MC^2: A Framework and Service for MPEG-7 Content-Modelling Communities

HARRY AGIUS*, MARIOS C. ANGELIDES AND DAMON DAYLAMANI ZAD

Electronic and Computer Engineering, School of Engineering and Design, Brunel University, Uxbridge, Middlesex UB8 3PH, UK
*Corresponding author: harryagius@acm.org

Harnessing the power of Web communities, the effort on creating metadata can be greatly reduced. Collaborative communities can create, update and maintain content models for multimedia resources more effectively than single users working alone. This paper presents MC^2, a framework for MPEG-7 content-modelling communities. MC^2 is based on the challenges to collaborative multimedia content modelling reported in the research literature and the results of an experiment undertaken to investigate user behaviour in collaborative content modelling. An MC^2 service has also been implemented as a proof of concept for this framework, which is evaluated with a population of users and against the challenges.

Keywords: MPEG-7; collaboration; content modelling; communities; metadata; standards

Received 29 February 2012; revised 18 May 2012
Handling editor: Marios C. Angelides

1. INTRODUCTION

The use of multimedia on the Web has grown massively in recent years and Internet traffic is predicted to quadruple in the near future predominantly due to online video [1]. Video-on-demand Websites such as BBC iPlayer, 4oD and Sky Anytime+ in the UK and Hulu and Netflix in North America, along with online global video Websites such as YouTube and social networking Websites such as Facebook, have changed user habits and attitudes towards Internet video usage and the statistics are staggering. While Google processes around 700 000 search results every minute [2], 60 h worth of videos are uploaded to YouTube every minute with 4 billion videos (100 million hours of video) viewed per day by over 27 million visitors [3, 4]. In 1 day, over 250 million photos are also uploaded to Facebook and 2.7 billion Likes and Comments are posted by 483 million daily active users. In total, Facebook stores over 100 PB of photos and videos [5].

Consequently, Web 2.0 can be said to have given rise to increasing use of multimedia content on the Web, leading to the emergence of many collaborative multimedia environments such as Flickr, YouTube and Delicious where users are able to share and view multimedia content in a community. These folksonomic communities or social-tagging applications allow users to describe multimedia content using simple, flat metadata such as titles, categories and tags. These applications suggest that users are willing to model multimedia content in a collaborative manner and therefore it is possible to extend this approach to more advanced media content-modelling applications, such as those based on the comprehensive, extensible MPEG-7 standard. In this way, the power of Web communities would be harnessed to overcome what is otherwise an intensely time consuming task for single users of creating, updating and maintaining content models for multimedia resources. When it is considered that non-expert users tend to add only three to four tags per content item on average [6], it also becomes apparent that there is great potential for improving the richness of the content models through collaboration.

However, there is a semantic gap between the current folksonomic communities and real-world semantics that stops the models created by these tools from being fully and comprehensively mapped. For example, in YouTube, the user has no means of clearly defining temporal relationships or defining the time an event starts and finishes. Other challenges include overcoming spelling mistakes, different terms being
used by different users to describe the same content and creating community feeling so that users productively engage, which currently impose barriers for media retrieval and affect the usefulness of the content models. When these challenges are overcome, more detailed and accurate content models become possible and a greater amount of multimedia resources are able to be content modelled.

In this paper, we present MC², a framework and service for MPEG-7 content-modelling communities. The framework consists of a system architecture for collaborative multimedia content-modelling communities along with an MPEG-7 profile for collaborative content modelling. The framework is based on the results of a user experiment and serves to address the challenges that stand in the way of collaborative multimedia content-modelling communities. The rest of this paper is organized as follows. In Section 2, we review related research work which reveals the challenges in creating a collaborative multimedia content-modelling community. Then, in Section 3, we briefly summarize our prior experiment design and results, from which we derived our framework requirements. In Section 4, we present the development of the collaborative multimedia content-modelling framework. Section 5 contributes the MC² service, which is an implementation of the framework and serves as a proof of concept. Finally, in Section 6, we validate the framework through an experiment and a comparison against the challenges, before concluding the paper in Section 7 with a discussion of the impacts and future directions of the research.

2. RELATED RESEARCH WORK

The amount and variety of multimedia content available on the Web has grown at an exponential rate in recent years, and thus the need to annotate multimedia content has become ever more imperative in order for users to be able to access these multimedia resources effectively [7]. Recent research in multimedia annotation approaches has witnessed an increasing interest in collaboration, not least due to the ability of Web 2.0 to support communities through the reuse and amalgamation of different Web services to provide rich application experiences. For example, YouTube integrates video streaming and forum technologies with AJAX to support video-based communities. In this section, we review the research literature to examine the challenges to collaborative multimedia content modelling. We organize the challenges under criteria from [8] which are traditionally used for data clustering but are adapted here for use in the context of content modelling:

1. Model stability: In data clustering, stability is the degree by which the learning rate tends towards zero after a finite number of learning iterations and no pattern changes their clustering thereafter. For content modelling, we adapt this definition to mean the degree by which the model stabilizes after a finite number of metadata additions so that no further model elements are required in order to achieve full closure of the semantic gap. Therefore, this is a measure of the degree of completeness of relationships between the semantics of the content and the metadata of the content model.

   (i) **Model plasticity:** In data clustering, this is the ability of the algorithm to adapt to new data without having to re-cluster. For content modelling, we adapt this definition to mean the degree by which additional tags may be incorporated within the model. Therefore, this is a measure of the degree of extensibility.

   (ii) **Model accuracy:** In data clustering, this refers to the degree with which clustering classes reference ground truth classes. For content modelling, we adapt this definition to refer to the degree of correlation between the real-world and the associated metadata; that is, the goal of the model. Therefore, this is a measure of the degree of correlation.

Figure 1 presents how the challenges are organized within the above categories.

![Figure 1](http://comjnl.oxfordjournals.org/)
great variety regarding what keywords and phrases they use within tags and how frequently they use them, stable, aggregate patterns are emerging that can be used to help structure the tagging process [9]. Ontologies also help provide pattern stability and one way to use a formal ontology with collaborative tagging is to derive it from the tags deployed within the system through data mining [10]. Ulges et al. [11] present a system that automatically tags videos by detecting high-level semantic concepts, such as objects or actions, through using videos from online portals like YouTube as a source of training data, while tags provided by users during upload serve as ground truth annotations. Another method is to employ ontology seeding, which embeds an ontology into the system before the users commence tagging and typically asks the users for additional semantic information to ensure that the tags they contribute follow the conventions of the ontology [10]. FolksAnnotation [12], a system that extracts tags from del.ici.ous and maps them to various ontological concepts, has helped to demonstrate that semantics can be derived from tags. Queueo.tv [13] is a recommendation system that uses tags added by users for various videos to build user and item tag clouds that allow for creating a folksonomy of the tags used by users to arrive at a semantic pattern for recommendations. However, before any ontological mapping can occur, the vocabulary must usually be converted to a consistent format for string comparison.

Studies, e.g. [10, 14], have shown that different users adopt different perspectives on how multimedia content should be modelled. Consequently, in order to understand which users have authored or updated specific elements of the metadata, to know which users are presently active, or to be able to simply contact users, identity awareness is also required. This includes self-awareness such that users may identify their own changes and activity. Patterns and profiles can be used here to both raise awareness about users and increase collaboration standards between users [15]. Kim et al. [16] have built collaborative user models (profiles) by leveraging user-generated tags. They take into account positive and negative tags the user has entered and from these they deduce the user’s likes and dislikes. Moreover, there have been several community configuration management tools developed, such as Palantir [17], a workspace awareness tool that informs a user of which other users have changed artefacts, calculates the severity of the changes and graphically displays the information in a configurable and generally non-obtrusive manner. As with logging and revision control systems, however, community configuration management tools have yet to be adapted to and implemented in multimedia communities, where additional, distinct activities must be catered for, such as updating different content features and media streaming.

2.2. Model plasticity

The challenges in this category focus on the extensibility of the model created through collaborative content modelling. This category includes the following challenges: synonym control, connectedness and metadata propagation.

One of the challenges in collaborative content modelling is that tags are highly individualistic and different users often use a variety of terms to describe the same concept [18]. This makes searching problematic since users may only search for one synonym but require results for the entire group of synonyms. In contrast to traditional subject indexing, metadata in communities is generated not only by experts but also by creators and consumers of the content. Usually, freely chosen keywords are used instead of a controlled vocabulary [19]. Batch et al. [20] point out that using free words chosen by users leads to tags being inefficient in terms of meanings and semantic interlinking capabilities and therefore present MTag [20] as a model for collaborative medical tagging. Consequently, such a lack of synonym control makes searching problematic since users may only search for one synonym but require results for the entire group of synonyms.

Exploiting the full connectedness of the tags is a further challenge. The full extent of relationships between tags that make sense to human beings is difficult to develop within tag-based systems and thus they are often not implemented. For example, if a user searches for ‘swimming’, the results should ideally include all terms related to swimming, not just synonyms but also related concepts such as medals, Olympic events and so on. Language tools to identify synonyms, acronyms and relationships have sought to address the above issues. One of the most popular tools used is WordNet [21], which is an English semantic lexicon. It groups English words into sets of synonyms, provides general definitions and acronyms, and records the various semantic relations between terms. At the core of WordNet is the use of synsets (also known as synonym rings). A synset is a group of words that denote the same concept and are interchangeable in many contexts. Each synset is linked to other synsets by means of conceptual relations. Some approaches build on the notion of a text-based synset for image annotation, such that a visual synset is used as an organization of images which are visually similar and semantically related. Each visual synset then represents a single prototypical visual concept with an associated set of weighted annotations [22, 23]. WordNet has been used to solve tagging issues and to develop retrieval systems that utilize folksonomies, e.g. TagPlus [18] retrieves from Flickr by using WordNet to correct, identify and group tags generated by users to improve the relevance of the presented results. San Pedro et al. [24] propose an approach that uses the content duplications in various video-sharing Websites to reach connectedness by discovering various tags used by various users to annotate the same content. However, such systems are add-ons to the existing systems and not integrated into the community, which limits the breadth and depth of their functionality.

Another challenge is that of metadata propagation. Often it is the case that multiple, related media streams require tagging, which may be carried out simultaneously or sequentially.
Such streams share many common content features and thus tagging becomes a repetitive process involving a lot of redundancy. This problem is exacerbated in collaborative modelling systems where there are multiple users carrying out the tagging. For example, photos of a certain occasion will often be taken by many users who were present and these photos will contain common content features, such as the people present and the location. If the metadata could be propagated or inherited by all related photos once one, or a small number of, users had tagged them, user effort would be greatly reduced while also greatly improving the consistency of the tags [25]. Ning et al. [26] present a hybrid probabilistic model which integrates the low-level image features and high-level user provided tags to automatically tag images. For instance, with two groups of images, one tagged with ‘sky’ and ‘tree’ and the other tagged with ‘tree’ and ‘grass’, a new group of images tagged with only ‘grass’ could be predicted to have the tag ‘sky’ even though ‘grass’ and ‘sky’ have never been tagged in the same image by any users.

2.3. Model accuracy

The challenges of this section deal with the accuracy of the model created through collaborative content modelling. In other words, they focus on how accurately the model relates to the real-world semantics. The challenges of this category are: tag expressiveness, tag-based ranking, inter-referential awareness and semantic awareness.

Tagging is so widely used now that users have become comfortable with it to the point where they do not give it much conscious thought and therefore do not consider the clarity, structure and subsequent usability of the tags they create [18]. Consequently, tag expressiveness is a non-trivial challenge. For example, unstructured text in tags often results in having many tags that are difficult to interpret and relate to other tags by anyone other than the tag author [9]. Similarly, incorrectly spelled keywords within the tags can cause retrieval problems and lead to orphaned content that can only be retrieved when the query also happens to contain the same misspellings. Acronyms also prove problematic and can reduce the usability of the tags when they are not commonly accepted and are decided by the content author alone. Consequently, users may search for different acronyms or search for the full phrase instead, both of which would fail to retrieve the required content. Also, users have different intentions while tagging, hence not all the tags available are related to the content of the annotated item. This can include tags such as self-references and personal tasks (‘my husband’, ‘to do list’) or may be expressing subjective opinions and qualities (‘nice book’, ‘dark movie’) [27].

Tag-based ranking refers to the means of providing ranking of search results based on the tags that are used to annotate the media and the strength of these tags in each media. Studies show that tag-based rankings produce more relevant results than traditional rankings and clusterings [28], as demonstrated by the recommendations of online music sites and communities such as Yahoo Music and Last.fm. Zhang et al. [29] propose a video blog management model which comprises of automatic video blog annotation and user-oriented video blog search. For video blog annotation, they extract informative keywords from both the target video blog itself and relevant external resources. As well as this semantic annotation, they perform sentiment analysis on comments to obtain an overall evaluation. For video blog search, they present saliency-based matching to simulate human perception of similarity and to organize the results by personalized ranking and category-based clustering. The HeyStaks [30] system is a social Web search system designed to help users collaborate during Web search tasks and it combines collaborative recommendation techniques with main stream search engines. It relies on users’ click-throughs, voting, sharing and tagging/commenting to rank the results from a Web search. Tagging is considered a more reliable indicator of interest than a simple result click-through or vote.

Inter-referential awareness concerns how a user may refer to specific elements of the content or the associated metadata during collaborative communication with others such that this reference may be identified and understood by all parties [31], both in asynchronous and in synchronous environments. While verbal references may be sufficiently effective, though inefficient, for metadata references [32], they are less suitable for references to content within the media stream since users may be attracted to the wrong features during playback or they may become engrossed in an event such that they do not notice certain objects appearing [33]. In the physical world, looks, nods and hand gestures serve to focus attention effectively and thus one would expect suitable surrogates to be available within a collaborative environment. Both Microsoft Office and Adobe Acrobat enable users to add basic annotations to their work so that users can draw the attention of other users. Similar tools catering for audiovisual (AV) files in collaborative multimedia communities would be invaluable.

When different users work collaboratively to model the same media content, different versions of the same content model may be generated. Consequently, it is important for users within the community to be informed about the type of changes that have been made to the metadata between revisions and which elements of the metadata have been authored by which users [34] in order to correct the metadata, continue their own content modelling or rethink their own decisions and modelling approach if necessary [35]. In collaborative content-modelling systems, the changes made to a content model are not limited to just changes in phrases and keywords but more importantly relate to full semantic content features such as objects, events and the relationships between them. To help provide semantic awareness, many logging and revision control approaches have been proposed, such as IceCube [36], which enables general purpose log-based reconciliation, where logs of alterations are combined into a single merged log and, by observing object and application semantics, are ordered in such a way as to
minimize conflicts. Another method proposed by Tang et al. [37] has been group profiling, where all the actions of users are gathered, similar to social media profiles, and grouped together around a concept such as a search query. Such approaches have yet to be adapted to the semantics of multimedia content models and incorporated into multimedia communities where they must work with content, context and language. Existing multimedia communities, such as YouTube and Flickr, either do not allow editing or they do not control the changes and conflicts during multimedia annotation.

Table 1 presents a summary of all challenges along with a brief definition of each challenge.

<table>
<thead>
<tr>
<th>Category</th>
<th>Challenge</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>Pattern stability</td>
<td>To ensure a stable pattern for the metadata that conforms to the pattern of the tags created by the users</td>
</tr>
<tr>
<td></td>
<td>Identity awareness</td>
<td>To understand which users have authored or updated specific elements of the metadata, to know which users are presently active, or to be able to simply contact users</td>
</tr>
<tr>
<td>Plasticity</td>
<td>Synonym control</td>
<td>To connect tags to their synonyms</td>
</tr>
<tr>
<td></td>
<td>Connectedness</td>
<td>To implement the full extent of relationships between tags in a way that makes sense to human users</td>
</tr>
<tr>
<td></td>
<td>Metadata propagation</td>
<td>To be able to tag a group of media streams at the same time/apply a tag to a group of media streams</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Tag expressiveness</td>
<td>To spell check tags and connect them to their synonyms and also clarify the type of tag they are</td>
</tr>
<tr>
<td></td>
<td>Tag-based ranking</td>
<td>To provide means to rank the search results based on the tags that are used to annotate the media and the strength of these tags in each media</td>
</tr>
<tr>
<td></td>
<td>Inter-referential awareness</td>
<td>To provide means for the users to refer to specific elements of the content or the associated metadata during collaborative communication with others such that this reference may be identified and understood by all parties</td>
</tr>
<tr>
<td></td>
<td>Semantic awareness</td>
<td>To provide facilities for users within the community to be informed about the kind of changes that have been made to the metadata between revisions and which elements of the metadata have been authored by which users</td>
</tr>
</tbody>
</table>

3. PRELIMINARY EXPERIMENT

Previously [38], we undertook an experiment in order to understand how users actually annotate multimedia collaboratively. The results of the experiment are used to develop MC², and the experiment design is replicated when the framework is evaluated. Therefore, this section summarizes the design and results of the previous usage experiment.

In the experiment, we asked 51 users to undertake a series of tasks using four existing multimedia metadata tools (three folksonomy tools and one MPEG-7 content-modelling tool) and their interactions were tracked. The users were chosen from a diverse population in order to produce results from typical users, similar to the ZoneTag average user approach [39]. Thus, the background of the users ranged from those who had no or very little familiarity with tagging media through to those who considered themselves experienced with folksonomy and/or MPEG-7 tools (only two of the users reported familiarity with MPEG-7). The users were unsupervised during the experiment, but were communicating with other users via an instant-messaging application, e.g. Windows Live Messenger, with conversation history enabled, to reveal important information about the behaviour of users in a collaborative community and the metadata if they are considered as comments on the content. This is similar to the approach of Yamamoto et al. [40] who tried to utilize user comments and blog entries as sources for annotations. Users were also interviewed after they completed all tasks.

The users were asked to use three folksonomy tools: YouTube for videos, Flickr for images and Delicious (in its original ‘del.icio.us’ incarnation at the time of experimentation) for Websites that contain AV content. They were also asked to use an MPEG-7 content-modelling tool, COSMOSIS [38]. Users were grouped into four common categories based on the type of content they were going to work on: personal, business, academic and recreational. Within these category groups, users worked together in smaller experiment groups to ease the logistics of all users in the group collaborating together at the same time. Thus, let $U$ be the set of all users taking part in the experiment and $|U| = 51$. Then, $G_i$ is a set of users assigned to category group $i$, where $1 \leq i \leq 4$, $12 \leq |G_i| \leq 13$, $G_i \subset U$ and $\bigcap_{i=1}^{4} G_i = \emptyset$, such that $G_i$ partitions $U$; and $H_j$ is a set of users assigned to experiment group $j$ within category group $i$, where $3 \leq |H_j| \leq 6$, $H_j \subset G_i$ and $\bigcap_{j=1}^{H_j} H_j = \emptyset$, such that $H_j$ partitions $G_i$.

Each user was required to tag and model the content of 15 images in Flickr, 10 Webpages containing AV content in Delicious and 3–5 min worth of videos in YouTube and COSMOSIS. After the users had undertaken the required tasks,
The tagging data, along with the data from the interviews and the communications transcripts, were analysed using grounded theory [41]. A grounded theory is defined as a theory which has been ‘systematically obtained through social research and is grounded in data’ [42]. The grounded theory methodology comprises of systematic techniques for the collection and analysis of data, exploring ideas and concepts that emerge through analytical writing [43]. Grounded theorists develop concepts directly from the data through its simultaneous collection and analysis [44]. The process of using this method starts with open coding that includes theoretical and constant comparison of the data, up to the point where conceptual saturation is reached (when no more codes can be assigned to the data and all the data can be categorized under at least one of the codes already available). This provides the concepts, otherwise known as codes, that will build the means to classify and initially understand the data ready for memoing. Thus, given the set of all tagging data, \( X \), the set of all interview data, \( I \), and the set of all communications transcripts, \( T \), all the data derived from the experiment can be represented by \( \mathbb{D} = X \cup I \cup T \). Open coding may therefore be considered to derive a set of codes associated with the data \( \{C_1, C_2, \ldots, C_n\} \). Each \( C \) is a pair \((c, D)\), where \( c \) is a code and \( D \subseteq \mathbb{D} \). All data are constantly compared such that conceptual saturation is reached once, or before, the full domain of comparison is considered; that is, \( X^2 – \Delta_D \cup I^2 – \Delta_I \cup T^2 – \Delta_T \cup X \times I \cup X \times T \cup I \times T \cup X \times I \times T \), where \( \Delta \) is the diagonal, and thus each data item is compared with every other data item but not itself. For example, some \( c \) reflected the key content features of AV content, which the experiment considered to be: objects (people, animals and inanimate objects); events (visual or aural occurrences within the video, with aural occurrences including music, noises and conversations); spatiotemporal locales (which position objects and events in content or media time and space, i.e. semantic location/time, precise clock date-time, geographic position or content time-point); properties of the previous features; and relationships between the previous features (temporal, spatial or semantic).

Memoing provides an additional, meaningful layer of description reflecting dimensions, properties and relationships on the codes and data to form theory. The memoing process was undertaken on three levels of analysis: individual, experiment groups and category groups, with each level including its own data as well as that of the levels below it (transcript sources are not considered at the individual level since this data is not generated from the experiments—users always communicated within groups). At the conclusion of the memoing process, a set of memos associated with the codes and data \( \{M_1, M_2, \ldots, M_n\} \) were derived. Each \( M \) is a triple \((m, \Gamma, D)\) where \( m \) is a memo, \( \Gamma \subseteq C \) and \( D \subseteq \mathbb{D} \).

In this particular experiment, we then visualized the memos using graphs and charts in order to assist with the analysis and subsequent derivation of the grounded theory which revealed a set of key implications. These are given in Table 2 and are used as a basis for designing the MC2 framework for collaborative content-modelling systems in this paper.

<table>
<thead>
<tr>
<th>Implication</th>
<th>Elaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facilitate semantic time for all features</td>
<td>Semantic time should be supported and able to be connected to all other features</td>
</tr>
<tr>
<td>Facilitate connection between co-related tags</td>
<td>User behaviour produced co-relation outcome, showing users tend to co-relate certain tags</td>
</tr>
<tr>
<td>Prioritize into metadata core</td>
<td>Certain tags should have higher priority in the metadata based on the user behaviour and preferences reported by the experiment</td>
</tr>
<tr>
<td>Inclusive and concise metadata structure</td>
<td>Comprehensive metadata is appreciated by users but its complexity is not</td>
</tr>
<tr>
<td>Facilitate property description</td>
<td>Users tend to actively associate properties with other content features</td>
</tr>
<tr>
<td>Facilitate addition of time points</td>
<td>Users tend to add time-points for content features</td>
</tr>
<tr>
<td>Enforce ease of use</td>
<td>Complexity and comprehensiveness go hand in hand and thus are related for both interface and metadata</td>
</tr>
<tr>
<td>Quick add facility</td>
<td>Users prefer to add most used tags quickly</td>
</tr>
<tr>
<td>Free-text tagging within the interface</td>
<td>Users appreciate free-text tagging</td>
</tr>
<tr>
<td>Facilitate showing all tags and authors per content</td>
<td>Need for authorship management to counter rudimentary tag-based access and anonymous tag authorship</td>
</tr>
<tr>
<td>Emphasis on asynchronous communication</td>
<td>Users found live communication unproductive</td>
</tr>
</tbody>
</table>

4. DEVELOPMENT OF THE MC² FRAMEWORK

In the previous section, we described how grounded theory was used to derive the key implications for collaborative content-modelling systems. In this section, we describe the development of a collaborative content-modelling framework for multimedia
Then we describe the functional requirements of the framework. Evident requirements were considered those challenges in order to derive the architectural components of the derived use case diagram, together with the implications and framework; and second, grounded theory was applied to this and a set of evident requirements in order to derive a use experiment from Section 3, the challenges given in Section 2 first, grounded theory was applied to the implications of the framework. Together with its various extensions, it provides the functional system requirements, implications and architectural components of the MC\textsuperscript{2} framework. In order for the framework to be fully functional; for example, there is a requirement for users to upload media so that it is available within the framework. Experiment participants had no direct input into either stage except through the preliminary experiment results. The outcomes of both stages are presented in the next two sections.

Table 3 compares the methodology for each of the two stages. Both approaches follow a similar practice, as they are both an adaptation of grounded theory to achieve a grounded truth. The data input in the two approaches share commonalities which are the challenges and implications due to the importance of both establishing the core requirements and the collaborative content-modelling framework. In both approaches, the input data will go through a cycle of theoretical and constant comparison to derive the initial, typically high-level, elements of the final diagram or framework. In the memoing stage, these are then refined and more precise elements are identified and described together with their inter-relationships (e.g. extend and include in the case of the use case diagram, process and data flows in the case of the framework). These memos are then visualized as grounded theories embodied by a use case diagram and architectural framework diagram, respectively. We now describe each in turn.

### 4.1. Research approach

Use cases were used to specify the design requirements of the collaborative content-modelling framework since they are a well-established means for capturing functional system requirements [45]. The use cases were then mapped to architectural components of the MC\textsuperscript{2} framework. In order that the functional system requirements, implications and challenges could all directly influence the framework design, we adopted a grounded theory approach throughout such that all of these influences could emerge into a well-defined and grounded specification that fully captured the design requirements. Although grounded theory is not normally used for such purposes, Glaser claims the dictum ‘all is data’ [46] and, thus, according to his approach to grounded theory, it can be applied and adapted for any research. Indeed, grounded theory has been used successfully in unconventional cases previously, such as software engineering [47] and creating process models [48]. Furthermore, the outcome of grounded theory is often one or more conceptual models. Since both a use case diagram and an architectural framework are forms of a conceptual model in essence, it is reasonable to consider them the valid outcomes of a grounded theory process.

Consequently, we adopted a two-stage process whereby: first, grounded theory was applied to the implications of the experiment from Section 3, the challenges given in Section 2 and a set of evident requirements in order to derive a use case diagram that specified the functional requirements of the framework; and second, grounded theory was applied to this derived use case diagram, together with the implications and challenges in order to derive the architectural components of the MC\textsuperscript{2} framework. Evident requirements were considered those which were not explicitly specified but were evidently necessary in order for the framework to be fully functional; for example, there is a requirement for users to upload media so that it is available within the framework. Experiment participants had no direct input into either stage except through the preliminary experiment results. The outcomes of both stages are presented in the next two sections.

Table 3 compares the methodology for each of the two stages. Both approaches follow a similar practice, as they are both an adaptation of grounded theory to achieve a grounded truth. The data input in the two approaches share commonalities which are the challenges and implications due to the importance of both establishing the core requirements and the collaborative content-modelling framework. In both approaches, the input data will go through a cycle of theoretical and constant comparison to derive the initial, typically high-level, elements of the final diagram or framework. In the memoing stage, these are then refined and more precise elements are identified and described together with their inter-relationships (e.g. extend and include in the case of the use case diagram, process and data flows in the case of the framework). These memos are then visualized as grounded theories embodied by a use case diagram and architectural framework diagram, respectively. We now describe each in turn.

### 4.2. Functional requirements

The use case diagram that was produced using the grounded theory approach described above is shown in Fig. 2. It represents the full extent of user actions when content-modelling multimedia collaboratively and the functionality needed to support them. Table 4 summarizes how the use cases map to the implications and challenges explained previously. It serves to validate the grounded theory approach used here by showing how each implication and challenge is addressed by one or more use cases (the last two rows do not map to any challenges as they were the direct result of implications only). Use cases which were the result of evident requirements only, such as manage media, are excluded from the table since they serve to enable and support other use cases.

As can be seen, the add/edit tags use case lies at the core of the requirements and caters for the tagging processes of the framework. Together with its various extensions, it provides the means for tagging the major content features that were revealed during the open coding process, such as objects and events. It is fully based on prioritizing the tagging behaviour of the users and addresses the challenges and implications emergent from...
FIGURE 2. Use case diagram portraying the functional requirements for a collaborative content-modelling framework.

<table>
<thead>
<tr>
<th>Implication</th>
<th>Target challenges</th>
<th>Supporting use cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facilitate semantic time for all features</td>
<td>Inter-referential awareness, pattern stability</td>
<td>Add/edit spatiotemporal locales</td>
</tr>
<tr>
<td>Facilitate connection between co-related tags</td>
<td>Connectedness</td>
<td>View followup tags, add/edit properties, sort tag list by relativeness</td>
</tr>
<tr>
<td>Prioritize into metadata core</td>
<td>Pattern stability</td>
<td>Add/edit objects, add/edit events, add relations, add/edit properties, add/edit spatiotemporal locales</td>
</tr>
<tr>
<td>Inclusive and concise metadata structure</td>
<td>Pattern stability, semantic awareness</td>
<td>Add/edit objects, add/edit events, add relations, add/edit properties, add/edit spatiotemporal locales</td>
</tr>
<tr>
<td>Facilitate property description</td>
<td>Connectedness</td>
<td>View followup tags, add/edit properties</td>
</tr>
<tr>
<td>Facilitate addition of time points</td>
<td>Inter-referential awareness</td>
<td>Add/edit time points</td>
</tr>
<tr>
<td>Enforce ease of use</td>
<td>Synonym control, tag expressiveness</td>
<td>Free text tagging, view predefined tags, predictive tagging, quick add, view followup tags</td>
</tr>
<tr>
<td>Facilitate showing all tags and authors per content</td>
<td>Tag-based ranking, metadata propagation, identity awareness, semantic awareness</td>
<td>Profiling, add to profile, view model, view user profile, view media, retrieve media</td>
</tr>
<tr>
<td>Emphasis on asynchronous communication</td>
<td>Identity awareness</td>
<td>Communicate, forum, messaging</td>
</tr>
<tr>
<td>Quick add facility</td>
<td></td>
<td>Quick add</td>
</tr>
<tr>
<td>Free-text tagging within the interface</td>
<td></td>
<td>Free text tagging</td>
</tr>
</tbody>
</table>

the preliminary experiment. Add/edit spatiotemporal locales facilitates both semantic time and time point features as an extension in order to provide inter-referential awareness. View followup tags serves to facilitate the connection between co-related tags by presenting the user with other tags that they might be interested in adding after a certain tag has been added, thereby improving connectedness.

Several use cases serve to enforce ease of use. Free text tagging provides users with the facility to describe content free of structural limitations, while the view predefined tags
and predictive tagging use cases form a recommendation component that checks tags, recommends new tags, suggests correct spellings and word usage, thereby enforcing synonym control and tag expressiveness. View predefined tags supports metadata propagation through tag reuse. The quick add use case enables users to add the most commonly used tags more easily and swiftly, as was revealed by the experiment.

The implications from the experiment also suggested that users exclusively prefer asynchronous communication over live chat. Consequently, the communicate use case supports messaging and forum capabilities and synchronous communication is not supported.

To facilitate the association of tags and authors on a per content basis, to propagate metadata easily, and to support identity and semantic awareness, profiling supports all major use cases by recording every action taken by the user in their profile. These actions can be retrieved through view user profile, view media and view model such that users are able to view the profiles of other users, their actions, and the actions and the tags contributed by them to the content model, while retrieve media supports tag-based ranking through ranked searching of the repository.

Finally, the manage media use case and its extensions emerged from the evident requirements and serve to populate and manage the media repository. The use of a player component within the framework is supported through the view media use case.

4.3. MC2 collaborative MPEG-7 content-modelling framework

This section presents the MC2 framework for collaborative multimedia content modelling using the MPEG-7 standard that was derived using the grounded theory approach presented in Section 4.1. The framework is given in Fig. 3 and each subsystem of the framework is now described in detail.

4.3.1. Parsing and metadata repositories

The parsing subsystem lies at the heart of the framework and is responsible for creating, parsing and updating the metadata which are all stored in MPEG-7 XML [49–52]. It encompasses five different metadata repositories catering for user details and actions, the content model, an archive for historical changes as well as user profiling, comments and notifications. Whereas frameworks commonly use non-XML databases for non-content model data such as user details and comments, MC2 uses MPEG-7 for all data storage and retrieval purposes throughout the framework, not just the content model. This single MPEG-7 schema provides the advantage of data uniformity throughout the framework.

The content model repository represents all media stream metadata. Its schema is defined as a bespoke MPEG-7 profile consisting of various description schemes (DSs), shown in Fig. 4. MPEG-7 profiles are subsets of the standard that apply
FIGURE 4. MPEG-7 profile for collaborative multimedia content-modelling communities.

to specific application areas [53] and hence the MC² profile is a subset catering specifically for the requirements established by this research. To take into account the frequency of tag usage while balancing metadata structure and complexity, user feature preferences, content feature frequency and functional availability have been balanced alongside the functional complexity of the content model. To incorporate the tags most used by users (from the add/edit tags use case and its extensions), the Semantic DS acts as a foundation for the majority of tags incorporated into the profile. AgentObject DS is used to describe people in different forms as Person DS, PersonGroup DS or Organization DS; Event DS is used for defining events; and Object DS is used to define objects. No specific DS exists within MPEG-7 for modelling animals, therefore the profile supports the use of Object DS or AgentObject DS, both of which are equally suitable, depending on the user’s viewpoints and preferences. The SemanticState DS enables modelling of people, event and object properties, while the SemanticTime and SemanticPlace DSs facilitate modelling of spatiotemporal locales for people, events and objects (semantic time may also be represented per se within the profile). All Semantic DS tags may include MediaOccurrence elements for associating start time and duration.

As with animals, there are no specific noise or music representation structures within MPEG-7; thus, noises are represented through an Event DS inside the Semantic DS, while music is represented via the Object DS. Furthermore, more advanced, music metadata may be described using the CreationInformation DS, which encapsulates the Creation DS and RelatedMaterial DS and may be linked to the Object DS using the Reference type. Conversations are another content feature not specifically supported in MPEG-7. Therefore, they are represented through the SemanticState DS within the profile, which supports time stamping through MediaOccurrence, with the text of the conversation stored as an unlimited number of AttributeValuePair elements. Conversations may be linked to AgentObject DSS using the Agent Semantic Relation so that the speaker of segments of conversation can be identified. All data are stored under the top-level type, mixedCollection, to satisfy Profiling use cases and Semantic Awareness challenges.
UserDescriptionType, which encapsulates the profile and login of the user as User from AgentType and all the user’s actions under the UsageHistory DS. The full structure is summarized in Fig. 5. UserActionList DS stores information such as action type, action time, stream name and a link to the element in XPath format which are stored, respectively, in ActionType, ActionTime, ProgramIdentifier and ActionDataItem. To support rapid facilitates adding the most used tags from the experience.

4.3.2. Tagging and recommending

The tagging subsystem results from the add/edit tags use case and its extension points and is responsible for adding and editing tags for media streams and storing them in the metadata repositories. Every tag added by this subsystem will add: an element in the Semantic DS to represent the tag in the content model repository, a UserActionList DS in the user details and actions repository to specify which user added which element in the content model, and a Collection DS in the archive repository to represent the element added in the content model and a reference to the user.

The full add/edit subsystem is responsible for comprehensively adding or editing a tag which conforms to the MPEG-7 profile of the content model. There are four processes that can be chosen based on the type of tag to be added or edited. Conversations are added in the form of a SemanticState DS, consisting of the time, duration and text of the conversation together with a reference to the speaker. Objects, people, events, times and locations are added in the form of a SemanticBase DS set to the corresponding type. Semantic, spatial or temporal relations are added via the Relation DS and have sources and targets that can be selected from tags already present in the content model. Additionally, multiple properties and time points may optionally be added or edited and associated with these tags, whereby the ID of the tag in question is stored as the source of the time point or property (property name and value). Where properties are added to existing content features, the user is required to select the source. In all cases, properties are stored in a SemanticState DS. The (optional) addition of time points enables conversations, properties, relations, objects, people, events, time and location to be associated with particular segments of the media streams, whereby SubInterval elements are added to the MPEG-7 content model containing the start point and (optionally) duration for each tag. Time points are derived from the viewing description below. Where tags are edited (updated or deleted), the user actions are described in a UserActionList in the user details and actions MPEG-7 repository and the tags are modified within the Semantic DS in the content model and replicated in a Collection DS in the Archive repository. The quick add subsystem rapidly facilitates adding the most used tags from the experiment, specifically objects, events and people. In this case, only the tag name is stored; though properties and time points may be added later when editing. Through the free-text tagging component, users may enter text for their tags freely, e.g. as they would do in YouTube. These free-text tags are stored in a SemanticState DS, as prescribed by the MPEG-7 profile.
Whenever tags are added to the content model, the recommender subsystem will retrieve the phrase and check the WordNet dictionary database for its roots, synonyms and derivatives. Then it will remove the duplicates and make a recommendation to the user from the remaining set so that more tags may be added if required, thereby broadening the range of meanings and synonyms. This subsystem maps to the predictive tagging and view predefined tags use cases described previously.

4.3.3. Versioning and viewing
The versioning subsystem caters for controlling the modifications made to the content models and corresponds to the profiling use case. When the content model is updated, the check owner process is instantiated which identifies the author of either the corresponding section of the content model for an update or the user who originally uploaded the media stream if a new tag is being added. In cases where the same user is not performing the action, then the author or uploader (whomever is appropriate) is notified through the owner notification process. Then, the apply change process makes the change effective in the system if the owner accepts the change or it will rollback the content model to its prior state if the owner rejects it.

The viewing subsystem handles stream playing functionality and is based on the view file use case. The player receives the media stream from the media stream repository and streams it to the user. Having the stream active while tagging helps improve tagging accuracy and our preliminary experiment revealed that users tend to use time points if facilities to support this are provided. Thus, the provision of a timer to handle all timing requests from the tagging subsystem serves to encourage and support this with a view to enabling more comprehensive and accurate content models.

4.3.4. Profiling and communication
The profiling and actions subsystem is derived from several use cases, namely profiling, view user profile and add to profile. It is responsible for initiating all the user actions and activities in the system. New users join the system through the create profile component which stores login information in the user details and actions repository after which the initialize user action history process will create a new (empty) action history for the user. The user area component enables reviewing of media uploaded to the system and media content modelled to date and commented upon. In this way, users are able to efficiently and effectively track their own behaviour and history within the system. The media actions component incorporates various processes that allow users to perform different actions on the media, namely uploading, retrieving, tagging and streaming media, as well as editing tags.

The communication subsystem is responsible for providing communication facilities for the users in the system. The private messaging component handles the creation and reading of private messages between users. It stores the messages in the system by storing the IDs of the sender, receiver and the message, so that they may be retrieved on request by the user. The commenting component caters for media comments, storing the user ID, stream name and comment text and handling retrieval of comments. Commenting has been ascertained as being a well established and desired functionality of any collaborative community, according to both our own experiment results and the research literature described previously.

4.3.5. Media management
This subsystem caters for the uploading and retrieval of media streams and is derived from the manage media use case. The upload media component uploads media into the system (from an originating request in the profiling and actions subsystem described above). Since there are many different media formats available, the framework converts all media into a uniform format before storage in the media streams repository. Then, a thumbnail is created for the stream in the thumbnails repository. The action of uploading is stored as a user action in the user details to record whom has uploaded this file. The media streams are linked to the metadata by storing their path in the MediaOccurrence element.

The retrieve media component enables users to search against tags in the system and to find matching media streams. The results are ranked according to the frequency of occurrence via the rank search results process. The rank top media process searches the model for the top media added or updated in the system and presents them to the user so that they can be aware of current media trends within the system. Both processes utilize

FIGURE 6. Tools and technologies used for implementation of MC2.
5. IMPLEMENTATION OF THE MC2 SERVICE

Having presented the MC² framework, we now discuss its proof of concept through the implementation of an on-line service. Figure 6 illustrates the technologies used to develop this MC² service and will be used as a reference point throughout this section. For consistency, all examples in this section are from a single video that has been deployed in the service: a 5-min clip from the Japanese manga series, *Naruto*, where Naruto combats a strong enemy in a jungle and uses his secret move, the Shadow Replications.

5.1. MPEG-7 profile

As stated previously, all events are modelled in MPEG-7 using the Event DS (through SemanticBase DS). Figure 7 presents the implementation of an event. The tag is stored under the Label and Definition, while the related media stream is indicated under the MediaOccurrence, and SubInterval represents the start time and the duration of the event. A new unique ID is created (in this case, id4) for each tag. The AgentObject DS used for people, the Object DS used for objects, the SemanticTime DS used for semantic time, the SemanticPlace DS used for locations and the SemanticState DS used for properties, all follow a similar format but with a different value for xsi:type and thus are not shown here to save repetition.

The Object DS is also used for modelling music tags and it too follows a similar pattern, however, the Creation DS and RelatedMaterial DS may also be attached to the Object DS for more detailed content modelling of music. The Creation DS includes metadata regarding the composer of the music while the RelatedMaterial DS includes more low-level audio information regarding the piece of music. Event DS is also used to model Noise tags and follows the same implementation as a regular event tag.

Figure 8 presents a sample conversation being represented in MC². As mentioned before, conversations are not specifically supported by MPEG-7, so SemanticState DS is used for this purpose. The Attribute name of the AttributeValuePair is set

---

**FIGURE 7.** Events in MC².

**FIGURE 8.** Conversations in MC².
to ‘Conversation’ and the text of the conversation is stored as the TextValue for each media occurrence. An AgentOf semantic relation connects the conversation to its speaker. When modelling conversations the Label and Property elements of the SemanticState DS are unused, which is not the case when SemanticState DS is used to model properties.

As mentioned in the previous section, every action in the system is stored in the user profile using the UserActionList.
DS. Figure 9 illustrates a sample where the user has created a new tag. The date and the time of the action are stored along with an XPath to the actual tag.

The semantic awareness challenge resulted in a need to track the changes made on tags. Figure 10 presents a sample of the archive in the MC² service where the tag was first added without time points and the time points were then added at a later date. The collection keeps the ID of the original tag but adds a versioning index to the ID of the tags it retains. Looking back at the previous figure, the UserActionList keeps a reference to the ID of the tag under the Collections DS rather than the original tag so that the service can track back which version of the tag was created/updated by each user.

5.2. Functionality

The functionality of the system was developed in Microsoft C# due to the powerful and extensive libraries and Web compatibility. The recommender works in conjunction with WordNet for database access, which is a large lexical database of English developed by Princeton University [54]. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual–semantic and lexical relations. The resulting network of meaningfully related words and concepts is widely used for spell checking and phrase recommendation by researchers.

The file conversion and thumbnail creation process are implemented using Microsoft Expression Encoder 4 [55] which will encode any audio and video into Windows Media Video (WMV) format which is the acceptable format for media playback in Silverlight (discussed further below). During the conversion process, Expression Encoder will also create a thumbnail as long as it is set in the JobFile that has been created for the process. Another strong capability of Expression Encoder is that it can convert audio files into videos with blank

FIGURE 11. An XML JobFile template.

```xml
<?xml version="1.0" encoding="utf-16"?>
<JobFile Version="1.0">
    <Job HtmlTemplate="App:\Templates\en\Pilot\Pilot" OutputDirectory="#outputDir#" SaveJobFile="True" AppendJobID="False" Log="On">
        <Mediafiles>
            <MediaFile FileType="Audio, Video" ThumbnailMode="FirstFrame" ThumbnailSize="160, 120" MarkerThumbnailSize="36, 28"></MediaFile>
        </Mediafiles>
    </Job>
</JobFile>
```

FIGURE 12. Adding an event in C# using LINQ.

```csharp
public string AddNewEvent(string name, string fn, string tp, string dur)
{
    if (!IsTaken(name, "EventType"))
    {
        string idd = GetID();
        var query = from c in MP7.Descendants
                     where (string)c.Attribute
                           ("{http://www.w3.org/2001/XMLSchema-instance}type") == "EventType"
                     select c;
        XElement temp = new XElement(query.First());
        temp.Descendants("urn:mpeg:mpeg7:schema:2001\Name")
                     .First().Value = name;
        temp.Descendants("urn:mpeg:mpeg7:schema:2001\MediaDuration")
                     .First().Value = tp;
        temp.Descendants("urn:mpeg:mpeg7:schema:2001\MediaDuration")
                     .First().Value = dur;
        temp.Attribute("id").Value = idd;
        query.First().AddAfterSelf(temp);
        MP7.Save(XMILFileName);
        arcx.CreateArchive(temp);
        return idd;
    }
    else
    {
        return "-1";
    }
}
```
or preset visual elements, therefore enabling users to utilize the same Web player for playing videos and audios. In order to use Expression Encoder, the encoder class will invoke ‘Encoder.exe’ as a new hidden process and add a JobFile as input argument for it. JobFile is an XML file that can be customized with specifications of the conversion process. Figure 11 presents the sample JobFile template.

The main functionality of the MC2 service is the tagging capabilities. As mentioned before, all the information of this system is stored in MPEG-7 which is an XML-based tool. Therefore, any information storage or retrieval action of the service relies on the XML-based metadata. The parsing algorithm implemented in this system for storing and retrieving data was developed using the Microsoft LINQ (Language-Integrated Query) library [56]. It offers developers a way to query data using strongly typed queries and strongly typed results, therefore making the parsing operation faster and more efficient. There are three classes responsible for metadata parsing, creating and editing: MP7Manage, UserAccount and ArchiverCollections. The MP7Manage class deals with adding, retrieving and editing tags using LINQ. This class creates a new XElement that is a clone of the template tag, setting its attributes and adding/replacing it in the MPEG-7 content model. Figure 12 presents sample code for adding events. The UserAccount class is responsible for adding actions. The action method of this class is called by the first class at the end of add or edit methods and Fig. 13 presents sample code of an action being added. The ArchiverCollections class caters for dealing with archiving and is called at the end of nearly every method of the first class to store the changes (Fig. 14). As the MC2 service generates and operates on large collections, the search results, tag previews and user area results all need to be stored in collections. The dynamic arrays provided by C# create a suitable medium for storing and operating on large collections of dynamic sizes.

5.3. Interface

The Web interface of the service was developed in ASP.NET and stylized using cascading style sheets (CSS) to ensure maximum platform compatibility, functionality

```
public static void AddAction
    (string sPath, string acttype, string userid, string fn, string elemid)
{
    ServerPath = sPath;
    string XmlFileName = ServerPath + GetUserFilePath(userid);
    XElement MP7 = XElement.Load(XmlFileName);
    var query = from c in MP7.Descendants
        c
    XElement temp = new XElement(query.First());
    temp.Descendants("urn:mpeg:mpg7:schema:2001:ActionType"
    temp.Descendants
        temp.Descendants
        ("urn:mpeg:mpg7:schema:2001:ProgramIdentifier"
        .FirstChild(), Value = fn;
    temp.Descendants
        ("urn:mpeg:mpg7:schema:2001:ActionDataItem"
        .FirstChild().Attribute("href"), Value = fn.Replace(".wmv", ".mp7.xml") + elemid;
    query.First().Parent.FirstNode.AddAfterSelf(temp);
    MP7.Save(XmlFileName);
    if ((acttype != "Create") &&
    if (IsCreator(ServerPath, userid, elemid, fn))
    {
        string crtr = FindCreator(ServerPath, elemid, fn);
    if (crtr != ")
    {
        SetNotification(ServerPath, fn, elemid,
        crtr, GetUser_Name(userid, ServerPath), acttype);
    }
    }
}
```

FIGURE 13. Adding a user action in C# using LINQ.
public void AddChange(XElement xe10)
{
    XElement xe1 = new XElement(xe10);
    var query = from c in Archi.Descendants()
        where ((c.Name.LocalName == "Collection") &&
        (c.Attribute("id").Value == xe1.Attribute("id").Value))
    select c;
    XElement temp = new XElement(query.First());
    XElement temp2 = query.First();
    temp.Attribute("id").Value = xe1.Attribute("id").Value;
    temp.Descendants("urn:mpeg: schema:2001:ContentRef"
        ).FirstOrDefault().Attribute("href").Value = RefUserID;
    int id = int.Parse(query.Last().Descendants().
        First().Attribute("id").Value.
        Replace(xe1.Attribute("id").Value + "","") + 1;
    xe1.Attribute("id").Value = xe1.Attribute("id").Value + "." + id.ToString();
    temp.Descendants().FirstOrDefault().ReplaceWith(xe1);
    temp2.Parent.LastNode.AddAfterSelf(temp);
    Archi.Save(XMLFileName);
}

UserAccount.AddAction(
    XMLFileName, Remove(
        XMLFileName, IndexOf("Archive\"), "Update",
        RefUserID.Replace("userlist.mpeg.xml",""),
        XMLFileName, Remove(0, XMLFileName, LastIndexOf("\") + 1).
            Replace(".mpeg", ".wmv"), xe1.Attribute("id").Value);
    )

FIGURE 14. Storing changes in the model in C# using LINQ.

and presentation. ASP.NET is a powerful Web developing tool that is compatible with most browsers and is widely used within the Web development community.

Figure 15 presents the Add Tag page in the system. As can be seen the page is divided into two areas: the media player and the tagging section. To ensure the interactivity of the interface and to facilitate its use while making the pages ‘light’ enough to be used easily by users, AJAX has been used where necessary to ensure a sound presentation of the interface and ease of use. In order to ensure that add/edit actions would not affect the playing back of media, the player and the add/edit panels are in two separate Update Panels.

In the tagging section, users choose the type of tag they want to add from a list, either fully committing themselves to adding it with all the options available or via the quick add option to quickly add objects, events or people. Figure 16 shows an example of the Edit tag page’s add panel.

To enable the Webpages to display comprehensive information and support comprehensive functionality without unnecessary redirections, and also to ensure the media stream is not stopped during user interactivity, Web controls have been used to portray different stages and options of tagging and tag editing. For each process within the add/edit components of the MC2 framework, a separate Web control is created. This enables quick interactive responses and ensures that each type of add/edit is performed independently. Figure 17 shows the available tagging options in the system. Selecting a tag in the list will dynamically unload the selection Web control and load a new Web control that is responsible for adding the selected tag instead.

The Web-based player of the service was developed using Silverlight [57]. With its XML-based format and non-complex functionality, it provides a light Web-player suitable for online streaming of media that is compatible with all major browsers. To interact with the player and extract information such as media time from the player, JavaScript functions access the player and extract the required information.

On the home page, the users have the option to upload new media or search for the existing media using keywords which will present them with ranked results based on the presence of these keywords within the content model. Also, the users are presented with the top 3 most recently updated media (recently uploaded or recently modified). As mentioned previously, both search results and most recent media are found based on their models stored in MPEG-7. The search class will parse the MPEG-7 of the user actions using LINQ and will select the most recent actions and media related to them. As LINQ allows querying of XML files, the system will look for distinct new media. The search class also looks for keywords in the system and will rank each result in the set based on the times the keyword has been mentioned in the model or the number of relationships to that keyword.
6. VALIDATION

The previous sections described the proposed MC\textsuperscript{2} framework and online service. In order to validate MC\textsuperscript{2}, we undertook an experiment with the same users who participated in our preliminary experiment and compared the results. In addition, we evaluated the stability, plasticity and accuracy of MC\textsuperscript{2} by considering its functionality against that of folksonomy and MPEG-7 tools in terms of whether they fully or partially address the challenges identified earlier.

6.1. Empirical validation

An experiment was undertaken to evaluate the MC\textsuperscript{2} online service and compare the results with the results of the experiment summarized in Section 3. The experiment design was kept similar except that only media streams were used, since they incorporate both video and audio and each frame can be considered as a still image. Using the MC\textsuperscript{2} online
service, the same 51 users from the preliminary experiment were asked to upload and tag a 3–5 min video (similar to the task they undertook with COSMOSIS, except that they were asked to try and model the frames as still images and the audio of the movie as music if it was suitable in their opinion). The users were also required to search for other videos on the system and add new tags or update existing tags in them. They were unsupervised, but communicated with other users via the forum and commenting features of the system. Users were interviewed after they completed all tasks. Each interview was semi-structured. Grounded theory was then used to analyse the experiment data and the same processes of open coding and memoing were carried out.

MC\textsuperscript{2} seeks to balance the positive aspects of both folksonomy and MPEG-7 tools while seeking to overcome their negative aspects. As discussed previously, the folksonomy tools provide a collaborative environment with easy to use interfaces that enable users to create metadata free of the limitations imposed by structured metadata; however, this freedom results in confusion, difficulties in searching, semantic ambiguity and identity awareness issues. Conversely, while MPEG-7 tools are very well structured and therefore produce well-formed metadata with precise semantics, which facilitates searching and browsing and raises identity awareness, this structure makes these tools harder to use and creates limitations for the users which often leads to less-detailed metadata and complex interfaces.

Figure 18 compares the tag usage in MC\textsuperscript{2}, the folksonomy tools that support video (YouTube and Delicious) and the MPEG-7 tool (COSMOSIS) as the average use of each tag per user in the respective tools (average across all tasks). As the results in the folksonomy tools were very similar, the mean is shown.

These results show an overall increase in tag usage in MC\textsuperscript{2}. This can be explained by the fact that MC\textsuperscript{2} sought to simplify the tagging process while utilizing the MPEG-7 structure which has
a positive impact on the user experience resulting in an increase in tagging.

Events, objects and people, which were the most used tags in both folksonomy tools and the MPEG-7 tool, are still among the most used tags in MC², while time and semantic relations have been used a considerable number of times, which was highlighted by their use in the MPEG-7 tools. The time tag has seen a great increase in use compared with the MPEG-7 tool. This can be explained by the fact that time has been prioritized in MC², resulting in better support for this tag in the content model and the interface. This shows that this prioritization in the framework was met positively by users.

As highlighted in the functional requirements in Section 4.2, the framework assigned a higher priority to properties, including higher priority in the content model as well as facilitating the use of it in the interface. This has been positively received by the users, resulting in an increase in the use of property tags.

The results indicate a slight drop in the use of spatial relations in comparison to the MPEG-7 tool but a slight increase over the folksonomy tools. As the use of other tags has been simplified and spatial relations were assigned a low priority, this may be explained by the fact that users have tended to use other tags more often and therefore the use of spatial relations has decreased. Further research and development would investigate how to counter this.

Figure 19 shows the users’ views towards the tools used in these experiments. Ninety-six per cent of the users (49) voted COSMOSIS as the hardest tool to use and none considered MC² the hardest tool to use. This suggests that even though both these systems are based on MPEG-7, MC² is significantly easier to use. At the same time, 90% of the users (46) found MC² the most functional tool, while 5 users voted for COSMOSIS, and none voted for any of the folksonomy tools. This suggests that MC² has successfully harnessed the power of MPEG-7 while minimizing its complexity such that users not only do not find it hard to use but consider it more functional than COSMOSIS.

The folksonomy tools were together considered the least functional tools by all users. All 51 users claimed that all folksonomy tools used in the experiments were equally low functioning. Thirty-one users (60%) found MC² to be the easiest tool to use while the rest of the votes went to YouTube (20) and Flickr (1). This majority support for MC² suggests that it has managed to provide a comfortable tagging environment for users without sacrificing functionality.

![Figure 18](image1.png)  
**Figure 18.** Tag usage in MC², folksonomy tools and the MPEG-7 tool.

![Figure 19](image2.png)  
**Figure 19.** Views on tagging systems.
TABLE 5. Comparing MC² with existing tools.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>MC²</th>
<th>Folksonomy tools</th>
<th>COSMOSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>Pattern stability</td>
<td>F</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Identity awareness</td>
<td>F</td>
<td>-</td>
</tr>
<tr>
<td>Plasticity</td>
<td>Synonym control</td>
<td>F</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Connectedness</td>
<td>F</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Metadata propagation</td>
<td>F</td>
<td>-</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Tag expressiveness</td>
<td>F</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Tag-based ranking</td>
<td>F</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Inter-referential awareness</td>
<td>F</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Semantic awareness</td>
<td>F</td>
<td>-</td>
</tr>
</tbody>
</table>

F, full support; P, partial support; -, no support.

The users in the previous experiment were asked to use live communication through instant messengers while tagging. All the users (51/51) found this distracting and unnecessary. In MC², users were asked to use the messaging and commenting capabilities of the service, and all users were unanimously satisfied with this means of communication.

6.2. Theoretical validation

The framework and online service proposed in this paper has aimed to meet the challenges identified in Section 2. Therefore, we considered these challenges as criteria by which to evaluate MC² and compare these results with those of the folksonomy and MPEG-7 tools used in the preliminary experiment. Table 5 summarizes the results of this comparison.

6.2.1. Model stability

The challenges in this area concerned the stability of the model against change. As opposed to existing folksonomy tools, the MPEG-7 profile used in this framework is a structured model with an extensible schema. It can be considered to have a stable pattern, even in light of new tags since the addition of new tags can be incorporated without modification to the schema of the existing tags. The same cannot be said for existing folksonomy tools as they are unstructured, without any well-defined schema. COSMOSIS also uses MPEG-7 but may be considered to have only partial pattern stability since it does not readily support the definition of new tags, e.g., there is no support for adding conversations, even through the use of the existing content features it does support.

MC² proposes an extensive profiling system that logs every action taken by the user. Consequently, it is very clear which tags have been created and edited by which users. This comprehensively increases identity awareness for users. In the folksonomy tools, such as YouTube and Flickr, it is not possible for other users to contribute tags and the user is never made aware of any modifications to the tags he himself has made. Another problem with these systems is that the tags are not defined separately and consequently the process of adding and editing tags is completely hidden to the user. COSMOSIS in itself is not a collaborative system and therefore there is only one author for each model.

6.2.2. Model plasticity

Model plasticity concerns the degree by which the model stabilizes after a finite number of metadata additions so that no further model elements are required in order to achieve full closure of the semantic gap by the MPEG-7-based content model. The framework proposed here ensures that once all the DSs in the scheme are used in the model, the user will be able to tag anything that is in the media file without having to use a new DS. This is reflected by synonym control, connectedness and metadata propagation. The existing folksonomy systems and COSMOSIS do not provide any means for synonym control. Users are free to tag as they like. MC² uses WordNet to control the tags that are entering the system and suggest the synonyms to the user so that they can be added as well, addressing the synonym control challenge as well as providing a comprehensive model. In the same way, WordNet also suggests connected terms that relate to the tag added by the user, which enables the user to create a more comprehensive and connected model. Also, as opposed to folksonomic tools, MC² allows the user to define their own relationships between the tags they have defined which provides an effective means for addressing the connectedness challenge. COSMOSIS partially addresses the connectedness challenge as it allows the definition of relations between certain tags. MC² also provides the means to tag sets of media streams as opposed to carrying this out sequentially, a facility not provided in existing systems; therefore, the user can propagate metadata through to a group of media streams quite readily, as opposed to just adding them to each one separately. COSMOSIS also addresses metadata propagation fully since a tag can be propagated to all frames in a video and used tags may be assigned to new videos.

6.2.3. Model accuracy

Challenges here focus on how accurately the model relates to real-world semantics. MC² ensures a high level of accuracy in the model by using WordNet and tag suggestions. The WordNet dictionary is used to identify and notify the user of misspellings. Furthermore, unlike the unstructured text of many existing tools, the structured tagging of MC² aims to clarify the tags through the MPEG-7 structure, which clarifies what type of tag each keyword stands for (such as object, concept, time). It will also increase the expressiveness of the model by defining the relationships between the tags. This structured tagging also serves to increase the accuracy of the search results as the results may be ranked according to the usage within the content models. Some folksonomy tools (Delicious) have clouds that represent further relationships between tags, but these relationships are defined by the users and not by the system.

Table 5 above illustrates the level of support the tools have for various aspects of content modeling.

F, full support; P, partial support; -, no support.
the demographics of the tags in media but this information is not used in ranking the media and searches, therefore these clouds cannot be considered to meet this challenge. A major aspect of the framework is satisfying inter-referential awareness. The framework allows users to define the time-points of the tags they are modelling, therefore enabling other users to pinpoint the exact moment in the video that the tags apply. Folksonomic tools do not provide such facility and therefore it is complicated and difficult to focus the attention of all users to certain tags at the required time during the video. COSMOSIS and MC\textsuperscript{2} both allow the users to add time points to the tags in the form of starting point and duration which helps to focus the attention to a specific time during a media stream. YouTube allows visual commenting on the videos with embedded time points but this is not the same as inter-referential awareness as they are not related to any tags. MC\textsuperscript{2} also provides a strong semantic awareness through its versioning component which not only keeps track of all the actions users perform, but also notifies the original creators of videos and tags of new tags being added or tags which they had created which have been amended or removed.

7. CONCLUSIONS

This paper has presented MC\textsuperscript{2}, a framework and online service for MPEG-7 content-modelling communities which is based on the challenges to collaborative multimedia content modelling reported in the research literature and prior results of an experiment undertaken to investigate user behaviour in collaborative content modelling. The results of validating MC\textsuperscript{2} with a user experiment and against the challenges were discussed. MC\textsuperscript{2} was shown to be an improvement over existing content-modelling tools, both showing a marked increase in tag usage over comparable tools while meeting all the challenges of collaborative-modelling approaches that were identified in the research literature. We now consider the various impacts and future directions of the research.

7.1. Impacts

This research has impacts wherever multimedia are used extensively and consequently it informs related research and development in broad domains such as metadata, ontologies, collaboration, personalization and ubiquitous computing. MC\textsuperscript{2} enables effective techniques, tools and concepts by which multimedia systems that rely greatly on metadata could adopt a collaborative community-driven approach to the creation and maintenance of that metadata, thereby avoiding the limited semantics of automated techniques and benefiting from richer, more detailed metadata. Experiment results have provided an understanding of how to effectively achieve collaborative multimedia tagging within content-modelling communities and through innovation of a collaborative community-driven online service that combines the strengths of both folksonomy-based and content-modelling based tools while minimizing their weaknesses, the research can be seen to contribute a means by which existing multimedia systems may be optimized as well as a means for innovative design of new multimedia systems. Significant understanding was gained regarding which aspects of the MPEG-7 MDS should be core for everyday users as well as discovering limitations of the MPEG-7 standard, such as the lack of specific DSs to cater for some content features that were repeatedly identified by users during the experiments. Consequently, the MPEG-7 profile that lies at the heart of the MC\textsuperscript{2} framework can be taken as a basis for implementation in any community-driven content-modelling system. The MC\textsuperscript{2} community-driven content modelling service stands as an example implementation of how the MPEG-7 profile can be comprehensively exploited, while the architectural framework of the MC\textsuperscript{2} service may be used as a benchmark for informing and advancing future systems research and development.

For industry professionals, notably system developers, MC\textsuperscript{2} has an impact on how collaborative multimedia systems should be designed and implemented. In application domains where search and query of multimedia is important, the architectural framework of the MC\textsuperscript{2} service, the MPEG-7 profile that it utilizes, and the associated techniques, tools and concepts, together inform the implementation of relevant systems. In this way, such application domains can take advantage of the power of online communities to improve the breadth, depth, accuracy and perspective of their multimedia metadata using the MC\textsuperscript{2} approach.

For end users, the use of an appropriately refined MPEG-7 profile at the heart of the service reduces the complexity and redundancy of MPEG-7 content models and thus makes them more accessible to everyday, non-expert users. In addition, user experiment data uncovered clear priorities by users in the description of content features and relationships between them and, through the exploitation of these results in MC\textsuperscript{2}, users are able to better work with metadata, both individually and collaboratively, making the community approach of the MC\textsuperscript{2} service more effective than existing tools for such users.

7.2. Future directions

There are opportunities for a number of future research and development directions to further build on the research presented in this paper. Some tags, such as spatial relations, were seen to have lower use in MC\textsuperscript{2} than the compared MPEG-7 tool and therefore future research will explore how to further improve the user experience with these tags. Another research and development thread will explore the application of the MC\textsuperscript{2} approach in less-conventional domains, such as collaborative gaming, and investigate whether users exhibit different priorities in the description of content features and relationships within these domains. Evidence from this investigation would be used to modify the MPEG-7 profile and
also the framework in order to cater directly for these users. Finally, there is a need for research into various community characteristics, such as quora in relation to media and searching requirements.

FUNDING

This research was supported by the UK Engineering and Physical Sciences Research Council (EPSRC), grant no. EP/E034578/1.

REFERENCES


