.016527| 0.022420| 0.023507| 0.020193| 0.040847| %
| 36.04     | 127.1   | 143.4   | 168.8   | 169.6   | 168.70  |
|           | 0.008473| 0.009560| 0.011253| 0.011307| 0.011247| %
| 51.04     | 236.6   | 188.8   | 146     | 178.3   | 260.20  |
|           | 0.015773| 0.012587| 0.009733| 0.011887| 0.017347| %
| 67.04     | 206.9   | 131.8   | 221.4   | 230.8   | 256.90  |
|           | 0.012931| 0.008238| 0.013838| 0.014425| 0.016056| %
| 81.04     | 198.2   | 1011.7  | 225.3   | 201     | 192.00  |
|           | 0.014157| 0.072264| 0.016093| 0.014357| 0.013714| %
| 96.04     | 137.7   | 161.3   | 128.3   | 171.2   | 155.80  |
|           | 0.009180| 0.010753| 0.008553| 0.011413| 0.010387| %
| 111.04    | 206.9   | 159.2   | 205.2   | 217.1   | 217.20  |
|           | 0.013793| 0.010613| 0.013680| 0.014473| 0.014480| %

Table 18: Metadata Extraction Experiment #1 (Total Duration Including Step 2)
<table>
<thead>
<tr>
<th>Timestamp ↓</th>
<th>1st Run</th>
<th>2nd Run</th>
<th>3rd Run</th>
<th>4th Run</th>
<th>5th Run</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.04</td>
<td>241.9</td>
<td>208.7</td>
<td>685.0</td>
<td>308.3</td>
<td>213.9</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.118578</td>
<td>0.102304</td>
<td>0.335784</td>
<td>0.151127</td>
<td>0.104853</td>
<td>%</td>
</tr>
<tr>
<td>3.04</td>
<td>124.7</td>
<td>132.7</td>
<td>139.7</td>
<td>195.1</td>
<td>233.2</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.1247</td>
<td>0.1327</td>
<td>0.1397</td>
<td>0.1951</td>
<td>0.2332</td>
<td>%</td>
</tr>
<tr>
<td>4.04</td>
<td>57.7</td>
<td>50.8</td>
<td>46.2</td>
<td>43.7</td>
<td>96.1</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.0577</td>
<td>0.0508</td>
<td>0.0462</td>
<td>0.0437</td>
<td>0.0961</td>
<td>%</td>
</tr>
<tr>
<td>7.04</td>
<td>99.5</td>
<td>106.7</td>
<td>131.9</td>
<td>194.4</td>
<td>125.7</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.033167</td>
<td>0.035567</td>
<td>0.043967</td>
<td>0.0648</td>
<td>0.0419</td>
<td>%</td>
</tr>
<tr>
<td>8.04</td>
<td>147.6</td>
<td>129.9</td>
<td>67.2</td>
<td>82.5</td>
<td>57.3</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.1476</td>
<td>0.1299</td>
<td>0.0672</td>
<td>0.0825</td>
<td>0.0573</td>
<td>%</td>
</tr>
<tr>
<td>9.04</td>
<td>117.4</td>
<td>166.4</td>
<td>76.6</td>
<td>77.6</td>
<td>172.1</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.1174</td>
<td>0.1664</td>
<td>0.0766</td>
<td>0.0776</td>
<td>0.1721</td>
<td>%</td>
</tr>
<tr>
<td>11.04</td>
<td>118.7</td>
<td>46.8</td>
<td>86.3</td>
<td>39.3</td>
<td>41.1</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.05935</td>
<td>0.0234</td>
<td>0.04315</td>
<td>0.01965</td>
<td>0.02055</td>
<td>%</td>
</tr>
<tr>
<td>12.04</td>
<td>128.4</td>
<td>98.6</td>
<td>58.2</td>
<td>209.6</td>
<td>81.1</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.1284</td>
<td>0.0986</td>
<td>0.0582</td>
<td>0.2096</td>
<td>0.0811</td>
<td>%</td>
</tr>
</tbody>
</table>

Table 19: Metadata Extraction Experiment #2 (Total Duration Excluding Step 2)

<table>
<thead>
<tr>
<th>Timestamp ↓</th>
<th>1st Run</th>
<th>2nd Run</th>
<th>3rd Run</th>
<th>4th Run</th>
<th>5th Run</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.04</td>
<td>245.5</td>
<td>213.2</td>
<td>688.6</td>
<td>312.5</td>
<td>217.9</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.120343</td>
<td>0.104510</td>
<td>0.337549</td>
<td>0.153186</td>
<td>0.106814</td>
<td>%</td>
</tr>
<tr>
<td>3.04</td>
<td>18.27</td>
<td>22.57</td>
<td>22.67</td>
<td>25.41</td>
<td>32.02</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.01827</td>
<td>0.02257</td>
<td>0.02267</td>
<td>0.02541</td>
<td>0.03202</td>
<td>%</td>
</tr>
<tr>
<td>4.04</td>
<td>14.67</td>
<td>11.58</td>
<td>11.02</td>
<td>14.87</td>
<td>15.51</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.01467</td>
<td>0.01158</td>
<td>0.01102</td>
<td>0.01487</td>
<td>0.01551</td>
<td>%</td>
</tr>
<tr>
<td>7.04</td>
<td>25.55</td>
<td>27.57</td>
<td>30.79</td>
<td>34.74</td>
<td>29.97</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.008517</td>
<td>0.009190</td>
<td>0.010263</td>
<td>0.011580</td>
<td>0.009990</td>
<td>%</td>
</tr>
<tr>
<td>8.04</td>
<td>23.36</td>
<td>19.39</td>
<td>13.22</td>
<td>15.55</td>
<td>11.43</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.02336</td>
<td>0.01939</td>
<td>0.01322</td>
<td>0.01555</td>
<td>0.01143</td>
<td>%</td>
</tr>
<tr>
<td>9.04</td>
<td>18.84</td>
<td>23.04</td>
<td>13.76</td>
<td>14.76</td>
<td>21.61</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.01884</td>
<td>0.02304</td>
<td>0.01376</td>
<td>0.01476</td>
<td>0.02161</td>
<td>%</td>
</tr>
<tr>
<td>11.04</td>
<td>21.57</td>
<td>15.28</td>
<td>20.23</td>
<td>13.83</td>
<td>17.21</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.010785</td>
<td>0.00764</td>
<td>0.010115</td>
<td>0.006915</td>
<td>0.008605</td>
<td>%</td>
</tr>
<tr>
<td>12.04</td>
<td>18.64</td>
<td>16.16</td>
<td>12.72</td>
<td>27.16</td>
<td>14.71</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td>0.01864</td>
<td>0.01616</td>
<td>0.01272</td>
<td>0.02716</td>
<td>0.01471</td>
<td>%</td>
</tr>
</tbody>
</table>

Table 20: Metadata Extraction Experiment #2 (Total Duration Including Step 2)
processing is between 0.0304% and 0.112% of the total time of 6.037967 seconds (which calculates from the beginning until the first metadata package was fully processed).

Including the overhead of Step 2, it is between 0.0342% and 0.116% of the total time of 6.037967 seconds. The overhead of Step 2 is in the range of per mill (‰) and therefore noncritical. In any event, Step 2 has the potential to be optimized, for example, not to check each PTS which is, in the case of the presented scenarios, an overhead of 29.97 frames and operations per second.

The maximum size of a metadata package, and therefore also the maximum number of data frames, is limited to the following factors:

- The timestamp in the broadcast schedule when the metadata must be fully processed.
- The PTS of the next included metadata.
- The size of the MPEG transport stream for the broadcast.

The recommendation is to keep the metadata packages as short as required and not to exceed 1% of the total time for one metadata package not to bring down the viewer's set-top-box.

In the worst case, the extraction of a metadata package takes a tenth of a percent of the period until the next metadata package is scheduled. Even in the second experiment, where the packages were scheduled with the minimum distance of one second, the extraction takes at a max a tenth of a percent of the duration until the next package is scheduled. Therefore it is safe to say that the designed CMC Link mechanism, by the inclusion of metadata, is valid for the broadcast area because the synchronization of data to video is not as sensitive as audio to video. The ITU⁴ tested that the threshold for detecting AV synchronization errors is -125ms to +45ms⁵. In the case of data frames, the threshold on detecting data to video or data to audio synchronization can be in the range of seconds, depending on when the content of the data package is scheduled in the broadcast content format.

The limited size and complexity of the metadata is not assessed as a shortcoming. In theory, several smaller packages can be included with the same frame rate as the video stream, which is mostly around 30 frames per second. The current limitation of the minimum distance of one second is caused by the Broadcaster/Producer Module whose user interface does not support finer granulation. Supporting finer granulation is one open issue for future work. In addition, the complexity of collaborative broadcast scenarios is limited as well, which is caused by the characteristics of the broadcast itself like the handling of delays or similar.

**Synchronization of the Teams** The synchronization of the collaborative teams at home and in the studio requires two steps, firstly the handling of the broadcast delays and, secondly, the consistency between the teams at home and in the studio.

---

⁴ International Telecommunication Union

Two types of broadcast latencies need to be handled, namely the predictable profanity delay (or seven-second delay) and unpredictable further delays caused by transmission and decoding circumstances.

Kooij et al. measured that regular TV broadcasts differ by up to 6 seconds in the play-out [49]. Based on this, an overall delay of 10 seconds is implicitly planned in the program schedule of Scenario 2 to comprise most of the delay.

Addressing synchronization and consistency, two types have to be differentiated - firstly, synchronization and consistency within the collaborators in a team and, secondly, synchronization and consistency to the team(s) in the TV studio (for instance the duration of a challenge needs to be the same for both parties).

For the first case, several of the reference architecture defined patterns were implemented to keep the members of a team consistent and synchronized (cf. Tables 21 and 22).

For the second case, a planned 10 second delay, combined with the use of status messages which are sent by the teams to the TV studio, is done. The progress of the teams at home can be projected onto a secondary screen in the TV studio to be visualized for passive viewers. The following messages were realized:

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ready</td>
<td>Collaborative group is ready (all peers are connected and alive).</td>
</tr>
<tr>
<td>alive</td>
<td>The collaborative group is alive.</td>
</tr>
<tr>
<td>challenge started</td>
<td>The collaborative group started the challenge. From now on,</td>
</tr>
<tr>
<td></td>
<td>the group has n seconds left to solve the challenge.</td>
</tr>
<tr>
<td>challenge stopped</td>
<td>Time is over, the group stopped the challenge.</td>
</tr>
<tr>
<td>peer died</td>
<td>One peer in the collaborative group died.</td>
</tr>
<tr>
<td>peer added</td>
<td>A new peer is added to the collaborative group.</td>
</tr>
<tr>
<td>bye</td>
<td>The collaborative group says goodbye.</td>
</tr>
</tbody>
</table>

**Preparation Time of the Client**  
Assuming that the player is already installed and ready-to-use, the preparation time includes the subscription to participation, the download, installation and start of the collaborative application. In the case of group participation, the group building process (whether it happens manually or automatically) needs to be considered as well.

Although the preparation time of the client is important for the real time capability, the resources to build up user test environments were not given. It is recommended to do this in future work (cf. Chapter 7).

**Network Delay**  
For measuring the network delay, a full test environment including at least three peer to peer setups need to be build up. Similar to the preparation time of the client, the resources for building up this complex scenario were not given. It is recommended to do this in future work (cf. Chapter 7).
6.3.1.2 Resource Utilization

The system runs mostly on set-top-boxes, which have limited resources. For the resource evaluation, the system performance of a Linux based set-top-box, called dreambox DM7080HD, which is an up to date set-top-box available in Austria, is assumed.

![Dreambox 7080HD](image)

**Figure 66: Dreambox 7080HD**

The DM7080HD has the following specification:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Dreambox OS</td>
</tr>
<tr>
<td>API</td>
<td>Dreambox API</td>
</tr>
<tr>
<td>CPU</td>
<td>7000DMIPS Dual Core Processor</td>
</tr>
<tr>
<td>Video Out</td>
<td>2x HDMI 1.4a port</td>
</tr>
<tr>
<td>Decoder</td>
<td>MPEG-2 / H.264 Decoder</td>
</tr>
<tr>
<td>DVB</td>
<td>Twin DVB-S2 Tuner</td>
</tr>
<tr>
<td>Flash</td>
<td>4 GByte Flash</td>
</tr>
<tr>
<td>RAM</td>
<td>2 GByte Ram</td>
</tr>
<tr>
<td>Hard Drive</td>
<td>Support for 3.5 inch internal hard drive</td>
</tr>
<tr>
<td>Internet</td>
<td>10/100/1000 MBit/s Ethernet Interface</td>
</tr>
</tbody>
</table>

The resource utilization is validated based on the following criteria:

<table>
<thead>
<tr>
<th>Resource Utilization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average CPU Usage</td>
<td>Measures the average CPU Usage for both experiments.</td>
</tr>
<tr>
<td>CPU Usage during</td>
<td>Measures the average CPU Usage for both experiments during the metadata extraction/insertion.</td>
</tr>
<tr>
<td>Metadata Extraction</td>
<td></td>
</tr>
<tr>
<td>Average Memory Usage</td>
<td>Measures the average Memory Usage for both experiments.</td>
</tr>
<tr>
<td>Memory Usage during</td>
<td>Measures the average Memory usage for both experiments during the metadata extraction/insertion.</td>
</tr>
<tr>
<td>Metadata Extraction</td>
<td></td>
</tr>
</tbody>
</table>

---

6 [http://www.dream-multimedia-tv.de/dm7080-hd](http://www.dream-multimedia-tv.de/dm7080-hd)
To measure the CPU and memory usage, a small shell script was used:

```bash
#!/bin/bash
#
# CPU and Memory Measurement Script
#
# Execute CBFPlayer
./CBFPlayer
#
# Get Process ID of CBFPlayer
pid=`pgrep CBFPlayer`

if [ -n "$pid" ]
then
    taskset -cp 0 $pid
    python log.py $pid > logFile.txt
else
    echo "Process is not running."
    break
fi
```

The shell script was necessary to, firstly, execute the CBFPlayer, secondly, to detect its process id, thirdly, to assign the process to one CPU core with id 0 (the measurements were done on a dual core system) and, finally, to start the python script (see below) which was doing and logging the measurements to a file. A measurement point was every half a second, which results in 316 measurement values each time for CPU and memory. The number of measurement values consists of a lead time of 2 - 4 seconds, 308 measurement points during the test sequence and a follow-up time of 1 - 2 seconds. The lead time is necessary to detect the process id of the player, assigning the CPU and starting the python script before the player starts opening and processing the test sequence.

```python
import time
import string
import sys
import commands

def get_cpumem(pid):
    d = [i for i in commands.getoutput("ps aux").split("\n")
        if i.split()[1] == str(pid)]
    return (float(d[0].split()[2]), float(d[0].split()[3])) if d else None

if __name__ == '__main__':
    if not len(sys.argv) == 2 or not all(i in string.digits for i in sys.argv[1]):
        print("usage : %s PID" % sys.argv[0])
        exit(2)
    print("CPU ( % )	 MEMORY ( % )")
    try:
        while True:
            x = get_cpumem(sys.argv[1])
            if not x:
                print("Process ID is not valid")
            exit(1)
```
To determine a reference value, the measurements were made on the test sequence without any metadata included, firstly, with the developed CBFPlayer and, secondly, with the VLC media player\(^7\) which is an established video player for Windows, Apple and Linux systems. The results are shown in Figures 67 and 68. The CPU and memory usage for both players are similarly allocated. Of course, the VLC player is more efficient in the CPU usage, which might be as a result of better development work.

![Figure 67: CPU and Memory Usage of the CBFPlayer when Running the Test Sequence without Metadata Included](image)

To determine the CPU and memory usage of the metadata extraction (Steps 1 - 3), the previously defined experiments were used. The source of Experiment #1 is a 2'34

\(^7\) [http://www.videolan.org/vlc/index.html](http://www.videolan.org/vlc/index.html)
minutes test sequence with metadata packages included at a regular distance of 15 seconds (cf. Appendix D). The source of experiment #2 is the same test sequence with metadata packages included at an irregular distance of 1 to 3 seconds (cf. Appendix D).

The average CPU and memory usage during the metadata extraction (Steps 1 - 3) were summarized in Figure 69 for Experiment #1 and in Figure 70 for Experiment #2. The average memory usage is constant, between 0.5% and 0.7%. The average CPU usage between 9.5% and 12.1%. No significant peaks could be observed for the metadata processing nor any significant difference to the reference measurements without any metadata was detected. Based on these results, the sufficiency and practicability of the developed CMC Link mechanism is assumed. In addition, it can be presumed that the prototypical implementation can be performed on the above listed or lowered powered set-top-boxes, considering the mentioned requirements to real-time and delays. For future work, optimization should be done to lower the average CPU usage.
6.3.2 Requirements Evaluation

In Chapter 3 mandatory and advisable functional requirements were defined for the three modules Consumer Client Module, Broadcaster/Producer Module and Broadcaster Server Module. In the following, the compliance of the deduced requirements is verified.

**Consumer Client Module** With the previously presented application scenarios, the CBFPlayer was developed. The player corresponds to the Consumer Client Module as defined in the reference architecture in Chapter 4. Table 21 recaps requirements deduced in Chapter 3. Its major task is to receive the MPEG-TS broadcast, demultiplex, decompress,

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player</td>
<td></td>
</tr>
<tr>
<td>Receive*</td>
<td>Analyze Medium (C.1.2)                                                   ✓</td>
</tr>
<tr>
<td>Analyze*</td>
<td>Analyze Medium (C.1.2)                                                   ✓</td>
</tr>
<tr>
<td>Play*</td>
<td>Access Medium (C.1.1)                                                    ✓</td>
</tr>
<tr>
<td>Notification</td>
<td>Enabler (C.6.1.1), Disabler (C.6.1.2)                                    ✓</td>
</tr>
<tr>
<td>Sub-Menu for Collaborative Services</td>
<td></td>
</tr>
<tr>
<td>Analysis</td>
<td></td>
</tr>
<tr>
<td>Analyze Medium</td>
<td>Analyze Medium (C.1.2)                                             ✓</td>
</tr>
<tr>
<td>Analyze Content</td>
<td>Not supported with MPEG-2                                      ✗</td>
</tr>
<tr>
<td>Participation</td>
<td></td>
</tr>
<tr>
<td>Volunteer Participation</td>
<td>Collaborative Session (C.2.1.1)                                    ✓</td>
</tr>
<tr>
<td>Anonymous Participation</td>
<td>Vote (C.4.5.1)                                                          ✓</td>
</tr>
<tr>
<td>Private Groups</td>
<td>Registration (C.3.1.2), Create Group (C.4.1.1), Build group manually (C.4.1.3)/automatically (C.4.1.2), Invitation (C.4.1.5), Quick Join (C.4.1.6)/Leave (C.4.1.7), Buddy List (C.3.1.4), Embedded Chat (C.4.1.4)</td>
</tr>
<tr>
<td>Open Groups</td>
<td>Registration (C.3.1.2), Quick Join (C.4.1.6)/Leave (C.4.1.7), Buddy List (C.3.1.4), Embedded Chat (C.4.1.4)</td>
</tr>
<tr>
<td>Administration</td>
<td>Create Group (C.4.1.1), Invitation (C.4.1.5)                          ✓</td>
</tr>
<tr>
<td>Channel Hoppers</td>
<td></td>
</tr>
<tr>
<td>Collaboration</td>
<td></td>
</tr>
<tr>
<td>Keep Consistency</td>
<td>Replicated Object (C.2.2.3), Distributed Command (C.2.2.6), Conflict Detection (C.2.3.2)</td>
</tr>
<tr>
<td>Pool of Collaborative Services</td>
<td></td>
</tr>
<tr>
<td>Guidance</td>
<td>Updater (C.6.1.3), Coordinator (C.6.1.4)                               ✓</td>
</tr>
<tr>
<td>Utilization of Outcome</td>
<td>Patterns are listed in C.5.3 and C.5.2 but not implemented</td>
</tr>
</tbody>
</table>

Table 21: Requirements for the Consumer Client Module as defined in Chapter 3 (* Collaborative Broadcast Content Format)
analyze and display the audio and video streams. In the case of a data stream including metadata tags, the data frames are synchronized to the video or audio stream. When a collaborative service is triggered by a tag, a notification is shown to the user, who can participate voluntarily. The participation may happen single-handedly, in private or open groups, personalized or anonymous. Groups can be built automatically or manually by the collaborators. Connection and administration functionality is provided via a sub-menu. The consistency of a collaborative session is provided by the implementation of the “Replicated Objects” and the “Distributed Commands” patterns to follow the “WYSIWIS⁸” paradigm and to keep the sessions in a consistent state. The session management is realized by the implementation of the “Persistent Session” and the “State Transfer” patterns.

The content analysis could not be provided fully, because of the limited functionality of MPEG-2. Other missing points are the handling of channel hoppers, the inclusion and replacement of collaborators at runtime. In addition, the utilization (analysis and measurement) of the collaborative activity is missing.

**Broadcaster/Producer Module** In the previous Chapter 5 the Broadcaster/Producer Module was developed and used for the application scenarios to insert the metadata. With the developed application scenarios, suitable collaborative services are provided. A pool of predefined collaborative services was not developed. To link these collaborative services to the broadcast content format, tags were defined to anchor predefined metadata to trigger collaborative activity. Tags for these scenarios were solely timestamps. Table 22 recaps requirements deduced in Chapter 3.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collaboration</strong></td>
<td></td>
</tr>
<tr>
<td>Pool of Collaborative Services</td>
<td>×</td>
</tr>
<tr>
<td>Administration</td>
<td>×</td>
</tr>
<tr>
<td>Guidance</td>
<td>Updater (C.6.1.3), Coordinator (C.6.1.4)</td>
</tr>
<tr>
<td><strong>Map</strong></td>
<td></td>
</tr>
<tr>
<td>Map Medium to Content</td>
<td>×</td>
</tr>
<tr>
<td><strong>CMC Link</strong></td>
<td></td>
</tr>
<tr>
<td>Tags in Medium</td>
<td>Timestamp (C.1.5.2), Frame (C.1.5.1), Scene (C.1.5.3)</td>
</tr>
<tr>
<td>Tags in Collaboration</td>
<td>×</td>
</tr>
<tr>
<td>Integration</td>
<td>Access Medium (C.1.1), Analyze Medium (C.1.2), Enhance Medium (C.1.4), XML Trigger (C.6.1)</td>
</tr>
</tbody>
</table>

Table 22: Requirements for the Broadcaster/Producer Module as defined in Chapter 3

6.3.3 Reliability

This part of the evaluation refers to the "Error Protection" of the developed scenarios, presented in Chapter 4. The reliability of this system depends on the validity and integrity of the metadata packages from broadcaster to client side. In addition, an upstanding connection between the collaborators is important. First, the presented scenarios have to

⁸ "What You See Is What I See"
be classified for their level of criticality as defined in Chapter 4. The level of criticality of the voting scenario is classified as Level 3 criticality. For the gaming scenario it is Level 1 and 2a.

The voting scenario is less critical since it is delayed (no live content) and in an open group. The metadata for the voting are broadcast repeatedly since the voting phase extends over a period of time. For the voting scenario, no further actions were taken. The metadata packages are included redundantly, three times with the distance of one second. The packet is extracted by the CBFPlayer, which passes it along to the CAManager (cf. Figure 46 in Chapter 4), and validated and processed by the CAManager to the corresponding collaborative application. If an error is detected the packet is thrown out. If the collaborative application notices a redundant package (which is already in queue), it is thrown out too. If it does not exist, it is put in the queue of the collaborative application. One improvement would be to check for redundancy at the very beginning when the CBFPlayer extracts the MPEG-TS frames to the data picture queue. See Experiment #2 of the metadata extraction for results.

The gaming scenario has level 1 (participation in closed groups) and level 2a (participation as single user) criticality since the broadcast content is time critical live content and the collaborators should act in a group synchronized to this live content where the metadata act as coordinator for the closed groups. For the gaming scenario, only the redundancy is implemented as well. Metadata packets are included redundantly at a distance of one second, 4 times. Experiment #2 of the metadata extraction demonstrates that this overhead is no problem for the system. For future work, it is strongly recommended to implement further error protection mechanism for level 1 and 2a scenarios since the redundancy causes data overhead for the broadcaster. More efficient mechanisms like the forward error correction, status messages (between peers in the first instance and to the server in the second instance), peer reallocation in case of dead peers, group replacement in case of dead groups and a backup plan in case all groups are dead. In case of dead peers within a group, another peer of a list of online peers should be assigned to a group after an idle time of 10 seconds. One improvement for future work would be to assign so-called backup peers beforehand to the groups. These peers must stay online to get assigned in the event that one peer is dead.
CONCLUSION AND OUTLOOK

“Broadcasting is definitely in my cards for the future, and I’m determined to work hard at it - to perfect it and create my style and niche.”
— Apolo Ohno

According to the given quote, the concluding chapter summarizes the contribution, open issues/future work and prospects of the direct integration of collaboration into broadcast content formats.

7.1 JUSTIFICATION OF THE HYPOTHESES

In Chapter 1, four hypotheses have been posed. Their verification is discussed in the following, according to the approach of comparison of prediction with experience.

Hypothesis #1: “On technology level, it is feasible to link collaborative activity and broadcasting technology as a medium through advanced technical solutions based on existing broadcasting infrastructure”.

By designing the CMC Link by using XML metadata, which are included into the MPEG-TS stream, this hypothesis is verified. The behavior of the concept of merging metadata into the broadcast transport stream and extracting and processing them on the viewer’s side is pointed out by the scenarios presented in the previous chapter, which uses settled broadcast standards.

Hypothesis #2: “Additional solutions on content level, as well as on technology level are required to link collaborative activity and program content and formats which also includes a greater understanding of the content production workflow and narrative aspects”.

Several scenarios could be realized (theoretically and practically) by using the developed reference architecture (cf. Chapter 6). Two extrema were realized which demonstrate the lower and upper bounds of practicality, including the theoretical conception of several scenarios in between. The scenarios were realized by using on market technology which demonstrates the technical feasibility of the concept, even though some limitations have to be accepted, especially caused by the delay in broadcast. In addition, the development of a collaborative broadcast content format has pointed out the need of additional solutions on content level. Collaborative and social interaction must be planned in the content creation phase of the value chain. Nonetheless the verification of hypothesis #2 is verified.
Hypothesis #3: “Digital broadcasting requires the exploration of new emerging technologies that enable the consumer to collaborate, in a particular reference implementation thereof to make the consumer aware about the potential”.

Hypothesis #3 is tested by the examination of existing patterns for collaborative services as well as the presentation of concepts on how collaborative activity in the television area can be measured and analyzed to, in turn, modify the broadcast content format. The hypothesis was discounted by the use of mostly existing collaborative concepts. Existing collaborative concepts and patterns were feasible for this purpose; they solely had to be selected based on the real-time demand of the broadcast and, especially, of the television sector.

Hypothesis #4: “The test of a collaborative broadcast system requires the definition of a new set of benchmark and evaluation criteria”.

Finally, hypothesis #4 is verified by the development of evaluation criteria which allows the evaluation of the four main components “Broadcast Technology as Medium”, “Program Content & Formats”, “Collaborative Activity” and the “CMC Link”. The realization of the CMC Link by the inclusion of metadata required the definition of a new set of benchmark criteria to evaluate the inclusion and extraction of the metadata as shown in the previous chapter.

7.2 DISCUSSION

This thesis explored the idea of integrating collaboration into broadcast content formats. The journey to reach this goal is outlined in Figure 72 and presented in detail in Figure 73.

In a first step, state of the art work in the topics of interactive, social and collaborative television were examined, classified and evaluated relating to a pre-defined catalogue of criteria (c.f. (1) in Fig. 72). The state of the art work was classified and evaluated relating to the following criteria: “The Narrative Space”, “Features of the Collaboration” like scope, reach, representation, interaction modality and synchronization, the level of “Link of Collaborative Activity to the Broadcast Medium and Program Content & Formats” and finally “Hardware Features”.

The defined criteria reflect the four main building blocks of this thesis, namely medium/content (referring to the broadcast content format), collaborative activity and CMC Link (c.f.(1) in Fig. 71 which was done in Year 1 by the steps abbreviated in Figure 73). This comprehensive evaluation and classification resulted in the findings that the popularity of linking broadcast content to collaborative activity rises but the technical and the substantial realization is insufficient. In most cases, the collaborative activity occurs
unsynchronized on separated platforms, for example on web platforms in the Internet. In cases of synchronized collaboration, standards are missing. Proprietary and individual solutions are prevalent. In addition, collaborative and social features are frequently offered to passive linear broadcast content.

Based on this, a detailed requirement analysis was performed resulting in use case scenarios, deduced functional and non-functional requirements (c.f. (2) in Fig. 72 which was done in Year 1 by the steps abbreviated in Figure 73).

One in-depth requirement was the development of a reference architecture to manage the direct integration of collaboration into broadcast content formats. The reference architecture, essentially, defines the use of metadata to realize the link of collaboration and the broadcast content format (c.f. (3) in Fig. 72 which was done in Year 2 by the steps abbreviated in Figure 73) as well as design patterns (c.f. (4) in Fig. 72) to describe design practices of the four main building blocks, being the broadcast technology as medium, program content & formats, collaborative activity and the CMC Link.
For a proof of concept, the system architecture containing the three main modules *Broadcaster/Producer Module* (enables the preparation and inclusion of metadata into the broadcast content format by using the “CBF Metadata Editor”), *Broadcaster Server Module* (supports back channel scenarios by process further collaborative activity) and *Consumer Client Module* (displaying and processing the broadcast content format and its containing metadata) have been prototypically implemented (c.f. (5) in Fig. 72 which was done in Year 3 by the steps abbreviated in Figure 73).

Finally, the approach has been evaluated fourfold (c.f. (7) in Fig. 72). First, two application scenarios (including the “CBF Voting App” and the “CBF Gaming App”) presented in Chapter 3 have been implemented by using techniques and the three main modules defined and implemented with the reference architecture (c.f. (5) and (6) in Fig. 72 which was done in Year 4 by the steps abbreviated in Figure 73). This step also includes the prototypical concept of the collaborative broadcast content format “Against the Others - Interactive”, inviting the audience to become an active and collaborative part. Second, a dedicated benchmark to estimate the applicability of the CMC Link was done (in Year 5 by the steps abbreviated in Figure 73). The benchmark was configured to evaluate main broadcast criteria, like the performance, system requirements and reliability. With
regards to time factor and reliability, the benchmark reveals the suitability of metadata as a link for the time critical broadcast sector. Third, a feature-based evaluation was done by comparing the proposed concepts to competing approaches on the basis of the framework discussed in Chapter 2. Fourth, the hypotheses presented in the Introduction were verified to the approach of comparison of prediction with experience. Steps three and four of the evaluation were also done in Year 5 by the steps abbreviated in Figure 73.

In conclusion, television was seen as a passive and single-user medium for a long time. With technological progress, interactive television appeared and experienced several awakenings from the rising start to the prognosticated death during the past decades. But, when the progress of the past 10 years is considered, the approximation of television and social media established a shift from watching in private to exchanging views about the broadcast content.

Focusing on the trends in science, it slightly changes from pure platform development to the need of social and collaborative content formats as it was done with this thesis, which is seen as partial success. The use of XML metadata as link is technically feasible but, in the meantime, better solutions, like the HbbTV standard, exist to integrate collaboration into broadcast content format.

7.3 Key-results and Contributions

The main result of this thesis is in the creation of the missing CMC Link of broadcast technology, program contents & formats and collaborative activity. This resulted in the following major outcomes:

- **Survey, categorization, and classification of the state of the art:** A survey, categorization and classification of state of the art literature in interactive, social and collaborative television was performed. The analysis was done concerning to the narrative perspective on broadcast content, the link of collaborative activity and broadcast content formats, the collaboration and hardware features. This analysis resulted firstly in a missing technological and substantial link of broadcast technology, program content & format and the collaborative activity and secondly in the abundance of different solutions and platforms. This causes i.a. a media disruption for the viewer/consumer, missing synchronization of the collaborative activity and the watched program content and an abundance of proprietary solutions and platforms for the consumers.

- **Development of scenarios and requirements:** Scenarios and requirements were defined with respect to the Austrian television market and landscape.

- **Complex collaborative and social interaction framework:** Through an examination, survey, and classification of possible narrative structures enabling collaboration, a reference architecture for the direct integration of collaboration into broadcast content formats was designed. The reference architecture defines i.a. the system architecture of the three main systems parts (consumer, broadcaster and producer module) and design patterns for collaborative broadcast systems.
• **A concept for the missing “CMC Link”:** The missing link of broadcast technology (MPEG-2), broadcast content and collaborative activity is technological feasible. The CMC Link was designed by the inclusion of XML metadata into the broadcast technology (MPEG-2). To enable the inclusion, the “CBF Metadata Editor” was implemented within the frame of the broadcaster/producer module. Further experiments have shown that XML metadata are feasible to be included to broadcast technology and to link broadcast technology as medium, broadcast content & formats and collaborative activity. The procession and synchronization of additional metadata to the Audio/Video data at the consumer side (consumer module) can be done in real-time without noticeable delays and computer performance. CPU and memory usage minor increases and the metadata extraction is done within deci-percent of milliseconds. To enable the reception and procession of the included metadata, the “CBF Player” was implemented within the frame of the consumer module.

• **Definition and design of a Collaborative Broadcast Content Format (CBF):** The Collaborative Broadcast Content Format (CBF) “Against the Others - Interactive” was prototypically designed to enable this type of collaboration.

• **Metadata definitions:** Metadata were defined to encode meta-information about the content structure, it's collaboration capabilities, and exchange format between consumer, broadcasters and other 3rd parties.

• **Generic reference architecture to integrate this CBF into a TV stream:** Part of the reference architecture is the definition of design patterns to firstly access, modify and enhance broadcast technology (MPEG-2). Secondly to describe, identify, observe and process collaborative activity. Existing collaborative and social services, concepts and paradigms are feasible for the broadcast sector. Existing design patterns for social and collaborative activity, that are suitable for the broadcast sector, were adapted and enhanced. Thirdly to describe the link of collaborative activity to broadcast content formats.

• **Application of the proposed reference architecture:** The proposed reference architecture was integrated into the development of applications and services, including an editing environment to create collaborative content.

• **Synchronization models:** Synchronization models were concept for the content presentation, interaction, collaboration, secondary and tertiary screen support, and streaming technologies.

• **Development of proof-of-concept implementation:** The concept was proven by the implementation of two scenarios. Firstly a simple voting scenario (“CBF Voting App”) as a first experiment for the reference architecture. Secondly a more complex quiz scenario (“CBF Gaming App”) which provides several possible constellations of rivals and collaborations. The second scenario also included the prototypical design of the collaborative broadcast content format “Against the Others - Interactive”. As mentioned, the inclusion, synchronization, reception and procession of the metadata can be done in real-time without noticeable delays and computer performance. A critical point is the accelerated effort in the planning and creation of the metadata during the content creation phase of the television content value chain.
7.4 shortcomings and prospects

The presented work may be continued in several ways, as detailed in the following subsections.

7.4.1 Including the HbbTV Standard

This research started in 2008, at that time the field of collaboration in broadcast content formats or collaborative television was quite unknown and interactive television experienced a revival after its multiply prognosticated death. Turning the passive medium “television” into an active one was popular in science but unattractive for the broadcaster and the viewer because of insufficient devices and user interfaces. This changed dramatically in recent years because of the appearance of mobile devices (like tablets and smartphones), the interconnection and Internet connection of most devices in households (also the television device) and the popularity of apps (and app stores) and social networks. The obstacles at this time in 2010/2011 (when the implementation was done) were missing standards to make television interactive, which is a precondition for collaboration. The MPEG-4 reference architecture and the DVB-MHP platform were down. Therefore the researcher was constrained to define a reference architecture which took much time and effort. In 2011, the HbbTV standard was published right after the implementation was finished, whereas the researcher decided to keep it. Therefore, the most significant missing part in the technical realization of this thesis is the inclusion of the HbbTV standard.

To the best of my knowledge, at this point in time (May 2015), the HbbTV standard does not include any mechanism for collaboration and its quality of service. Future work in this area should be the extension of the HbbTV standard for the direct integration of collaborative and social interaction and the buildup of scenarios using this standard, because HbbTV [31] defines a basic set to build up interactive and social television apps and applications, also for second screen scenarios (since May 2014), which is the best base to extend and build up collaborative scenarios.

7.4.2 Experimental Setup

The experimental setup, presented in the previous chapter, is based on a Linux computer. The next consequential step is transferring the implementation to an up-to-date set-top box and the installation of an experimental setup for comprehensive user studies. This includes at least four or more setups respectively equipped with a television set, set-top box and a second screen device to realize user studies with up-to-date technology. Four setups are necessary to test the collaborative setup to allow group building or battling against each other. The defined benchmarks should be repeated within this user setup to get more meaningful results. In addition, the reliability and the error consistency can be tested by running the setup for a test period of at least 72 hours with three fictional peers. The test period should
include different active fronts, different times of day and other culprits to do funded quality of service measurements of the broadcast and the network. In addition, user-centered evaluations focusing on the real-time demands need to be done.

7.4.3 Define Further Metadata Patterns

In the course of this thesis, a first set of patterns has been proposed. Nevertheless, it is a basic set which is open to be extended to further modules and patterns. Further metadata patterns are, for instance, a “modifier” pattern to modify the course and characteristics of the broadcast medium (like switching the video stream) or more sophisticated coordinator and updater patterns.

7.4.4 Implement a Second Screen Scenario

Using a mobile device, like a tablet or a smartphone, as a second screen is planned in the reference architecture but not implemented yet. Realizing a scenario, which displays the video on the television device and swaps the CBF platform, for user inputs and collaborative activity, on a second screen device, is a future prospect of the presented work. Since most TV devices are already connected to a home network, the second screen device could receive the metadata from the TV device and, further, both devices can communicate and exchange triggers.

7.4.5 Identify, Observe and Measure the Collaborative Activity

Patterns to identify, observe and process the collaborative outcome are defined, but not considered in the prototypical implementation which, clearly, is a future prospect of the presented work. The evaluation of the collaborative outcome is an important part to realize back channel scenarios and to realize tags and trigger in the collaboration, which is an open issue as well.

7.4.6 Implement the Back Channel to Modify the Course and Characteristics of the Broadcast Content Format

The Broadcaster Server Module defines a back channel to the broadcaster, but the scenario is not considered in the prototypical implementation. One example would be the inclusion of several video streams or several different versions of story parts into the MPEG-TS container. Depending on the collaborative outcome, the collaborators or groups could see different parts of the story.

7.5 FUTURE WORK

In the field of interactive, social and collaborative television, the problem with the missing link of interactivity/social activity/collaborative activity and broadcast content formats is not solved sufficiently until now (September 2015). New research work (i.e. Shin et al.
workshop results (like the “Media Synchronization Workshop\textsuperscript{1}”) and startups (i.e. “TV Plus\textsuperscript{2}” and “Snapscreen”), prove the rising attention to this topic.

In addition to the open issues listed above, the future work concentrates in the cooperation with the Vienna company “Snapscreen”. Their second screen app “Snapscreen” allows taking and analyzing screenshots of currently consumed television program content. In case “Snapscreen” detects the program content, it offers available information, interactive/social services, or similar. The convergence to the Web and the missing link between interactivity/social/collaborative activity and the broadcast content format is realized by taking and analyzing the screenshot.

\textsuperscript{1} https://sites.google.com/site/mediasynchronization/mediasync2015
\textsuperscript{2} http://www.tvplus.com/
Part III

Appendix
Collaboration / Collaborative Activity / Collaborative Actions:
Collaboration (also termed as Collaborative Activity or Collaborative Actions) is defined as mutual engagement of participants in a coordinated effort to solve some problem together [25]. From a computer science point of view, a collaborative system is software that allows people to work together on files and projects via local and remote networks [17]. Hence, the term coordination can be defined generally as “managing dependencies between activities” [68] by using a coordination media (or coordinator) to manage these effort.

Broadcast:
From a technical perspective, broadcast is sending video and audio signals by attaching them to a carrier wave of electromagnetic energy that radiates in all directions [65]. From a communications perspective, broadcast can be termed as mode of communication which serves many receivers concurrently, typically from one transmitter. Typical broadcast channels are radio and television (TV). Additional application areas apply data transmission in mobile communication and local networks; for example the transmission of data packages to all nodes within a network [8].

Broadcast Content:
Broadcast Content is used as a generic term and defines an abstract model of the substance consumed by the audience [48].

Broadcast / Television Program Content / Program Content:
Broadcast Program Content (also termed as Television Program Content or Program Content) is a segment of broadcast content intended for broadcast on television, e.g. a specific television show or a movie1.

Broadcast / Television Program Format / Program Format:
A Television Program Format (also termed as Program Format, TV Program Format, TV Content Format or TV Format) is a master plan and branding of a copyrighted television program2.

Broadcast Technology as Medium / Broadcast Medium / Medium:
Broadcast Technology as Medium (also termed as Broadcast Medium or Medium) transmits the content to the viewer. Medium denotes the technical realization and representation of the content. Well-known data formats are, for example, MPEG-2 and MPEG-4, as used in the Digital Video Broadcasting (DVB) standard in Europe.

1 https://en.wikipedia.org/wiki/Television_program
2 https://en.wikipedia.org/wiki/Television_program
**Broadcast Content Format:**
The definition of the term Broadcast Content Format (also termed as Program Content Format or Content Format) poses a challenge and depends on the field of study. For example, Boiko defines content formats in general as “encoded formats for converting a specific type of data to displayable information” [15]. In the context of this thesis, content format (also termed as program content format) is used in the field of broadcast and is split into the parts of broadcast technology as medium and program contents and formats.

**Collaborative Broadcast Content Format (CBF):**
With the scope of this thesis, we define Collaborative Broadcast Content Format by enabling the collaboration between consumers to reach a common goal, by using the broadcast content format as a coordinating unit on a more intermediate modality.

**Collaborative Program Content / Collaborative Program Format:**
More specific subsets of collaborative broadcast content formats are, for example, Collaborative Program Content (also termed as Collaborative TV Content Format and Collaborative Program Formats (also termed as Collaborative TV Program Format or Collaborative TV Format) which enable the collaboration between consumers to reach a common goal by using the program content and / or format as coordinating unit on a more intermediate modality.

**Participatory:**
In the context of broadcast and television, the term Participatory defines that the audience can play an active role [18].

**Participatory Program Format:**
In Participatory Content Formats, the audience can play an active role in the process of manipulating, collecting, reporting and analyzing the program content [18].

**Narrative / Narration:**
A Narrative (also termed as Narration) is commonly referred to as a story. A “story” is a term that is understood intuitively but hard to define. For our purpose, a narrative is defined as a series of events that are linked together in a number of ways, including cause and effect, time, and place. Something that happens in the first scene causes the action in the second scene, and so on (while also moving forward in time) [35].

**Narrative / Narration / Content Structure / Space:**
The Narrative Space (also termed as Narration / Content Space / Structure) is defined as a place for immersing the viewer into the narrative flow, not only explicitly as a kind of co-director, but also by building virtual communities, communicating with other interested viewers and obtaining natural or narrated multimedia assets [54], [73], [35]. Furthermore, it is an indicator whether a given broadcast content is applicable for content-related interaction and collaboration.

**Narrative Modification / Alternation Collaborative Activity:**
The term Narrative Modification (also termed as Alternation Collaborative Activity) defines
collaborative activity which triggers modification to the narration flow.

**Non-Narrative Modification / Alternation Collaborative Activity:**
The term *Non-Narrative Modification (also termed as Alternation Collaborative Activity)* defines collaborative activity which does not trigger any modification to the narration flow.

**CMC Link:**
The term *CMC Link* defines the *Collaboration Medium Content Link* which is the link between broadcast content format (and its subsets) to collaborative activity. The link can be realized first on a technological level by for instance software trigger (events) which cause collaborative activity; And second on a thematic level where collaborative activity is caused for instance by announcements of the TV host. In the continuous text, the CMC Link is also termed as link.
In the context of Web and Internet, link(s) are termed as hyperlinks or weblinks.

**Hyperlink / Weblink:**
A *Hyperlink (also termed as Weblink)* is a reference to another passage in a document or another document. More exactly, it is a reference from a hypertext document to data at another location that can be followed directly by clicking or hovering [76].

**Broadcaster:**
The *Broadcaster* transmits the broadcast content format to the audience for public or general use.

**Producer:**
The term *Producer* refers to the person in charge of a specific program content.

**Target Audience:**
The *Target Audience (or Audience / Consumer / Viewer)* is defined as the persons reached by the television broadcast.

**Tag:**
*Tag* is a term to represent a certain timestamp, scene, period of time, or any other defined reference of the broadcast technology as medium or the program content. Its purpose is to anchor trigger for collaborative activity.

**Trigger:**
A *Trigger* is usually anchored to one or several tags and prompts (enable, disable, update) collaborative activity with different parameter and under different conditions. In this thesis, trigger are realized in form of XML metadata.
DESCRIPTION OF APPROACHES ENABLING COLLABORATION IN BROADCAST CONTENT FORMATS

The appendix presents, the classified approaches of collaboration in broadcast content formats. As previously mentioned, they are pre-categorized concerning to the link of collaboration and the broadcast content format. Each approach can be summarized by the categories Content, Collaboration, CMC Link (if relevant), and Hardware Feature (if available).

Note: This list of approaches is not exhaustive. The most cited and successful approaches were chosen to give a representative overview of the state of the art. Regarding social television, only a few approaches were considered and only for the sake of completeness, as social television is not the main topic of this thesis.

B.1 CMC LINK OF COLLABORATIVE ACTIVITY TO THE BROADCAST TECHNOLOGY AS MEDIUM

#1 - CiTV
CiTV focuses on the collaborative delivery of on-demand IPTV services [59].

<table>
<thead>
<tr>
<th>Content:</th>
<th>The program content and its narrative structure are not relevant for the social and collaborative features.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration:</td>
<td>The system focuses on the collaborative delivery of IPTV content.</td>
</tr>
<tr>
<td>Hardware:</td>
<td>Grouped IPTV service architecture.</td>
</tr>
</tbody>
</table>

Table 23: Synopsis for CiTV

#2 - LepidaTV / Multipler
The Italian regional digital terrestrial television channel (LepidaTV) and the centralized digital content management system Multipler (which allows collection, conversion, storage, visualization, and dissemination of digital content) were combined with a collaborative IPTV system. The new system allows for collaborative IPTV schedules, which are characterized by sharing uploaded content and live streams between registered users [12].

<table>
<thead>
<tr>
<th>Content:</th>
<th>The program content and its narrative structure are not relevant for the social and collaborative features.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration:</td>
<td>The scope of collaboration is to share content between registered users, and to provide shared schedules and social experience.</td>
</tr>
<tr>
<td>Hardware:</td>
<td>Personal computer or laptop with IPTV.</td>
</tr>
</tbody>
</table>

Table 24: Synopsis for LepidaTV / Multipler
#3 - Group TV

GroupTV is a cloud-based social television system that provides mechanisms for social experience (like messaging and social awareness while watching TV), group recommendations, and the collaborative delivery of distributed video [82].

| Content: | The program content and its narrative structure are not relevant for the social and collaborative features. |
| Collaboration: | The system provides social TV experiences, group recommendations and collaborative delivery. |
| Hardware: | Personal computer or mobile devices (e.g. tablet), and dynamic adaptive streaming over HTTP. |

Table 25: Synopsis for GroupTV

#4 - CollaboraTV

CollaboraTV, an AT&T research project, joins a special interface to existing TV content, which provides synchronous and asynchronous text communication between people watching the same channel. The viewers use avatars to express their opinions, rate material and recommend them to others. Each social and collaborative action is simultaneously annotated to the broadcast content format [63].

| Content: | The program content and its narrative structure are not relevant for the social and collaborative features. |
| Collaboration: | The scope of collaboration are synchronous and asynchronous social experiences like messaging, ratings and recommendations. |
| CMC Link: | Link to the medium is a given, because comments are attached at the corresponding temporal index. |
| Hardware: | A client server architecture. |

Table 26: Synopsis for CollaboraTV

#5 - Zync

Zync is a plugin for Yahoo! Messenger\(^1\) that provides social and collaborative experiences for online videos. Participants can share content and its controls with their Yahoo! Messenger buddy list [51].

| Content: | The program content and its narrative structure are not relevant for the social and collaborative features. |
| Collaboration: | The scope of collaboration is modified synchronized by the broadcast technology as medium by sharing the controls with friends. Friends are represented by a closed buddy list and interaction is limited to text based communication. |
| Hardware: | Zync is a prototypical plugin for the Yahoo! Messenger. It focuses more on video streaming, and less on broadcasting. |

Table 27: Synopsis for Zync

\(^1\) https://messenger.yahoo.com/plugins/view/7551/
#6 - CwaCTool
The CwaCTool supports the collaborative annotation of video content. While a selected video frame is being annotated, people can simultaneously view each others’ ink annotations, and discuss them using the chat tool. The resulting annotated video files can be shared through the underlying peer-to-peer infrastructure, allowing other users to complement annotations already made or to watch the videos created on an interactive TV platform. The users can annotate audio, text, and digital ink as well as to view all annotations made by other users. Annotations are shown as icons in the video and can be asynchronous or synchronous; in the latter case the icons are shown as soon as other users finish their own annotations [62]. A similar approach is presented in [40].

| Content: | Annotations are independent of content but they modify the broadcast medium. Annotations may turn non-interactive linear content into interactive linear content. |
| Collaboration: | The primary scope of collaboration is to create new content by adding artifacts to existing content. It also supports social interaction, like text and audio communication. In both cases, the interacting group may be open or closed. |
| Hardware: | Delivery after annotations happen via decentralized peer-to-peer architecture. Both, streaming and broadcasting of the content is possible. |

Table 28: Synopsis for CwaCTool

#7 - MIST
The Mobile and Interactive Social Television (MIST) service provides viewers with television content they are interested in via mobile devices in a virtual shared space [56, 57]. This allows for shared remote controls, awareness of other viewers, and the exchange of experiences via text, audio, and video conferencing.

| Content: | The program content and its narrative structure are not relevant for the social and collaborative features. |
| Collaboration: | The system provides a social TV experience by using shared remote controls, having awareness of others and communicating via text, audio, and video. |
| Hardware: | Client server system for mobile devices. Multimedia data are transported via UDP. |

Table 29: Synopsis for MIST

#8 - ResNet.TV
ResNet.TV provides fine-grained user feedback and channel recommendations to friends via a social network (e.g. facebook) and social awareness while watching through a highly instrumented user interface [45, 61].
The program content and its narrative structure are not relevant for the social and collaborative features.

The system provides a social TV experience (facebook comments, etc.) for closed groups and group recommendations.

IPTV service for personal computer.

### B.2 CMC Link of Collaborative Activity to the Program Contents & Formats

#### #9 - The Truth about Marika

Sweden’s SVT (Swedish television channel) participation drama *The Truth about Marika* is about a young woman who invites the viewers to help her search for her lost friend. The search took place all over Sweden and happened online. The hub of the search was the website [http://www.conspirare.se](http://www.conspirare.se), but several systems like TV, smart codes, Java programs, GPS, chat rooms, forums, and so on were integrated to aid the online search for Marika. Television, Internet, and reality were combined in an interactive narration.

| Content: | Conditional branched narrative structure (with optional scenes) by the connection of several worlds (real world, Internet) through a TV story. |
| Collaboration: | The scope of collaboration is to modify the broadcast medium by executing smart codes, Java programs and other provided gadgets. Depending on the outcome of this interactivity, the storyline is modified. |
| Hardware: | The main storyline was broadcast on Swedish television. The gadgets and the Internet platform were realized via a client-server architecture. |

#### #10 - Akvaario

*Akvaario*, broadcast by the Finnish TV channel YLE, is an interactive production of a large database of clips (approximately 5,000) automatically edited in real time, according to viewers’ interactions via telephone. By keying a number, viewers can modify the mood of the protagonists, who then “react” to the aggregated input (the appropriate clips are automatically showed). However, there is no narration, or spoken dialogue. The script does not focus on creating a story, but on generating a conversational space [72, 80, 81].

| Content: | Branched Content, composed from a large database of clips. |
| Collaboration: | Assemble new content from existing clips by collaborating via phone calls. |
| Hardware: | Conventional television environment and telephone. |

*Table 30: Synopsis for ResNet.TV*

*Table 31: Synopsis for the Swedish Drama “The Truth about Marika”*

*Table 32: Synopsis for the Finnish Production “Akvaario”*

---

2 [http://www.conspirare.se/](http://www.conspirare.se) - URL is offline.
#11 - Sofia’s Diary

Sofia’s Diary (by UK Fiver TV) is designed to deliver and act with the protagonist over the social network Bebo.com\(^3\). Sofia is a teenager, and constantly faced with dilemmas that viewers can comment on, blog about, and help to resolve. Their exchanges can be incorporated into the very short webisodes as they are produced. This show (season 3 was produced in 2009 and adapted for the US, Germany, Brazil, Chile, Turkey, and Vietnam) is addressed mainly to teenagers, and has a strong bias towards social exchange and development, rather than immersive dramatic narrations.

<table>
<thead>
<tr>
<th>Content:</th>
<th>Conditional branched and optional scenes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration:</td>
<td>Scope of collaboration is the modification and creation of new content.</td>
</tr>
<tr>
<td>Hardware:</td>
<td>The program content is delivered by an on-demand stream, with a conventional client-server architecture behind it.</td>
</tr>
</tbody>
</table>

Table 33: Synopsis for the British Drama “Sofia’s Diary”

#12 - SMS Games

In SMS games, a mobile phone is used as an input device for game moves done by sending an SMS, while the TV acts as a display for the game states [71]. SMS games usually use coordinates that must be chosen by the participants in order to fulfill a certain task, like, for example, throwing a snowball towards the host or kicking a football into a moving goal. Moves are sent by SMS, and TV sets serve as display devices for the game states. Essentially, iTV-SMS-games are pay-per-play games with SMS serving as both the method for player input and billing. Although the scope for these game mechanisms is limited, it is exciting for the viewer to become a participant in a live show, using only his/her mobile phone. The mobile-TV game Horse Derby goes one step further and turns the TV into a game console. Using the phone as a game pad, the participant can make his horse faster by pressing the “one” and “three” buttons constantly [71].

<table>
<thead>
<tr>
<th>Content:</th>
<th>Branched content.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration:</td>
<td>Scope of collaboration is to modify the content. The user interacts with the broadcast technology as medium per mobile phone and SMS.</td>
</tr>
<tr>
<td>Hardware:</td>
<td>Conventional television environment with mobile phone.</td>
</tr>
</tbody>
</table>

Table 34: Synopsis for SMS Games

#13 - TV Quizzes

Other interactive TV games are the well known quiz programs seen on “9Live” in Austria and Germany. These quizzes started soon after the first mobile TV games launched. Several kinds of quizzes exists, though most of them are based on luck for the viewer to get through to participate. The questions are usually very easy, like finding a female first name with exactly three “A” in it. In these games, one try/message/game move costs several Euro. The prizes can range from pride in your nickname to some amount of money [79].

\(^3\) http://www.bebo.com/sofiasdiary
#14 - LIVE
The LIVE project was done within an Austrian field trial during the Olympic Summer games in 2006. The idea was to build a system to help the director produce a digital bouquet of several output streams from a few hundred live streams and archive materials, covering a single event in real time. During the stream, the viewer could rate the content. Depending on these rates, the producer changed the broadcast content [83]. The goal of the LIVE feedback system is to enable the production team to observe the viewer behavior and preferences in real-time. The mood of the viewer is deduced from how they answer simple questions. The analysis results are provided as feedback for the director, showing how the viewer accepts the content. The director can react immediately, depending on the satisfaction of the viewer [83].

<table>
<thead>
<tr>
<th>Content:</th>
<th>Linear content of live sport genre.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration:</td>
<td>Scope of collaboration is to modify the content (e.g. by giving active and passive feedback).</td>
</tr>
<tr>
<td>Hardware:</td>
<td>Conventional television environment.</td>
</tr>
</tbody>
</table>

Table 36: Synopsis for the LIVE Project

#15 - The Social Quiz Game
The Social Quiz Game is a social TV content format, which integrates social network mechanisms and television in that way that social activity has impact on the course of the television quiz game. Players play in teams against each other via a social network and against a team of experts which entertains in the TV studio. During the air time of the show, the experts asks multiple choice questions which are answered by the players via the social network [11].

<table>
<thead>
<tr>
<th>Content:</th>
<th>Branched.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration:</td>
<td>Players on a social network play against a team of experts in the TV studio.</td>
</tr>
<tr>
<td>Hardware:</td>
<td>Conventional TV linked with social network app.</td>
</tr>
</tbody>
</table>

Table 37: Synopsis for the Social Quiz Game

#16 - BBCs Spooks Interactive
BBC’s Spooks Interactive[^4] allows people to interact with the story world through observations, tests, and puzzles. This content format features the use of a multistream interactive application on satellite and terrestrial television in the UK. The results of the observations, tests, and puzzles modify the way the story unfolds. The video streams switched to provide different outcomes depending on user inputs. The exploratorium is provided alongside the TV series.

## B.2 CMC Link of Collaborative Activity to the Program Contents & Formats

### Table 38: Synopsis for the British Drama “BBC’s Spooks Interactive”

| Content: | Linear Broadcast Content extended by an Exploratorium |
| Collaboration: | The scope of collaboration is to modify the broadcast technology as medium by solving tests and puzzles. |
| Hardware: | The program content is delivered by an on-demand stream with a client-server architecture. |

**#17 - ConnecTV**
This research project from TNO enhances TV-watching using social and collaborative applications, like buddy list, voice communication, content sharing, recommendations [14].

| Content: | The program content and its narrative structure are not relevant for the social and collaborative features. |
| Collaboration: | The scope of collaboration is a social experience, like watching TV content with invited friends, exchanging opinions, and sending recommendations via text and voice communication. Invited friends are represented in a buddy list. |
| CMC Link: | Correlation to the medium and the content is a given, as the broadcast content format can be shared with buddies and allows changes of views. |
| Hardware: | A personal computer equipped with a TV-card, remote control, and the connecTV software was used. In addition, the PC was connected to the TV screen. |

**Table 39: Synopsis for ConnecTV**

**#18 - 2BeOn**
The approach 2BeOn enhances the TV platform with social and collaborative functionality, like instant messenger, chat, voice and video communication, recommendations [7, 13].

| Content: | The program content and its narrative structure are not relevant for the social and collaborative features. |
| Collaboration: | The scope of collaboration is social interaction, like messaging, conversation, and voice and video communication. It also provides content recommendations to other users. |
| Hardware: | Conventional television environment. |

**Table 40: Synopsis for 2BeOn**

**#19 - AmigoTV**
AmigoTV is a research project of Alcatel-Lucent. It is a social and collaborative application for IPTV subscribers. While watching TV, consumers are aware of each other, and use voice communication and rich multimedia messaging possibilities (like emoticons) to share their opinions and avatars [27].
The program content and its narrative structure are not relevant for the social and collaborative features.

The scope of collaboration is a social experience, like communication and community support. Communication may happen via text messages or voice communication. Other users are represented in buddy lists and can be identified by avatars, which reflects the user's current emotions.

There is correlation to the content since viewers are able to discuss the current broadcast content, and the emotions of their avatars are adapted to the current content. For example, an avatar would show happiness when the user’s favorite football team scores a goal.

The prototypical implementation of AmigoTV, as presented in [27], uses a PC web interface and IPTV.

<table>
<thead>
<tr>
<th><strong>Table 41:</strong> Synopsis for AmigoTV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content:</strong></td>
</tr>
<tr>
<td><strong>Collaboration:</strong></td>
</tr>
<tr>
<td><strong>Hardware:</strong></td>
</tr>
</tbody>
</table>

---

#22 - “Schuld”
The German television company ZDF developed a second screen app for smartphones and tablets to correspond with the crime series “Schuld” (“guilt”) by Ferdinand von Schirach. The second screen app provided background information, forensic explanations, and the question of guilt alongside the current broadcasted episode of the crime series.

<table>
<thead>
<tr>
<th>Content:</th>
<th>The broadcast content structure is linear. Additionally, an exploratorium was offered online.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration:</td>
<td>Viewers can immerse themselves in the show via a second screen app that provides background information simultaneously with the broadcast. In addition, the viewer can discuss the show using social functionality and eventually answer the question of the protagonists’ guilt.</td>
</tr>
<tr>
<td>Hardware:</td>
<td>Conventional TV enhanced with a second screen app for smart phones and tablets.</td>
</tr>
</tbody>
</table>

Table 44: Synopsis for the German Crime Series “Schuld”

#23 - “Dina Foxx”
“Dina Foxx” is another crossmedia event produced by the German television company ZDF. “Dina Foxx” was a two-parter crime novel that was broadcast on German television. There were also online enhancements, like additional scenes and interactive games, offered on a web platform to solve open questions and puzzles. People had the choice to only watch the two-part episode, but if viewers wanted to solve the whole story, they had to go online.

<table>
<thead>
<tr>
<th>Content:</th>
<th>The broadcast content structure is linear. Additionally, an exploratorium was offered online.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration:</td>
<td>The additional content on the web is not really collaborative but interactive. Viewers could solve open questions posed by the broadcasted movie by watching additional scenes and playing interactive games on the Web.</td>
</tr>
<tr>
<td>Hardware:</td>
<td>Conventional TV enhanced with additional scenes and interactive games on the Web.</td>
</tr>
</tbody>
</table>

Table 45: Synopsis for the German Crime Series “Dina Foxx”

#24 - “De Ridder”
David Geerts et al. wrote in their paper [36] about a drama series, called “De Ridder”. The series was aired by the Belgian TV station “Eén” from October 2013 to January 2014, and is about a young prosecutor Helena De Ridder, who is fighting her own war for justice. It featured a newly developed, second screen, application, accessible by any device with Internet connection. It showed a timeline presenting additional content, which was synchronized and correlated to the broadcast content format. An animated slider visualized when the next update was scheduled. The correlated content could be quotes from the show, polls the users could respond to, additional information, maps of...

---

8 [http://www.zdf.de/dina-foxx/dina-foxx-35058724.html](http://www.zdf.de/dina-foxx/dina-foxx-35058724.html)
9 Information to the "De Ridder" are limited because of the lack of English-language sources.
10 Description taken from [http://www.eyeworks.tv/countries/belgium/local_productions/local_productions_item/t/de_ridder](http://www.eyeworks.tv/countries/belgium/local_productions/local_productions_item/t/de_ridder)
the location of characters, etc. [36]. The viewers could share and “like” the content via Facebook or Twitter.

<table>
<thead>
<tr>
<th>Content:</th>
<th>The broadcast content structure is linear. Additionally, an exploratorium was offered online which presented content synchronized and correlated to the broadcast content format.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration:</td>
<td>The additional content on the web was mostly additional substance to the broadcast content. Viewers could share and “like” the content via Facebook and Twitter. Polls were offered, but the course of the content was not modified.</td>
</tr>
<tr>
<td>Hardware:</td>
<td>Conventional TV enhanced with additional substance on the Web.</td>
</tr>
</tbody>
</table>

Table 46: Synopsis for the Belgian Drama “De Ridder”
DESIGN PATTERNS FOR THE INTEGRATION OF COLLABORATION INTO BROADCAST CONTENT FORMATS

The patterns in this Appendix result from the reference architecture defined in Chapter 4. Their purpose is to give basic building blocks to provide and manage collaborative activity and to integrate collaboration into a broadcast content format, thereby building a collaborative broadcast scenario.

The identified design patterns represent a base but raise no claim to completeness, and are open to be extended.

As defined in Chapter 4, the pattern description is based on the Alexandrian form designed by Christopher Alexander in his book “A Pattern Language (APL)” [10]. Patterns marked with one asterisk represent preliminary solutions, which are adapted for the broadcast application. Patterns marked with two asterisks are “nice-to-have” but not realized in the course of this thesis because they would be out of scope. Patterns which are not marked are redefined for the reference architecture. Patterns that are used for the prototypical implementation are listed in Chapter 5 and 6.

![Figure 74: Structure of the Pattern Language](image)

The patterns are structured as defined with the reference architecture in Chapter 4 and shown in Figure 74. Figure 74 reflects the basic building blocks of this thesis, namely the Broadcast Technology as Medium, Program Contents & Formats, Collaborative Activity and the CMC Link. Patterns defined for the Collaboration block are based on the work of
C.1 broadcast medium and program content support

Schümmer et al. as defined in the book “Patterns for Computer-Mediated Interaction” [74] and adapted for broadcast scenarios.

C.1 BROADCAST MEDIUM AND PROGRAM CONTENT SUPPORT

This section describes patterns that should be considered if the broadcast content format (medium and/or content) should be linked to collaborative activities. They mainly describe how to access the medium for embedding, extracting and processing the trigger. The main target groups for this layer are producer, broadcaster and the audience. The producer and broadcaster use the Broadcaster/Producer Module in the role of creating triggers and embedding them into the existing medium. The broadcaster, uses the Broadcaster Server Module, to include/exclude triggers during the runtime (which means during the broadcast), if possible. The audience, using the Consumer Client Module, is involved implicitly by patterns describing the analysis of the medium for embedded triggers and the parsing of them.

C.1.1 Access Medium

Context:
A participatory collaborative TV program is produced and triggers should be added to the broadcast content format in a next step.

Problem:
Adding the trigger to the broadcast content format requires, at least, quick and easy high-level access to single elements of the broadcast medium, which are in case of MPEG-2 single frames, sequences of frames (scenes), time code, video, audio and data stream.

Solution:
Provide an application which allows, also for non-professionals, quick and easy access to the available elements of a broadcast content format. The available streams must be presented to and be processable by the user. In addition, the application should provide access to the low-level elements of the single streams (like time codes, frames, etc).

Scenario:
The producer of a TV show needs to include prepared triggers into the broadcast medium, as they are marked up in the story board. To manage this, the producer opens the tool “CBF Metadata Editor” which simply allows the loading of an existing broadcast content format (e.g., MPEG-TS file). The video stream and its parameters (time, frames, etc) are presented, as well as the data stream (if present). If any triggers are already included in the medium, they are marked on the timeline and shown at the current tags where they are inserted (time stamps. scenes, etc). For instance, a trigger is inserted at frame 1374, the editor displays the included package with the corresponding frame.
Symptoms:
This pattern should be applied when
- Triggers should be included into the broadcast content format synchronized to the video or audio stream.
- Easy access to the available streams of a broadcast content format should be provided, also for non professionals.
- Overview of already included trigger is needed.

c.1.2 Analyze Medium

Context:
A participatory collaborative TV program is received and decoded by the consumer's broadcast receiver station. The broadcast medium needs to be analyzed for possibly included triggers.

Problem:
The MPEG-2 medium, which is a sequence of frames, should be accessible for collaborative services.

Solution:
Provide a player which analyzes the decoded medium. In case of MPEG-2, streams and frames of the medium can be analyzed.
The medium is, in the first instance, checked for included streams and whether a data stream and alternating video and audio streams are included. If a data stream is included, analyze the data frames for triggers. In a next step, included triggers need to be extracted, decoded, synchronized and passed through for further processing.

Scenario:
The consumer is watching television. The TV set receives and analyzes the broadcast content format for a data stream and included triggers.

Symptoms:
This pattern should be applied when
- The received broadcast content format should be analyzed for included triggers.

c.1.3 Modify Medium

Context:
The collaborative outcome of a live collaborative broadcast content format should modify characteristics of the future broadcast.

Problem:
General characteristics (like a second video stream, single frames or sequences of frames) should be modified for the future broadcast.

Solution:
Implement patterns of the categories “Collaboration Analytics”, “Access Medium” and “Analyze Medium” to evaluate collaborative activity and, in turn, include triggers, depending on the results of the collaborative activity. The triggers can be included in sections of the collaborative broadcast content format which aren’t
broadcast yet or which are already received but not yet processed by the collabora-
tor's receiver.

**Scenario:**
A collaborative live TV show is broadcast. The collaborative outcome is analyzed and returned to the broadcaster.

**Symptoms:**
This pattern should be applied when
- The future broadcast of a collaborative broadcast content format is modified by the collaborative outcome.
- Sections or parallel streams of the collaborative broadcast content format, which are already received but not yet processed, should be modified by the collaborative outcome.

### C.1.4 Enhance Medium

**Context:**
For the direct integration of collaboration into the broadcast content format, the medium has to be enhanced with triggers.

**Problem:**
The MPEG-2 medium, which is a sequence of frames, should be accessible for collaborative services.

**Solution:**
Insert previously defined triggers, anchored to tags of the medium into an existing data stream. If no data stream exists, use the “Access Medium” pattern first.

**Scenario:**
A collaborative broadcast content format is produced. For this purpose, triggers are created and time stamps are defined as tags to anchor the triggers to the medium. The triggers are inserted into the previously created data stream at the defined time stamps.

**Symptoms:**
This pattern should be applied when
- Collaborative broadcast content format is produced.
- The broadcast medium is enhanced with triggers.

### C.1.5 Tags to Anchor the Triggers to the Broadcast Medium and Content

#### C.1.5.1 Frames

**Context:**
Collaborative activity should be initiated by displaying a certain frame or sequence of frames.

**Problem:**
A technical anchor is needed to tie the trigger to the broadcast medium and content.

**Solution:**
Use a certain frame (audio or video) or a sequence of frames (audio or video) as tags to anchor the triggers.
Scenario:
Collaborative activity should be triggered by the beginning of a predefined sequence of frames displaying a commercial break. The producer includes the trigger, to enable a collaborative application, synchronized to the first frames of the sequence. A second trigger, to disable the collaborative application, is inserted with the last frames of the sequence.

Symptoms:
This pattern should be applied when
* The occurrence of a certain frame or a sequence of frames should trigger collaborative activity.

Context:
Collaborative activity should be initiated at a certain point in time.

Problem:
A technical anchor is needed to tie the trigger to the broadcast medium and content.

Solution:
Use a certain frame, that occurs at the desired point in time, as a tag to anchor the trigger.

Scenario:
Collaborative activity should be triggered at a certain point in time, in a collaborative broadcast content format. Therefore, the producer includes the trigger to enable a collaborative application, synchronized to the video frame that occurs at the desired point in time.

Symptoms:
This pattern should be applied when
* A certain point in time should trigger collaborative activity.

Scene

Context:
Collaborative activity should be initiated at a certain scene.

Problem:
A technical anchor is needed to tie the trigger to the broadcast medium and content.

Solution:
Scenes can be considered as a sequence of video and audio frames. In the case of MPEG-2, use an audio or video frame (or a sequence), that occurs at the desired scene, as a tag to anchor the trigger.

Scenario:
Collaborative activity should be triggered at a certain point at the beginning of a certain scene in a collaborative broadcast content format. When using the MPEG-2 standard, the producer adds the trigger, to enable a collaborative application, synchronized to the first frames that occur at the beginning of the desired scene.

Symptoms:
This pattern should be applied when
* A certain scene should trigger collaborative activity.
C.2 COLLABORATION - APPLICATION MANAGEMENT

This category defines patterns for the basic infrastructure of the system, excluding the network structure. The network structure of the system can be assumed, therefore it is not necessary to define patterns for this. These patterns are targeted to handle the session management, to manage shared objects and data consistency. These patterns are mainly targeted at developers who have to work out how shared objects should be managed and how information exchange is mediated by the computer system.

C.2.1 Session Management

C.2.1.1 Collaborative Session **

Context:
A group of users wants to participate in a collaborative broadcast content format in the late afternoon. How can they plan and coordinate their collaboration?

Problem:
Although the shared context is already given (which is the collaborative broadcast content format in the late afternoon), it is necessary to provide the participants with previous information. They must be able to prepare for the show. Whether this preparation is mandatory must be decided in advance.

Solution:
Provide previous information on a related platform. Model the preparation activity of the participants (groups) as a shared session object, which is stored locally (private) or on the broadcaster’s side (public). The session state must, at least, be visible to those who are involved. Users should be supported in starting, joining, leaving and terminating the session, which is possible at each point in time. When the session is started (users are joining), the necessary collaboration tools will start automatically.

Scenario:
David, Sara and Tim plan to join a participatory TV game show, which is broadcast in the late afternoon. The show requires previously built groups and selected collaborative services for the group. These can be chosen from a pool of services. Participation is, therefore prepared by building a group and selecting the collaborative services.

Symptoms:
This pattern should be applied when
- Users need a lot of time to do the necessary preparation which might be critical in broadcasting.
- The participation of a critical number of users is expected, which might be critical in broadcasting as well.
- Users should be animated and provided in participation.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].
c.2.1.2 Persistent Session *

Context:
A group of users just participated in a collaborative broadcast content format. Make the achieved results available for resuming or for further participation in future episodes of a certain TV show.

Problem:
After the end of the TV game show, the user wants to resume their collaborative results but they are not available, or the collaborative group wants to participate in a future episode of the TV game show together, but their session information is gone.

Solution:
Store the results of a synchronous collaborative session replicated at each collaborative actor or, in the event of following participation in future episodes, share the session on a central server belonging to the TV game show and allow users to access the results.

Scenario:
Steve, Anna and Mark participate in a TV game show as a randomly built group. They work well together and were the winners of this episode. As a result, they can battle it out in the final round against the winners of other episodes. Their results, group and session information were stored locally on a server on the broadcaster’s side.

Symptoms:
This pattern should be applied when

* The results, group and session information should stay available and accessible for some reasons.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.2.1.3 State Transfer *

Context:
Users are allowed to join, leave and rejoin a collaborative session at each point in time.

Problem:
The broadcast area is characterized by fluctuation in audience interaction (viewer turning TV on/off or switching the channel). Most collaborators will not participate from the beginning or until the end. This is a problem because, in the first case, lateral entrants do not know the intermediate results, state, rules and perhaps even the purpose/goal of the collaboration. In the second case, built and participating groups may crumble.

Solution:
Transmit the current state of the session and its shared objects to the newcomer from one current participant. Since all participants hold the current state of the shared objects, any client is able to perform the state transfer. State consistency is required.
Scenario:
Thomas is skipping through the channels and comes across a collaborative participatory TV show. He wants to join the collaboration but only 25 minutes of the show remain. By means of the BuddyList, Thomas decides and is permitted to join group #89. He can infer the current task from the status of the group. However, he doesn't know the progress in the collaboration (current state of the shared item).

Symptoms:
This pattern should be applied when
- Users should be able to join/leave/rejoin collaborative sessions.
- Users need to understand the current interaction focus and context.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

C.2.2 Shared Object Management

C.2.2.1 Centralized Objects *

Context:
Allow users to share objects and access these objects remotely.

Problem:
Participants need to share data to collaborate.

Solution:
Manage concerned data and instances centralized on a server and provide access for all collaborators.

Scenario:
In the participatory TV show ‘1 against the Others - interactive’ one participant in the TV studio competes against 100 collaborating participants in front of their TV sets. Participants at home share centralized objects for gaming, which are held on the broadcaster’s side.

Symptoms:
This pattern should be applied when
- Users have to participate in person on the TV show because they don’t know how to share and exchange data.
- Users cannot participate in the TV show at all because they don’t know how to share and exchange data.
- Users cannot establish a common ground to get an understanding of shared data.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

C.2.2.2 Consistency of Shared Objects (Remote Subscription) *

Context:
Shared objects, which are edited and may change over time, are provided to the collaborators. The state and the visual representation of the objects need to be consistent.
Problem:
Shared objects maintained by a server are assumed to be valid. Caused by the collaboration, objects and their representation may change. Since the object may be shown over a long period of time, the representation and state may be outdated. People using the object then receive incorrect information.

Solution:
Provide subscription for clients to the state of shared objects. Subscribers are informed about changes of shared objects.

Scenario:
Sara and her group are participants in the collaborative broadcast content format '1 against the Others - interactive'. They have to solve different tasks together, thus they share the definition of the task and the solution space as shared objects.

Symptoms:
This pattern should be applied when
- It is important that participants do not use outdated information because of the real-time quality of broadcasting.
- Polling for new states is not possible (due to resource restrictions).

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.2.2.3 Replicated Objects *

Context:
Improve the performance of shared objects which are of concern in the broadcast area.

Problem:
By using the centralized objects, the response time depends on the client-server communication (e.g., by network latency). Dependencies on the client-server communication may cause slow response times, which is a particular problem in broadcast technology because of the probable high participation rates and the real-time demand.

Solution:
Replicate shared data to client sites. Local replicas are accessed, modified and synchronized by the users.

Scenario:
Sara and her group are participants in the collaborative broadcast content format '1 against the Others - interactive'. They have to solve different tasks together, thus they share the definition of the task and the solution space as shared objects. More complex operations, like drag and drop, would be too slow with centralized objects, because of the high participation rate.

Symptoms:
This pattern should be applied when
- Permanent network connection between collaborators is not required (e.g., if collaborators do not know each other).
- Users perform many incremental changes on large objects.
- High participation rate causes (would cause) network delays.
- The response time is too slow.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.2.2.4 Update Dispatcher/Centralized Updates *

Context:
The use of replicated objects requires consistency, which requires synchronization of the replicated objects. It is important to keep the overload low.

Problem:
Keeping consistency by exchanging messages directly is complicated and error-prone since the clients must identify each other and establish communication. This is especially so in the television area, where high fluctuations are not predictable and require many updates.

Solution:
A central site maintains a list of all clients that are interested in changes to a replicated object. The clients report all changes to replicated objects to this central site, which in turn will distribute an update message to all other interested clients.

Scenario:
Sara and her group are participants in the collaborative broadcast content format ‘1 against the Others - interactive’. They have to solve different tasks together, thus they share the definition of the task and the solution space as shared objects. More complex operations, like drag and drop, would be too slow with centralized objects, because of the high participation rate.

Symptoms:
This pattern should be applied when
- Clients don’t know each other.
- Clients don’t want to exchange data directly.
- Clients change frequently.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.2.2.5 Decentralized Updates *

Context:
Synchronization of replicated objects is required.

Problem:
Collaborators want to use replicated objects, but centralized updates are not possible.

Solution:
Messages about changes on shared objects are propagated, directed at all clients who maintain a replica as well. Consider that all clients have received this message by, for example, acknowledgments.
Scenario:
Sara and her group are participants in the collaborative broadcast content format '1 against the Others - interactive'. They have to solve different tasks together, thus they share the definition of the task and the solution space as replicated objects. More complex operations, like drag and drop, would be too slow with centralized objects, because of the high participation rate.

Symptoms:
This pattern should be applied when
- Replicated artifacts are used for collaboration.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümm et al [74].

c.2.2.6 Distributed Command *

Context:
The use of replicated objects requires consistency, which demands synchronization of the replicated objects. It is important to keep overload low.

Problem:
Replicated objects are used for collaboration. When distributing the changes to other clients holding a replica, more information than necessary may be distributed. This increases the network load and response time, which is critical in real-time television scenarios.

Solution:
Distribute the commands that are executed on replicas. The commands are re-executed locally by the clients on their replicas.

Scenario:
Sara and her group are participants in the collaborative broadcast content format '1 against the Others - interactive'. They have to solve different tasks together, thus they share the definition of the task and the solution space as replicated objects. More complex operations, like drag and drop, would be too slow with centralized objects, because of the high participation rate.

Symptoms:
This pattern should be applied when
- The propagation of the new state of shared objects takes too long and produces too many overheads.
- The description of the change to the shared data is relatively small, compared to the size of the shared data itself.
- Shared objects are changed very often.
- High participation and high activity causes many changes and, therefore, much data exchange.
- Response time is too long and not applicable for real-time broadcast scenarios.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümm et al [74].
C.2.3 Data Consistency Management

C.2.3.1 Pessimistic Locking *

Context:
To avoid inconsistencies, when many users share objects, ensure that all changes are applied, even if many users try to apply many changes at the same time.

Problem:
If users are allowed to change shared objects simultaneously, inconsistencies can occur.

Solution:
Use distributed lock, which must be requested by the client who wants to change the state of a shared object. The lock can have different grain sizes, which means the whole object does not need to be locked each time a client wants to apply changes. After applying the changes, the lock is released by the client.

Scenario:
Sara and her group are participants in the collaborative broadcast content format '1 against the Others - interactive'. They have to solve different tasks together, thus they share the definition of the task and the solution space as shared objects. More complex operations, like drag and drop, would be too slow with centralized objects, because of the high participation rate.

Symptoms:
This pattern should be applied when
• Undoing changes cannot be performed (e.g., in real-time broadcasting scenarios, changes rely on external events).

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

C.2.3.2 Conflict Detection *

Context:
The use of replicated objects can cause conflicting changes.

Problem:
Changes may interfere when at least two users change the same data at the same time. If the conflict is not solved, the data may become inconsistent. If the users are unaware of this conflict, it can contradict their intentions.

Solution:
The client should be aware of all local changes that have not yet been applied by other clients. Whenever a change, affecting an object, is received from a client, and those changes have not been applied by the other clients, warn the client if the changes will produce a conflict. Perform arrangements that will enable all clients to have a consistent state (like undo or transformation of the change).

Scenario:
Sara and her group are participants in the collaborative broadcast content format '1 against the Others - interactive'. They have to solve different tasks together, thus they share the definition of the task and the solution space as replicated objects.
More complex operations, like drag and drop, would be too slow with centralized objects, because of the high participation rate.

**Symptoms:**
- This pattern should be applied when
  - Users are working with local replicas, which may lead to temporary inconsistent views.
  - Users are allowed to perform parallel changes, which may lead to a conflict.
  - The social protocol does not ensure consistency.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

### C.3 Collaboration - Community Support

This section contains patterns for establishing and managing a community, which is necessary since collaboration in a large organizational context is applied. The community is seen as optional superior structure, which can be split into groups, if necessary. These patterns should be considered if communities are provided. They mainly describe (social) processes to establish or close, enter or leave a community, support, security and administrative issues. The main target groups are the audience and the broadcaster, with the audience using the consumer client in the role of community member only, and the broadcaster using the broadcaster server module in the role of community founder and, therefore, also administrator. In many cases, these patterns can be implemented by tailoring existing community support technology (those patterns are marked with an asterisk).

#### c.3.1 Community Management Activities

##### c.3.1.1 Create Community

**Context:**
- A new community should be established.

**Problem:**
- People want to participate in a certain collaborative broadcast content format (CBF).

**Solution:**
- Provide a CBF community or a community to a certain collaborative broadcast content format, which people have to join if they want to participate collaboratively.
- The registration may be necessary to enter and build groups.

**Scenario:**
- David wants to participate in a collaborative TV show as part of a collaborative group. To be able to participate, David needs to register and enter the CBF community.

**Symptoms:**
- This pattern should be applied when
  - A certain community should be provided.
A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

### c.3.1.2 Registration *

**Context:**
A new community is established and open for new users.

**Problem:**
To build groups automatically, or to participate within an open group, people have to register for the community first. Since the broadcast sector is very public and time critical, people hesitate in giving too much private information to the public.

**Solution:**
Provide a simple registration procedure where people can register with as little information as possible. This reduces, firstly, the inhibition level of disclosing too much information to the television and, secondly, it reduces the time exposure of the registration process, which is important for as time critical an area as the broadcast sector is.

**Scenario:**
Barbara just noticed a community of a collaborative broadcast content format. She moves to the registration page where she has to fill in a short questionnaire relating to personal data (screen name, eMail address and password). After transmitting the questionnaire, Barbara receives a confirmation email, which includes her login data.

**Symptoms:**
- This pattern should be applied when
  - A new community was built and is open for new users.

### c.3.1.3 Login *

**Context:**
Anonymous collaboration should turn into personal collaboration.

**Problem:**
Personal collaboration (not anonymously) is not supposed in the collaborative space.

**Solution:**
Require username and password before granting access to the collaboration space.

**Scenario:**
Tom and Josh want to participate in a collaborative TV show as a private group. For this, they have to, firstly, create a user account and log in.

**Symptoms:**
- This pattern should be applied when
  - Private and public collaboration is provided.
  - Users are annoyed about not knowing with whom they are collaborating.
- The collaboration should be kept up for several episodes of the broadcast content format.
- Access to shared data should be restricted or granted.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

### C.3.1.4 Buddy List *

**Context:**
The collaboration space is shared with many other users.

**Problem:**
When many users participate in the collaboration space, the number of users could exceed the number of relevant contacts for a specific user. The user list grows very large, making it difficult to find relevant people.

**Solution:**
Provide Buddy Lists of users who are of interest to each user. The user is able to bookmark other users for the Buddy List.

**Scenario:**
For the game show ‘1 against the Others - interactive’, a buddy list was created for the 100 players to recognize each other and to know who is already retired.

**Symptoms:**
- This pattern should be applied when
  - Time for users to find others should kept short.
  - Users are interested only in a small set of other users.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

### C.3.1.5 Quick Goodbye *

**Context:**
Make it easy for people to leave the community.

**Problem:**
Complex procedures to leave a community will lead to inactive users in the community. In addition, they act as a deterrent to potential new members of the community.

**Solution:**
Users will leave the community at any time. It is necessary to make it easy for them to leave the community by providing access to the required procedures. Users should not have to know the details of how the community system handles the registration.

**Scenario:**
Since a certain collaborative broadcast content format changed the rules for participation, Barbara is not interested in participation any more. She wants to leave the community. Her registration was a couple of months ago, and she has forgotten the procedure to unsubscribe. Fortunately, a simple procedure for unsubscribing is provided.
Symptoms:
This pattern should be applied when
• Users have to contact other community members for asking how to leave the community.
• Many inactive user accounts exist in a community (no answers to requests, etc).

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.3.2 Security & Privacy

c.3.2.1 Reciprocity *

Context:
The collaborative output is processed by the user's input. To reach the best output, all group members should participate in the process.

Problem:
The goal should be rewarding for every member in the group, otherwise some active users might become frustrated and turn into passive followers.

Solution:
Provide reciprocity. All members’ activities must result in the group result. Only active members who are willing to help the group should benefit from the group result.

Scenario:
Sara, Tom, Richard and Andrea are participating as a private group in a collaborative game show. They compete against other groups and against a group in the TV studio. Sara, Tom and Andrea solve most of the given tasks. Richard is a member of the group but is not participating in the competition. The recognition and the benefit is shared between the leading collaborators in the group.

Symptoms:
This pattern should be applied when
• Users are not consistently involved in the collaboration, even though it is required by the collaboration process.
• Individual group members have a feeling that they do all the work and that others only act as free riders.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.3.2.2 Masquerade *

Context:
A collaboration environment is always monitored somehow, which means that personal information (like email address, work status, fragments of content) are visible to others.
Problem:
Monitoring users is somehow necessary to provide awareness information to others. This often causes users to feel insecure. They want to avoid providing any information to others, especially in as public a sector as the broadcast sector is.

Solution:
Users should decide what information they want published, and in which context. This requires methods for the users to filter information that is revealed from their personal information.

Scenario:
As a preparation of the participation to the TV show ‘1 against the Others - interactive’ the participants have to decide and authorize what information, in correlation to the show, is permitted to be public on the Web and TV.

Symptoms:
This pattern should be applied when
- Personal information might be abused by others, which probably happens in a TV scenario.
- Users do not trust each other.
- Users do not know each other.
- Users want to remain anonymous, either in the group or in the broadcasting context.
- Users want to provide personal information only to selected users.
- Users hesitate to participate because they fear that their contributions will be criticized.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümm er et al [74].

C.4 collaboration – group support

The group support section provides patterns for supporting groups in their establishment, enhancement, closing, communication and interaction. These patterns should be considered if groups are provided but not yet built. They mainly describe (social) processes to build, close, join or leave a group, support, security, communication and administrative issues. The main target group is the audience using the Consumer Client Module in the roles of group member or founder and, therefore, also administrator. Similar to the community patterns, in many cases, these patterns can be implemented by tailoring existing group support technology (those patterns are marked with an asterisk). The critical factor is the delay in broadcasting scenarios, which means joining/building/leading/interacting in the group is time-critical.
C.4.1 Group Management Activities

C.4.1.1 *Create Group*

**Context:**
A new group should be established.

**Problem:**
For a collaborative broadcast content format, the participants need to be arranged in groups.

**Solution:**
Provide establishing open and closed groups. Open groups are anonymous, informal groups, which everybody may join and leave. The group size is not mandatory given. Closed groups are personalized, formal groups. To become a member, an invitation or access granted by the group leader or other members is necessary. The group size may be limited or previously defined.

**Scenario:**
David wants to participate in a collaborative TV show. To enable this, he needs to join a group.

Case 1 - David joins an open group: David wants to stay anonymous in a large open group, but he joins the open group.

Case 2 - David joins a closed group: David wants to compete within a collaborative group against the candidates in the show. Therefore, he can build his own group, join a friend's group, or join a group of strangers.

**Symptoms:**
This pattern should be applied when
* Open and closed groups should be provided.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

C.4.1.2 *Build Group Automatically*

**Context:**
A participative TV show is applying for spontaneous collaboration.

**Problem:**
Building a group for collaboration might be a time-consuming process. In television scenarios, time-consuming administrative activities are critical because of the real-time demand.

**Solution:**
Provide voluntary automatic group building for users who don't know each other, for those who access the show later and in case of spontaneous collaboration. Ask users if they want to be available for spontaneous groups when the status turns online.

**Scenario:**
In a collaborative TV quiz show, the candidate Walter decides to use the audience joker, which allows him to ask the audience at home. Now spontaneous collaboration is required. Users who are online and agreed in spontaneous groups are
now assigned to groups to solve the task collaboratively. The group outcomes are processed in the show.

**Symptoms:**
This pattern should be applied when
- Time is too short for building groups manually.
- Users have no possibility to get to know each other for building groups.
- Late-comers should be allowed to participate in the show.

---

### c.4.1.3  *Build Group Manually*

**Context:**
A collaborative TV show requires group building in advance to participate.

**Problem:**
Users may not know and need to find each other or friends want to build a group for participating in the collaborative broadcast content format.

**Solution:**
Provide services and support to build groups easily and quickly. The process of building a group must linger in the user’s memory.

**Scenario:**
A collaborative TV show promotes participation. Mark, Anton and Tim are interested in participation and, therefore, build a group in advance, to participate later in the evening.

**Symptoms:**
This pattern should be applied when
- Groups are required (in advance) for collaboration.
- Collaboration in private groups is provided.

---

### c.4.1.4  *Bell*

**Context:**
A group was established manually or automatically. How should latecomers be dealt with?

**Problem:**
In television, users often switch channels several times during a show, or during the advertising break. Therefore, users may join in or abandon the collaborative participative show. When latecomers want to join a collaborating group, they may be disturbed by uninvited visitors or they simply may not notice that someone wants to join.

**Solution:**
Provide for each group to declare whether it is open or closed. If it is open, announce it with a bell that allows latecomers to draw attention to their desire to participate in the group.

**Scenario:**
Tina and Andrea build a group to act as a team in a collaborative TV show. Hannah wants to join them, since their group is declared as “open”. She asks to join them by using the bell.
Symptoms:
This pattern should be applied when
- It is not certain that all users will participate in the collaboration from the beginning.
- It is not certain that all users will participate in the collaboration until the end (group may get free space).
- Access to groups should not be taken for granted (private groups should be allowed).

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.4.1.5 Invitation *

Context:
A single user wants to collaborate within a TV show.

Problem:
A single user wants to collaborate with another one. The other user is unavailable or busy in another group.

Solution:
Provide sending and tracking invitations to collaboration, including the context (for which TV content) and other meta-information (e.g., in which group the inviting user is active, etc).

Scenario:
Hannah is already busy in a group that is solving a time-critical task in a TV show. Hannah recognizes Martha, who just joined the community. In agreement with the other group members, and with the rules of the TV show, Hannah is sending an invitation to Martha to join the group.

Symptoms:
This pattern should be applied when
- A user wants to collaborate with another user but doesn’t know how to tell them.
- Users don’t want to get disturbed during collaborative activity.
- You want to support the group building process.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.4.1.6 Quick Join

Context:
A user gets an invitation from a group or wants to join an open group in the context of a collaborative broadcast content format.

Problem:
A user wants to join a group quickly to participate the last 15 minutes of a collaborative content format. But joining a group is a long-term process.

Solution:
Provide techniques for users to join a selected group quickly.
Scenario:
Markus switched to a channel where a collaborative show has been running for 30 minutes. He wants to participate in the last 15 minutes by quickly joining Hannah’s group.

Symptoms:
This pattern should be applied when
- Users have inhibitions to join existing groups because the procedure is too complicated.
- Users should be able to join a group temporarily and quickly.

C.4.1.7 Quick Leave

Context:
A user gets an invitation from a group and therefore wants to leave the current one.

Problem:
A user switches channels, but the other group members do not notice the absence since the user didn’t log off from the group.

Solution:
Provide methods for users to leave a selected group quickly and, if necessary, automatically.

Scenario:
Markus switched to a channel where a collaborative broadcast content format has been running for 30 minutes. He wants to participate in the last 15 minutes by joining Hannah’s group quickly and leaving his current group automatically.

Symptoms:
This pattern should be applied when
- Collaboration comes to a standstill because users do not log off from their groups.
- Users can’t switch the group because they don’t know how to log off from the current group.

C.4.2 Synchronous Group & Inter-group Awareness

C.4.2.1 Group List *

Context:
Several groups are competing within a broadcast content format.

Problem:
For a good competition, the groups must be aware of each other, including some sort of ranking of the results.

Solution:
Provide a group list of all participating groups, showing intermediate and final results.

Scenario:
Groups are competing within a collaborative TV show. After solving a task, a ranking is published.

Symptoms:
This pattern should be applied when
• Several groups need to be aware of each other.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.4.2.2  *Activity Indicator*

**Context:**
Users may not know each other and work geographically distributed in highly synchronous sessions, requiring frequent request-response interaction, which TV scenarios may require.

**Problem:**
In collaboration with distributed users, they are not aware of other users’ activities. This may result in conflicting work or unnecessary delays.

**Solution:**
Activities of other group members are displayed in the user interface using a symbolic representation, for instance a growing balloon, traffic light colors or similar.

**Scenario:**
A group is collaborating in the TV show '1 against the Others - interactive'. The best performing participant gets a prize. For this, the performance level of the collaborators is displayed on a screen in the TV studio showing the ranking.

**Symptoms:**
This pattern should be applied when

• Users don’t like distributed collaboration because they don’t know what other users are doing.
• Users interact concurrently.
• Users don’t know if others are interacting or not.
• Users work at cross purposes.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.4.2.3  *Active Neighbors*

**Context:**
Build groups automatically.

**Problem:**
For high spatial distances, delays may disturb the group work.

**Solution:**
Build groups of spatially close collaborators. The assumption is that people want to share their locations.

**Scenario:**
Groups from different communities compete against each other. Therefore, spatially close collaborators of single communities form a group.

**Symptoms:**
This pattern should be applied when
• Delay within the group work should be minimized.
• Spatial distance or neighborhood is the matter.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.4.3 Asynchronous Group & Inter-group Awareness

c.4.3.1 Aliveness Indicator *

Context:
Users collaborate asynchronously on shared objects, over a long period.

Problem:
Asynchronously working users experience only a subset of collaborative activities, compared with others. Mostly, it is not obvious whether other groups or community members were active or not during their absence.

Solution:
Provide an indicator showing the aliveness and fluctuation in the group or community.

Scenario:
An exploratorium to a crime story is provided on the Internet. Until the next episode, collaborators of an open group should solve different common puzzles and tasks asynchronously. Collaborators get indicators if group members are still alive and about the progress of the collaboration.

Symptoms:
This pattern should be applied when
• Users are confused about whether other members are active or not.
• Users ask the group who is still active or not.
• Users do not know if others will follow their requests or not.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.4.4 Communication Activities

c.4.4.1 Embedded Chat *

Context:
Users are collaborating within the context of a collaborative broadcast content format.

Problem:
For successful synchronous collaboration, users need to communicate in real-time.

Solution:
Provide a communication tool to the collaborators, which allows users to send messages to each other in real-time. The communication should occur only within the group.
Scenario:
Sara, David and Tom collaborate within a group, participating in a collaborative broadcast content format. They communicate within the group via a messenger.

Symptoms:
This pattern should be applied when
- Collaboration is not fruitful because of missing communication possibilities.
- Oral communication is not possible.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

Digital Emotions *

Context:
Within a group or community, users should express their opinions with digital emotions. Digital emotions, in turn, can be further analyzed to modify the discussion of a show, the genre of a music channel or the competition between two candidates in a studio in, for example, building a sandcastle.

Problem:
People within a community have no further communication space.

Solution:
Provide at least expressing emotion via the gui, or further show emotions automatically by measuring the viewer’s speech, face, gesticulations, mimics or similar.

Scenario:
In a TV game show, two in-studio participants are competing in building a sandcastle. Participators score the sandcastles by expressing their emotions.

Symptoms:
This pattern should be applied when
- A low level form of expression is needed.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

Collaborative Artefact Management

Vote *

Context:
A community relating to a musician TV show should give their opinions on the content.

Problem:
An anonymous collaborative tool is required, where participants can express their opinions and make decisions easily.

Solution:
Provide an easy set up for doing a poll. After the vote is over, present the result on television or process it for further modification.
Scenario:
An open community to a TV channel should vote for a “Big Bang Theory\(^1\)” marathon.

Symptoms:
This pattern should be applied when
- The group or community’s opinion matters to the broadcaster.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.4.5.2 Floor Control *

Context:
Participants of a collaborative broadcast content format interact synchronously as a closed group in a collaboration space, but only one user at a time acts in the shared space.

Problem:
Synchronous interaction within a time critical environment can lead to conflicting actions confusing the collaborators.

Solution:
Create a token based interaction model, where the token owner is allowed to modify or access the shared resources. A fair group process, which passes the token among the collaborators, or an external coordinator which determines the user that is allowed to do modifications for a certain time is necessary.

Scenario:
Tim, Tom and Tony participate as one private group in a broadcast content format. The coordinator is the content format, which decides that it is Tony’s turn to guess the next letter in the guessing show.

Symptoms:
This pattern should be applied when
- Conflicting and parallel actions confuse interacting users.
- Parallel use of shared resources is not possible

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

C.5 COLLABORATION – COLLABORATION SUPPORT

This section lists patterns, firstly, to provide collaboration and, secondly, to make the outcome of the collaborative activity available for further processing. It is assumed that users (the audience) have found a collaboration context (which is the TV content) and that they want to perform tasks that advance the group towards its collective goal. Therefore, these patterns should be considered if collaborative activity is linked to the broadcast content format and if the outcome of the collaboration should be collected, analyzed and/or further processed somehow. The patterns mainly describe how the collaborative activity is provided, how collaborative activity is measured and how the outcome is made available.

\(^1\) American Sitcom [http://www.cbs.com/shows/big_bang_theory/]
of the collaborative activity is identified, observed and processed. This layer defines just the basic support for collaboration, not specific collaborative application, which allows the continuous composition and add-on of collaborative applications.

c.5.1  Collaboration Provision

c.5.1.1  Schedule Collaborative Activity *

Context:
Enable and disable collaborative activity. Coordinate group activity by scheduling the collaborative activity.

Problem:
Collaborative activity should be coordinated by the collaborative broadcast content format. Therefore, collaborative activity should not be active all the time.

Solution:
Coordinate group activity by enabling and disabling collaborative activity by including triggers into the broadcast medium anchored at previously defined tags. Tags in the medium could be frames or timestamps.

Scenario:
A collaborative broadcast content format provides several rounds of games, where people at home compete against the participants in the TV studio. Several tasks have to be solved, which requires different collaborative functionality at different timestamps for different intervals. The coordination happens by the inclusion of triggers (for instance, as the metadata are) into the broadcast medium.

Symptoms:
This pattern should be applied when
- Collaborative activity should be activated at specific timestamps for a certain duration.

A generalization and more extensive description of this pattern can be found in the book “Patterns for Computer-Mediated Interaction” by Schümmer et al [74].

c.5.2  Collaboration Analytics

c.5.2.1  Simple Majority

Context:
Collaborative activity should be analyzed to modify further broadcast scenarios or to analyze the results of the winning team.

Problem:
The collaborative outcome is of interest to the broadcaster and needs to be itemized. Analyze the opposite or prevailing opinion of a group on any topic.

Solution:
Do a simple voting procedure to get the simple majority.
Scenario:
Thomas is member of the open CBF community, therefore he is invited to pass his opinions on sandcastles, which were built by two competing participants during a live game show.

Symptoms:
This pattern should be applied when
• Collaborative activity should modify the further course or characteristics of the broadcast content format.
• The opposite or prevailing opinion groups are of interest.

c.5.2.2 Raised Questions

Context:
Collaborative activity should be analyzed to modify further broadcast scenarios or to analyze the results of the winning team.

Problem:
The collaborative outcome is of interest to the broadcaster and needs to be itemized. Raising questions and ambiguities within groups should be detected.

Solution:
Analyze the live chat protocols of the single groups (with their agreement) for repeating questions, keywords, tags or similar. Provide avatars, emoticons, simple radio buttons or check boxes to allow the participants to express their moods.

Scenario:
Thomas is member of the open CBF community, therefore he is invited to participate in a chat correlating to a live discussion show about a current political topic. In the chat, people have the possibility to guide the live discussion of the show within the topic. In the chat, recurring thoughts, questions and opinions relating to the topic are projected in the studio, the discussion people include them in their discussion or answer.

Symptoms:
This pattern should be applied when
• Collaborative activity should modify the further course or characteristics of the broadcast content format.
• The opposite or prevailing opinion groups are of interest.
• Specific morale, questions and thoughts of a group are of interest.

c.5.2.3 Gaming Results

Context:
Collaborative activity should be analyzed to modify further broadcast scenarios or to analyze the results of the winning team.

Problem:
Groups compete in a collaborative broadcast content format where different tasks, puzzles and games have to be solved. The results of the single groups have to be identified, observed and processed.
Solution:
Provide a graphical user interface on first or second screens, which allows each group to publish their results.

Scenario:
Several groups are competing in a collaborative gaming show. The single games are presented in a graphical user interface on the first or second screens of the collaborators. The overall results of the games are displayed and sent to the broadcaster who provides a high-score of the competing groups.

Symptoms:
This pattern should be applied when
• Collaborative activity should modify the further course or characteristics of the broadcast content format.
• Collaborative groups are competing each others.
• Results of competing groups should be scored and stored for further episodes of the broadcast content format or other usage.

c.5.3 Collaboration Interaction Observation

c.5.3.1 Level of Activity

Context:
Collaborative activity should be identified and observed - not its content.

Problem:
It has to be considered if collaboration happens and to which extent.

Solution:
Measure the level of activity within a group or community and find out if it is group/community interaction or single activity. For example, measure the number of messages between group members or measure the number of actions on a shared object within a certain time frame.

Scenario:
To check which participating groups are alive and active, the level of their activity is measured. For this, the groups automatically send an activity message back to the broadcaster about messages sent within the groups and operations executed on the shared object, which is a puzzle to be solved together.

Symptoms:
Provide this pattern when
• Collaborative activity should modify the further course or characteristics of the broadcast content format.
• The level of activity is of interest.
• The level of activity should be quantified.

c.5.3.2 Kind of Activity

Context:
If collaborative activity happens, analyze which one.
Problem:
Which collaboration happens?

Solution:
Measure the level of activity within a group or community by using the “Level of Activity” pattern and find out if it is group interaction or single activity. For example, measure the number of messages between group members and measure the number of actions on single shared objects and detect the peaks.

Scenario:
To check the activity for alive groups, the groups automatically send status messages back to the broadcaster about a sent message within a group and about an operation on the shared object, which is a puzzle to be solved together. The broadcaster automatically analyses the status messages for peaks.

Symptoms:
Provide this pattern when
- Collaborative activity should modify the further course or characteristics of the broadcast content format.
- The kind of collaborative activity is of interest.
C.6 CMC LINK

This section describes patterns about how the broadcast technology as medium, the program content and collaborative activity are finally linked. More precisely, they define examples of trigger and linking parameters of collaborative applications in the form of XML metadata. Other elements to trigger collaborative activity are conceivable, whereas this category is open to be extended. Target groups are the producers, who have to consider, design and plan the link of collaborative activity to broadcast content formats during pre-production and production phases and, finally, to include the trigger (metadata in this case) in the broadcast medium during the post-production phase.

c.6.1 XML Trigger

c.6.1.1 Enabler

Context:
A certain collaborative application should be enabled, anchored to a specific tag (e.g., time stamp).

Problem:
A certain collaborative application should be available only at certain conditions.

Solution:
Include metadata into the broadcast medium, anchored to a specific tag, to enable a collaborative application.

The minimum DTD is as follows:

```
1 <!ELEMENT basics (id, name, parameter, repository, packetUpdate, guiUpdate, round, gui_enabler)>
2 <!ELEMENT id (#CDATA)> integer of unique identifier
3 <!ELEMENT name (#CDATA)>
4 <!ELEMENT repository (#CDATA)>
5 <!ELEMENT packetUpdate (#CDATA)> 0 - update packet, 1 - store packet
6 <!ELEMENT guiUpdate (#CDATA)> 0 - gui is not updated, 1 - update gui
7 <!ELEMENT round (#CDATA)> integer of round number
8 <!ELEMENT gui_enabler (#CDATA)> 0 - gui is not shown/offered, 1 - show gui
```

Basically, the element gui_enabler needs to be 1 to make the collaborative application available.

Scenario:
The collaborative voting application should be enabled from the fifth minute of the collaborative TV show. 05:00 is the tag to which the enabler is added into the medium.

```
1 <?xml version="1.0"?>
2 <application>
3   <id>99</id>
4   <packetUpdate>1</packetUpdate>
5   <guiUpdate>1</guiUpdate>
6   <gui_enabler>1</gui_enabler>
```
Symptoms:
This pattern should be applied when
• A collaborative application should be enabled by a certain tag.

c.6.1.2 Disabler

Context:
A certain collaborative application should be disabled only at certain conditions.

Problem:
A certain collaborative application should be available only under certain conditions.

Solution:
Include metadata into the broadcast medium, anchored to a specific tag, to disable a collaborative application.

The minimum DTD is as follows:

```
<!ELEMENT basics (id, name, parameter, repository, packetUpdate, guiUpdate, round, gui_enabler)>
<!ELEMENT id (#CDATA)> integer of unique identifier
<!ELEMENT name (#CDATA)>
<!ELEMENT repository (#CDATA)>
<!ELEMENT packetUpdate (#CDATA)> 0 - update packet, 1 - store packet
<!ELEMENT guiUpdate (#CDATA)> 0 - gui is not updated, 1 - update gui
<!ELEMENT round (#CDATA)> integer of round number
<!ELEMENT gui_enabler (#CDATA)> 0 - gui is not shown/offered, 1 - show gui
```

Basically, the element `gui_enabler` needs to be 0 to disable the collaborative application.

Scenario:
The collaborative voting application should be available only during the broadcast of the collaborative TV show. **90:00** is the tag to which the disabler is added into the medium.

```
<?xml version="1.0"?>
<application>
  <id>99</id>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>1</guiUpdate>
  <gui_enabler>1</gui_enabler>
</application>
```

90:00 If the TV show takes 90 minutes, the disabler is added into the medium at this point in time.

```
<?xml version="1.0"?>
<application>
  <id>99</id>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>1</guiUpdate>
</application>
```
Symptoms:
This pattern should be applied when
• A collaborative application should be disabled by a certain tag.

c.6.1.3 Updater

Context:
A certain collaborative application should be updated at certain conditions.

Problem:
A collaborative application needs updated information. The Updater pattern focuses on updating existing packages in the queue.

Solution:
Include metadata into the broadcast medium, anchored to a certain tag to update a specific collaborative application.
Additionally to the data to be updated, the minimum DTD is as follows

```
<!ELEMENT basics (id, packetUpdate>  
<!ELEMENT id (#CDATA)> integer of unique identifier  
<!ELEMENT packetUpdate (#CDATA)> 0 - update packet, 1 - store packet
```

The id must be set to the application id and the packetUpdate element needs to be set to 0.

Scenario:
The user interface of the collaborative service to a gaming show is updated with the content for the new gaming round. For instance:

```
<?xml version="1.0"?>  
<application>  
 <id>55</id>  
 <packetUpdate>1</packetUpdate>  
 <guiUpdate>1</guiUpdate>  
 <round>1</round>  
 <path>../CA_Quiz</path>  
 <name>CBF Gaming Application</name>  
 <task>Task: Guess the Phrase</task>  
 <question>any word or phrase to guess</question>  
</application>
```

Symptoms:
This pattern should be applied when
• A collaborative application should be updated under certain conditions.

c.6.1.4 Coordinator

Context:
Coordinate collaborative activity.

Problem:
The activity of collaborators in a group needs to be coordinated.
Solution:
Include metadata into the broadcast medium, anchored at a certain tag to coordinate collaborative activity.

The minimum DTD is as follows:

1. `<!ELEMENT collaboration (id, task, peer)>`
2. `<!ELEMENT id (#CDATA)>` integer of unique identifier
3. `<!ELEMENT task (#CDATA)>` which task the peer needs to solve
4. `<!ELEMENT peer (#CDATA)>` number of the peer in the group

Scenario:
One example of a coordinator pattern could be the coordination of the peers to guess in a quiz application.

Symptoms:
This pattern should be applied when
- A group needs to be coordinated.
Below, the metadata which were included into the 2.34 Minutes test sequence, and the measurement results of the metadata extraction and the CPU and memory usage are listed.

### D.1 METADATA

<table>
<thead>
<tr>
<th>#</th>
<th>ID</th>
<th>Timestamp (Experiment #1)</th>
<th>Timestamp (Experiment #2)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ca_e</td>
<td>0'06</td>
<td>0'02</td>
<td>Enable the collaborative application</td>
</tr>
<tr>
<td>2</td>
<td>ca_r</td>
<td>0'21</td>
<td>0'03</td>
<td>Send / show the rules of the game show</td>
</tr>
<tr>
<td>3</td>
<td>g_d</td>
<td>0'36</td>
<td>0'04</td>
<td>Disable the group building process</td>
</tr>
<tr>
<td>4</td>
<td>01_r</td>
<td>0'51</td>
<td>0'07</td>
<td>Send / show the rules for game #1</td>
</tr>
<tr>
<td>5</td>
<td>01_e</td>
<td>1'06</td>
<td>0'08</td>
<td>Enable game #1</td>
</tr>
<tr>
<td>6</td>
<td>01_c</td>
<td>1'21</td>
<td>0'09</td>
<td>Send coordinators for game #1 at an interval of 10 seconds in a period of 4'00 minutes</td>
</tr>
<tr>
<td>7</td>
<td>01_d</td>
<td>1'36</td>
<td>0'11</td>
<td>Disable game #1</td>
</tr>
<tr>
<td>33</td>
<td>ca_d</td>
<td>1'51</td>
<td>0'12</td>
<td>Disable notifier for all user modes</td>
</tr>
</tbody>
</table>

Table 47: Schedule of the XML Metadata Merged for this Scenario

Figure 75 and Table 47 recap the timeline of the experimental 2 minute 34 seconds video sequence with metadata included in the first experiment constantly every 15 seconds, beginning with the first package at timestamp 0'06. In the second experiment (cf. Figure 75 below), the metadata were included irregularly and partly with a minimum distance of one second. Note: The measurements stop after the last metadata package.

### #1 - Enabler (0’06 / 0’02)

```xml
<?xml version="1.0"?>
<application>
  <id>55</id>
  <guiEnabler>1</guiEnabler>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>1</guiUpdate>
  <round>0</round>
  <path>../CBF_Gaming_App</path>
  <name>CBF Gaming Application</name>
  <repository>ftp://123.45.67.890</repository>
</application>
```
This first package enables the gaming application to the viewer. When the package is processed, a notification is shown to the viewer that the collaborative application is available. The packet is split into two MPEG-TS frames since its file size is 315 bytes.

#2 - Updater (0’21 / 0’03)

Your team of three members competes live against 2 teams of collaborators playing in the TV studio. Your team is assembled automatically, so you do not know each other but you can communicate via the collaborative application. The candidates in the studio also do not know each other.
The teams play against each other and have to solve previously defined tasks. The team that solves the task faster or better collects points and wins the task. The team that collects the most points is the winner.

The second package is an updater and sends the introduction and rules to the application. The GUI of the application is updated and the rules are presented. The packet is split into five MPEG-TS frames since its file size is 881 bytes.

### #3 - Disabler / Coordinator (0'36 / 0'04)

```
<?xml version="1.0"?>
<application>
  <id>55</id>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>0</guiUpdate>
  <round>1</round>
  <path>../CBF_Gaming_App</path>
  <task>Group building is done</task>
</application>
```

The third package is a disabler, which disables the group building process. From now on, no newly formed groups are accepted. The packet is split into two MPEG-TS frames, since its file size is 215 bytes.

### #4 - Updater (0'51 / 0'07)

```
<?xml version="1.0"?>
<application>
  <id>55</id>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>1</guiUpdate>
  <round>0</round>
  <path>../CBF_Gaming_App</path>
  <name>CBF Gaming Application</name>
  <description>
    A given phrase needs to be guessed by the teams within 4 minutes. In each round, one letter can be guessed by one candidate. The candidate who guesses is chosen randomly, but one candidate may guess several times consecutively. A maximum of eleven errors are allowed. If they were made, the team loses one point. If one team guessed the phrase, the game is over and the team gets the point. None of the other teams get points.
  </description>
</application>
```

Package number four sends the rules for the first game. The gui is updated and the text is presented to the participants. The packet is split into four MPEG-TS frames, since its file size is 689 bytes.
### #5 - Enabler (1'06 / 0'08)

```xml
<?xml version="1.0"?>
<application>
  <id>55</id>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>1</guiUpdate>
  <round>1</round>
  <path>../CBF_Gaming_App</path>
  <name>CBF Gaming Application</name>
  <task>Starting Round #1</task>
  <question>Any word or phrase to guess</question>
</application>
```

Package number five sends the question, which must be solved by the teams. The gui is updated and the task is presented. The packet is split into two MPEG-TS frames, since its file size is 299 bytes.

### #6 - Coordinator (1'21 / 0'09)

```xml
<?xml version="1.0"?>
<application>
  <id>55</id>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>0</guiUpdate>
  <round>1</round>
  <path>../CBF_Gaming_App</path>
  <name>CBF Gaming Application</name>
  <task>It's peer 2's turn to guess</task>
  <peerPos>2</peerPos>
</application>
```

The sixth package is a coordinator, which coordinates that it is peers number 2's turn to guess. The packet is split into two MPEG-TS frames, since its file size is 281 bytes.

### #7 - Disabler (1'36 / 0'11)

```xml
<?xml version="1.0"?>
<application>
  <id>55</id>
  <packetUpdate>1</packetUpdate>
  <guiUpdate>1</guiUpdate>
  <round>1</round>
  <path>../CBF_Gaming_App</path>
  <name>CBF Gaming Application</name>
  <task>Finishing Round #1</task>
</application>
```

Package number seven disables the first game. The packet is split into two MPEG-TS frames, since its file size is 249 bytes.
D.2 METADATA EXTRACTION RESULTS

Hereafter, the measurement results of the metadata extraction are constituted. The results are divided into the four steps of (i) extraction, (ii) PTS check, (iii) data frame extraction and (iv) process.

Step 1 - Extraction of MPEG-TS data frames into data picture queue
Figures 76 and 77 sketch the results of step one. The X axis shows the timestamp of the data frames (cf. Table 47 and Figure 75), the Y axis scales the duration of step one in microseconds.

![Metadata Extraction - Step 1](image_url)

Figure 76: Measuring Step 1 of the Metadata Extraction (Experiment #1)
Figure 77: Measuring Step 1 of the Metadata Extraction (Experiment #2)

Step 2 - Check PTS of the next data frame with the PTS of the current video frame

Figures 78 and 79 summarize the overhead of the PTS comparison. The data stream is synchronized with the video stream. Accordingly, the comparison of each displayed video frame is done at a frame rate of 29.97 frames per second. The X axis shows the PTS of the video frames in steps of one to two seconds. The Y axis scales the duration of the comparison in microseconds.

Figure 78: Measuring Step 2 of the Metadata Extraction (Experiment #1)
D.2 metadata extraction results

Figure 79: Measuring Step 2 of the Metadata Extraction (Experiment #2)

Step 3 - Extract data frame and commit it to the corresponding application
Figures 80 and 81 sketch the extraction and committal of the current metadata package.

Figure 80: Measuring Step 3 of the Metadata Extraction (Experiment #1)

The X axis shows the timestamp of the data frames (cf. Table 47 and Figure 75). The Y axis scales the duration of the extraction and committal process in microseconds.
D.2 Metadata extraction results

Step 4 - Process the data frame

Figures 82, 83, 84 and 85 outline the processing time of the data frame by the corresponding application. The X axis shows the timestamp of the data frames (cf. Table 47 and Figure 75). The Y axis scales the duration of the extraction and committal process in microseconds.

Figure 81: Measuring Step 3 of the Metadata Extraction (Experiment #2)

Figure 82: Measuring Step 4 of the Metadata Extraction (Experiment #1)
Figure 83: Measuring Step 4 of the Metadata Extraction (Experiment #1)

Figure 84: Measuring Step 4 of the Metadata Extraction (Experiment #2)

Figure 85: Measuring Step 4 of the Metadata Extraction (Experiment #2)
BIBLIOGRAPHY

[1] FFmpeg Documentation (for FFmpeg version 0.7).

[18] Shayne Bowman and Chris Willis. We media - how audiences are shaping the future of news and information. online, July 2003.


All references and web addresses were checked in September 2015.
Curriculum Vitae

Dipl.-Ing.in Sabine Lösch

Johannes Kepler University Linz
Department of Telecooperation

Altenberger Straße 69, 4040 Linz, Austria
sabine.loesch@jku.at

Version: July, 2015
PERSONAL DATA

Name: Dipl.-Ing. in Sabine Lösch
Address: Linzer Straße 6
4532 Rohr im Kremstal
Austria
Mobile Phone (private): +43 650 / 63-55-300
Date of Birth: May 2nd, 1981
Place of Birth: Linz, Oberösterreich
Civil Status: Married
Children: Jakob (May 15th, 2012), Eva (January 27th, 2014)
Nationality: Austrian
Language Skills: German (native), English (fluent), French (high school level)
Finnish (basic)

EDUCATION

1988 – 1992 Dr.-Ernst-Koref-Schule (VS 29), Ramsauerstraße 61, 4020 Linz
1996 – 2002 Higher Education for Economic Occupations, Landwiedstraße 80, 4020 Linz

ACADEMIC STUDY

Johannes Kepler University

Degree: October 30th, 2006
Bachelor thesis: „Passive Performance Measurements in Computer Networks“ In cooperation with the Blekinge Institute of Technology, Karlskrona Sweden

Nov. 2006 – July 2008 Master-Study in Computer Science
Specialization in Pervasive Computing / Embedded Systems
Degree: July 18th, 2008 (Graduation passed with distinction)
Masters thesis: „XML-Databinding for File Systems“ Funded by Sony DADC, Salzburg Austria

August 2008 – September 2015 PhD-Study in Computer Science
PhD thesis: “Direct Integration of Collaboration into Broadcast Content Formats”
First Supervisor: Univ.Prof. in Dr. in Gabriele Anderst-Kotsis (Johannes Kepler University Linz, Austria)
Second Supervisor: Associate Prof. Dr. Artur Lugmayr (Curtin University, Perth Australia)
University of Helsinki (and Helsinki University of Technology)

Jan. 5th – July 16th 2007  International Exchange Student Program in Helsinki Finland

FURTHER EDUCATION

Sept. 2009 – Sept. 2010  University didactic class at the Johannes Kepler University Linz
Courses on didactics, presentation skills, potential assessment, research skills

Nov. 12th – Nov. 13th 2010  PHD Training in Bratislava, organized by SAISA and the Austrian Federal Ministry of Research
Training of presentation skills and proposal writing

Oct. 2010 – May 2011  Participation in "Lehrgang karriere_links" at the Johannes Kepler University Linz
Courses on project management, communication, leadership, research administration

WORK EXPERIENCE

Currently:
July 1st, 2015 - Present  Part-time University Assistant at Johannes Kepler University Linz, Department of Telecooperation.
Head of department: Univ.Prof. in Dr. in Gabriele Anderst-Kotsis
Line of activity: Research work in the field of multimedia, supervision of students as well as upcoming functions in the research area (reviews, conference organization, etc.)

August 1st, 2008 – April 26th, 2015  Full time University Assistant at Johannes Kepler University Linz, Department of Telecooperation.
Head of department: Univ.Prof. in Dr. in Gabriele Anderst-Kotsis
Line of activity: PhD in the field of multimedia, giving lectures, supervision of students as well as upcoming functions in the research area (reviews, conference organization, etc.)

March 23rd, 2012 – Feb. 27th, 2015  Parental leave for both children

Feb. 1st, 2008 – June 30th, 2008  Freelancer at Sony DADC Anif, Salzburg
Line of activity: Software engineer in the context of my masters thesis in Computer Science (Johannes Kepler University Linz)

March 21st, 2006 – Nov. 17th, 2007  Project “Admina.at”
Line of activity: Organizing and giving workshops for female first-year students to facilitate the access to technical fields of study
Focus of workshops: computer hardware, Java, Windows XP, Linux
Sep. 1st, 2004 – Jan. 31st, 2008  Freelancer at voestalpine Stahl GmbH (division steel), Strategic IT-Management (10 – 15 hours per week)
Line of activity: project work i.a. in IT-Security, process (re-)organization, data harmonization, editorial project work

Sep. 29th, 2004 – Dec. 31st, 2005  Freelancer at LifeTool Linz (8 hours per week)
Line of activity: i.a. testing of software systems

March 1st – March 31st, 2004  Editorial project work at voestalpine Stahl GmbH (division steel), Strategic IT-Management


Aug. 5th – Sep. 1st, 2002  Internship at Oberbank AG, district office "Neue Heimat"

DUTIES IN UNIVERSITY COMMITTEES

October 2010 – May 2012  Member of the Faculty Assembly for the Faculty of Engineering and Natural Sciences

October 2010 – May 2012  Representative of the Research Assistants in the Governing Panel of the Department of Telecooperation (Institutskonferenz)

October 2010 – May 2012  Member of the Committee for Business Informatics Curriculum Development (Studienkommission)

2011 – May 2012  Member of the Committee for Students Grants for the Faculty of Engineering and Natural Sciences

TEACHING EXPERIENCE

Summer term 2009: Exercise „Multimedia systems for Bachelor-Study in Computer Science“ (4 hours per week)
Summer term 2010: Exercise „Multimedia systems for Bachelor-Study in Computer Science“ (4 hours per week)
Summer term 2011: Exercise „Multimedia systems for Bachelor-Study in Computer Science“ (3 hours per week)

RESEARCH ACTIVITY

PhD

Topic: „Direct Integration of Collaboration in Broadcast Content Formats“
First Supervisor: Univ.Prof. Dr. Gabriele Anderst-Kotsis (Johannes Kepler University Linz, Austria)
Second Supervisor: Associate Prof. Dr. Artur Lugmayr (Curtin University, Perth Australia)
Conferences

Volunteer Conference Assistant:
10th International Conference on Information Integration and Web-based Applications & Services (iiWAS2008) and 6th International Conference on Advances in Mobile Computing & Multimedia (MoMM2008), Linz, Austria, November 24th – 26th, 2008.

Organizing Committee:
10th International Conference on Information Integration and Web-based Applications & Services (iiWAS2008) and 6th International Conference on Advances in Mobile Computing & Multimedia (MoMM2008), Linz, Austria, November 24th – 26th, 2008.

Publicity Chair:
11th International Conference on Information Integration and Web-based Applications & Services (iiWAS2009) and 7th International Conference on Advances in Mobile Computing & Multimedia (MoMM2009), Kuala Lumpur, Malaysia, December 14th – 16th, 2009.

Reviewer:
15th MindTrek Conference and the International Academic Conference (Academic Track), Tampere, Finland, September 28th – 30th, 2011.

Participant:
7th International Conference on Advances in Mobile Computing & Multimedia (MoMM2009), Kuala Lumpur, Malaysia, December 14th - 16th, 2009.

Scientific Talks

“Collaboration meets Digital Television - Fact or Fiction?”

“New Social and Collaborative Interactive TV Program Formats”
7th International Conference on Advances in Mobile Computing and Multimedia (MoMM2009), December 14th - 16th, 2009, Kuala Lumpur, Malaysia

“The Convergence of TV and Web (2.0) in Austria and Finland”

“Collaboration in Broadcast Media and Content”
Visiting Scholar
June 8th - 22nd, 2010, Tampere University of Technology, Tampere, Finland
June 6th - 11th, 2011, Tampere University of Technology, Tampere, Finland

Publications

Sabine Bachmayer, "XML Data Binding for File Systems", July 2008, Master thesis supervised by the Department of Telecooperation (Johannes Kepler University Linz) and funded by Sony DADC Austria


Sabine Bachmayer, Artur Lugmayr, Gabriele Kotsis, "Convergence of Collaborative Web Approaches and Interactive TV Program Formats", in International Journal of Web Information Systems, Vol. 6, Number 1, Emerald, Pages 74-94, 2010, ISSN: 1744-0084


