Interaction Issues in Computer Aided Semantic Annotation of Multimedia

Chris Bowers, Will Byrne, Jonathan Melhuish, Peter Lonsdale, Chris Creed,
Charlie Pinder, Russell Beale, and Robert Hendley

School of Computer Science
University of Birmingham
B15 2TT, UK
C.P.Bowers@cs.bham.ac.uk

Abstract. The CASAM project aims to provide a tool for more efficient and effective annotation of multimedia documents through collaboration between a user and a system performing an automated analysis of the media content. A critical part of the project is to develop a user interface which best supports both the user and the system through optimal human-computer interaction. In this paper we discuss the work undertaken, the proposed user interface and underlying interaction issues which drove its development.

1 Introduction

Online video capture, editing and storage are becoming increasingly popular and have resulted in a number of large, searchable video archives. Annotation of this multimedia content is critically important for effective archival and retrieval of these documents. Unlike text documents, multimedia material poses a problem for search engines because they cannot extract any information about the semantics of the content. Although there have been advances in the automatic analysis of images and video, it is still the case that producing a semantically meaningful description of multimedia material requires the skill of a human annotator.

The premise of the CASAM (Computer Aided Semantic Annotation of Multimedia) project is that by building an environment for *collaborative* human and machine analysis and annotation of multimedia documents we can produce better annotations more efficiently. A user of CASAM should be able to improve upon the time taken to annotate, the quality of annotation and/or the quantity of annotation when compared to working with other systems.

The model implemented in CASAM is of the user and the automated components working co-operatively and asynchronously towards the production of an annotation:

- The annotations provided by the user direct the workings of the autonomous systems.
- The user can see the provisional results from the automated components and can confirm or reject these results.

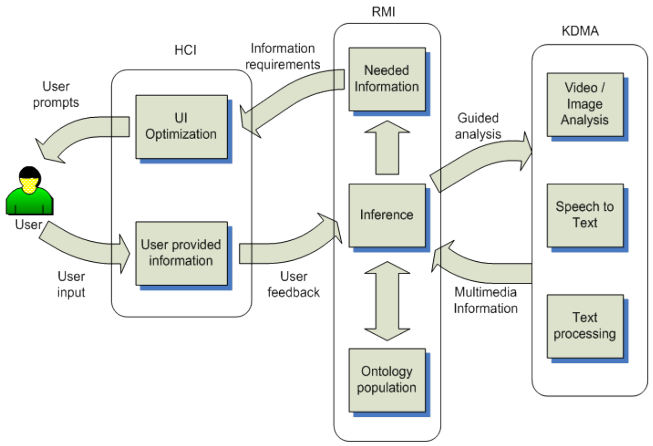


Fig. 1. CASAM interaction loop

- The user is presented with explicit requests for information from the system and can respond to these.

The CASAM system consists of three primary components. The *Knowledge-Driven Multimedia Analysis (KDMA)* component extracts low-level information from the multimedia content. The *Reasoning for Multimedia Interpretation (RMI)* component infers new higher level concepts from the results of the KDMA component and attempts to resolve ambiguities and the *Human Computer Interaction (HCI)* component provides an intelligent interface which attempts to optimise user interaction and successfully direct the annotation process. An outline of the interaction between these components is provided in figure 1.

In order to better showcase the CASAM system and benefit from a high level of end user feedback, the domain for the CASAM project is constrained to news production in news agencies and broadcasters. In this paper we discuss the CASAM HCI component and the associated interaction issues that arise as a result of the approach taken by the CASAM system.

2 Background

Initial work [4] focused on understanding the unique user annotation requirements for this domain and the limitations of the current annotation workflows in place. This was achieved primarily through discussions and focus groups with each of our end-user partners (Agencia de Noticias de Portugal (Lusa), Deutsche Welle (DW) and the European Journalism Centre (EJC)) and also with the British Broadcasting Corporation (BBC). Here we discuss some of the outcomes.

Two primary user groups are typically involved in the annotation process in a media organisation. A journalist in the field needs to be able to annotate quickly, yet provide sufficient information for the piece to be picked up by editors. Archivists, however, are annotating the content for long-term storage and retrieval purposes and therefore need to be more thorough and detailed, thus taking more time.

Multimedia documents are typically annotated at the global/top level, or as completed programmes. Metadata standards such as IPTC (International Press Telecommunications Council) are used to indicate the genre/subject of a programme at the top level, but this is not available at shot level. The subject keyword vocabulary is derived from IPTC, but this is restrictive and so often found to be inadequate.

More sophisticated annotation tools apply annotation at the scene or shot level where more detailed annotations can be provided. Individual pieces within a single document can be identified as separate assets. These systems may perform automatic scene-break analysis to help the user work through the piece. Annotation is typically performed by selecting in and out points. Although a restrictive vocabulary is often provided, the majority of annotators seem to prefer to use free-text fields to include summary descriptions as well as specifying shot types etc. However, with no fixed or prescriptive vocabulary for how information is entered into the content description field, this can be problematic. Often annotators are provided with guidelines to ensure their annotations have maximum impact on retrieval via search results. This may be as simple as specifying what to include in the document title.

After entering annotations which relate to a specific location within a multimedia document these locations may be modified to change the scope of the annotations. This is sometimes necessary if, for example, it becomes apparent that a shot is part of a group of similar shots. Previous annotation entries can be re-used but this is usually provided through some purpose-built mechanism (drop-down menu option). However users indicated a preference for a more familiar copy and paste tool, which is commonplace in document editing interfaces.

In organisations where media is stored offline it is much more difficult to annotate at the shot level because it is more difficult to associate some annotation to some moment within the video when it is a physical object, so most content is annotated at the global level. It is common for the density of annotation to vary greatly within an archive. This can depend on the particular staff and their work load, the content of the material and the format in which it was captured. A commonly raised issue was the difficulty in motivating staff to thoroughly annotate because it was generally perceived to be a large overhead requiring significant time or effort with little obvious benefit to the annotator.

3 Interaction issues in multimedia annotation

The user of CASAM will, typically, have a very limited amount of time available, in some use cases a small number of minutes. The aim of the system must be to extract the maximum value from this available resource. This means that the HCI component must reason over the information presented to the user and try to optimise the interaction. Whilst the information provided to the HCI component has measures of confidence and importance, this is not on its own sufficient to determine what to present and when to present it to the user.

The interaction with the user also has a cost in terms of time and cognitive load. This cost is dependent upon, amongst other things, the difficulty that the user has in performing the task and the current context of the users task. The requirements of the HCI component are therefore twofold:

- Provide an effective user interface onto the CASAM system.
- Manage the interaction with the user in order to maximise the information gain for the system and minimise the cost to the user, both in terms of time spent and cognitive load.

4 The CASAM HCI component

The CASAM user interface is split into two key implementation components. The CASAM HCI backend handles communication with the rest of the CASAM systems and parses and processes the incoming information ready for display. The CASAM HCI backend can be notionally thought of as deciding *what* to display. The display components produced by the backend are then passed to the CASAM HCI frontend, which implements the actual user interface and handles user interaction. This frontend component decides *where* and *when* to display these elements. This division into two components ensures the CASAM HCI component is always available and responsive to both the user and the rest of the CASAM system. An overview of the CASAM HCI component, including the internal architecture of the frontend and backend components is provided in figure 2.

4.1 The CASAM communication architecture

The CASAM system is structured according to the cloud computing paradigm. Several distributed components form the CASAM system but this is all abstracted from the user behind a single interface. In order to best support scalability and robustness these resources are built as web services with a predefined communication architecture. The architecture consists of:

WSDL (Web Service Definition Language) documents describe the operations that the web services must provide or consume. These documents act as contracts between the various components. The orchestration of communication within the CASAM system is defined by a BPEL (Business Process Execution Language) document which is enforced by a CASAM Orchestrator component. All communication within the CASAM system is via this orchestrator.

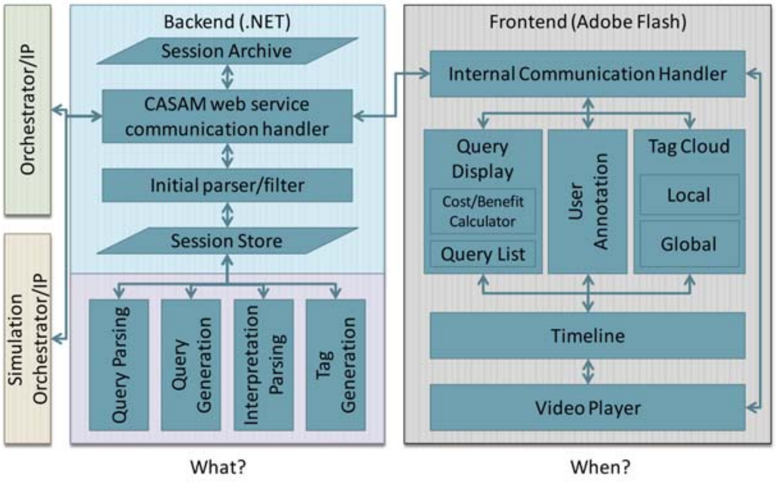


Fig. 2. Overview of CASAM HCI component architecture

Description Logic is the form taken by the messages passed between the various CASAM web services. These messages describe the logical structure of the semantic annotation utilising a set of ontologies. Each element of the semantic annotation comes in the form of three distinct elements: the *subject* of the argument, the *object* being referred to and the *relation* between the subject and the object. In the context of the CASAM systems these are referred to as an *assertion* if generated by automated analysis of the multimedia content (KDMA) or an *interpretation* if generated by reasoning over an ontology (RMI).

4.2 CASAM HCI Backend

The backend handles communication both in and out of the CASAM HCI component. When the CASAM system generates some new annotation or a query this is communicated to the backend which then parses the content of this fresh annotation. The resultant assertions/interpretations are then processed by four key components which act asynchronously (see figure 2):

Interpretation/Assertion Processing Interpretations and assertions are received from both the RMI and KDMA components respectively. In order to construct some human readable concept to display in the user interface a number of interpretations/assertions are required, each providing a piece of the required information. Table 1 describes an example set of interpretations/assertions required in order to generate a human readable concept for the user interface (UI). In this case 8 assertions/interpretations are required to determine that a person called "John" exists within the video between 10s and 22s.

The objective of the interpretation/assertion processing component is to process through all received interpretations/assertions and, where possible, generate

Subject	Relation	Object
IND-1	RDF - type	EDO - Person
IND-1	MCO - hasConcreteValue	"John"
VideoSegment-1	RDF - type	MCO - VideoSegment
VideoLocator-1	RDF - type	MCO - VideoLocator
VideoSegment-1	MCO - hasInterpretation	IND-1
VideoSegment-1	MCO - hasSegmentLocator	VideoLocator-1
VideoLocator-1	MCO - hasStart	"10 seconds"
VideoLocator-1	MCO - hasEnd	"22 seconds"

Table 1. A set of interpretations forming a displayable entity

visual representations which encapsulate what to display (in the form of a string) and when to display it (in the form of a set of start and end times).

Query Processing This process extracts queries received as a result of an ambiguity raised by the RMI component. The main task for this process is to produce a human readable version of the set of propositions generated by the RMI component.

First it parses the query to check that it is in the correct form of a list of possible assertions. Next, the subject of the query is extracted. If the query contains assertions with subjects that relate to a tangible entity such as a specific person or physical object then it can be considered a concrete query and the query can be phrased around that concrete subject, for example "What is the profession of John?" If a query contains assertions that cannot be associated with a tangible entity then the query is assumed to be an abstract query and so a generic statement is used, such as "Select all that apply".

In the same manner as the interpretation/assertion processing, query processing results in a UI data object which encapsulates the logical and human readable forms of the query as well as time points in the multimedia document to which it relates.

Query Generation A significant proportion of the assertions generated by the KDMA or RMI components may not be easily related to the video content directly. This is especially true for assertions generated from auxiliary documents such as existing text description of the content of the video. The purpose of the query generation process is to attempt to link these assertions with concepts that are already associated with the video by either position or time. This is achieved by identifying situations whereby some valid *relation* may be proposed between an entity that is associated with a video segment and one that is not. If one or more syntactically valid assertions can be formed (subject-relation-object) using description logic then a query can be raised.

Here we provide a specific example. We assume that the KDMA component has identified a person's face in the video along with the time at which that face appears within the video. We also assume that a number of names have been identified by the KDMA component, for example "John", "Paul", "George" and "Ringo". However, because these names were derived from auxiliary text there is no way to associate them with any particular time within the video.

In this case the query generation process would identify these names and the face as valid subject and object components of a relation between a person's face and a name. The following query would be generated with time points set to that of the location of the video segment in which the face appears:

```
What is the name of this person?
       John
       Paul
       George
       Ringo
```

Tag Generation The tag generation component uses the current state of annotation to gather domain knowledge about a given video segment and bootstrap the existing interpretations for that video segment with new concepts, in the form of keywords. These keywords are generated outside of the framework of the CASAM vocabulary and without the need for formal reasoning across the ontologies. However, this does allow the user to assert concepts that are outside the existing ontologies. In this case, the concept can be sent as free text to the KDMA component which can attempt to match it to something within the existing ontology or a machine learning component can attempt to introduce the new concept to the ontology directly.

These new concepts are generated by finding terms most likely to be associated with the ones provided by CASAM, using a TF-IDF (Term Frequency - Inverse Document Frequency) approach. Given a set of current annotations, a set of relevant Wikipedia article titles is generated, D. Given each term, t_i, for each document, $d_j \in D$ the term frequency is defined as

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

where $n_{i,j}$ is the number of times term t_i occurs in document d_j. The inverse document frequency for document d is calculated for each term t_i using

$$idf_i = log\frac{|D|}{|d : t_i \in d|} \tag{2}$$

where $|d : t_i \in d|$ represent the number of documents within which the term t_i occurs. The TF-IDF value for term t_i in document d_j is then calculated as

$$tfidf_{i,j} = tf_{i,j} \times idf_i \tag{3}$$

Resulting terms are filtered for 'stop words' (words which are either semantically insignificant or very common, such as articles or prepositions) to avoid trivial terms. A subset consisting of terms with the highest TF-IDF value are used to generate concepts to suggest to the user. For example, the terms

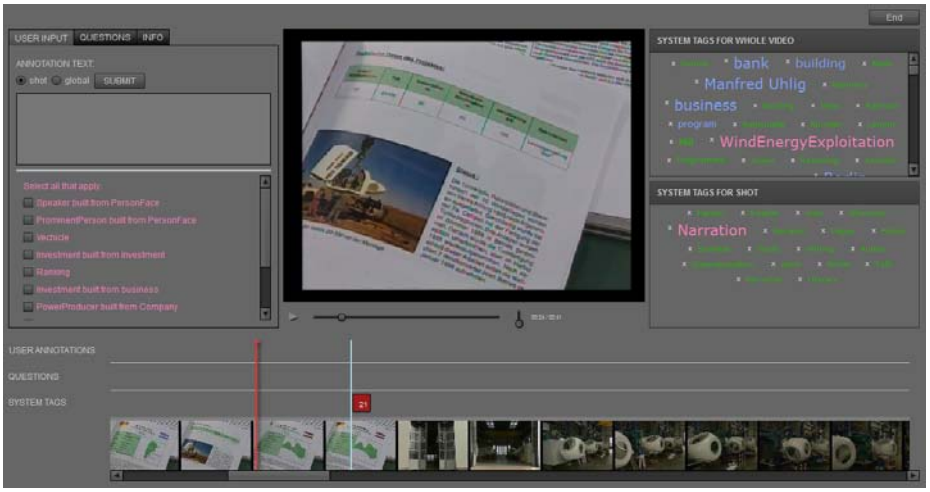


Fig. 3. Screenshot of CASAM user interface

```
wind turbine, noise, ugly
```

result in the following highest TF-IDF valued terms:

```
wind, blade, power, sound, tower, windmill, axis, electric
```

4.3 CASAM HCI Frontend

Figure 3 depicts an instance of the CASAM user interface. It consists of four primary components: the user input canvas, the annotation canvas, the video player and the timeline.

Careful thought had to be given to the layout and structure of the different visual components making up the user interface. For example, what size should each component be? Which ones should be larger and take up more screen space? What should be the focal point of the interface? In commercial and research applications that focus on multimedia annotation, the video that is being annotated is normally given precedence in the interface [8, 10, 16]. This is because all of the other components are reliant on the contents of the video and often its current playhead time (e.g. dynamic adaptation of the tag cloud depends on what is currently being displayed in the video).

As the user will mostly be making annotations whilst watching the video, the annotation canvas and user input components (the two primary ways for the user to add/modify annotation) were placed at either side of the video. This allows the user to easily switch between adding free text, tags, and navigating the video. The timeline and annotation canvas area were coupled together as they are likely to be used in tandem. That is, the user will look for annotations that have previously been made (via the timeline) and will then want to review

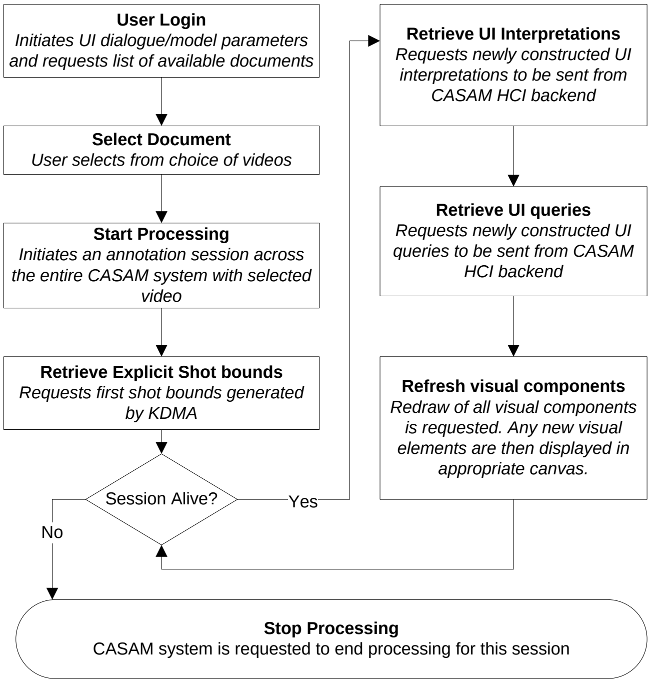
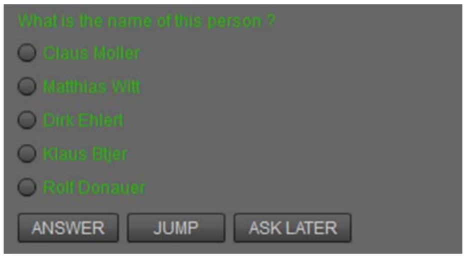

User Login
Initiates UI dialogue/model parameters and requests list of available documents
Select Document
User selects from choice of videos
Start Processing
Initiates an annotation session across the entire CASAM system with selected video
Retrieve Explicit Shot bounds
Requests first shot bounds generated by KDMA
Session Alive?
Yes
No
Retrieve UI Interpretations
Requests newly constructed UI interpretations to be sent from CASAM HCI backend
Retrieve UI queries
Requests newly constructed UI queries to be sent from CASAM HCI backend
Refresh visual components
Redraw of all visual components is requested. Any new visual elements are then displayed in appropriate canvas.
Stop Processing
CASAM system is requested to end processing for this session

Fig. 4. Process flow in the CASAM HCI Frontend

and possibly edit them (through the review area). More screen space was given to the timeline as opposed to the annotation canvas as it was felt that it was more important. The editing of text is quite a simple process that does not require too much space, whilst the timeline serves many purposes and needs to give the user a sense of what is contained in the video and the annotations that have been added at different time points.

The process flow for the frontend is outlined in figure 4. Once the user has logged in and chosen a video, the HCI frontend begins to poll the HCI backend for information. First it requests shot bounds. These are produced on initialisation of KDMA and indicate the natural shot boundaries within a video. These are used by the timeline component which is discussed later. The process flow then enters a loop which continuously polls the HCI backend for new UI queries and interpretations that have been produced by the HCI backend and populated with relevant times and human readable text. Once new queries and interpretations have been received and handled a refresh of the interface is performed which results in all newly received annotation being displayed as appropriate. This loop only ends when the user indicates they wish to stop the annotation session. The user ends the annotation session by clicking the “End” button located at the top right of the interface. At this point the annotation session is terminated across the entire CASAM system.



What is the name of this person ?

○ Claus Moller

○ Matthias Witt

○ Dirk Ehlert

○ Klaus Btjer

○ Rolf Donauer

Fig. 5. An example query as presented in the user interface

The User Input Canvas has a number of functions, each represented by a tab at the head of the canvas and each related to a requirement for user input. It allows the user the freedom to annotate the document as they see fit using plain text. Once submitted this is then sent on to the KDMA component for text processing. This may lead to new KDMA assertions, and so lead to further RMI interpretations/queries.

The user input canvas also presents queries to the user in a human readable form and captures the user response. An example of how queries are presented to users is outlined in figure 5. When a query is presented, the video playback pauses, creating a clear interruption of the annotation task.

The user input canvas also provides a tab to allow the user to view the IPTC data for the chosen multimedia document.

The Annotation Canvas Given the nature of the CASAM system, it is clear that there needs to be some facility by which users can review annotations that the system has autonomously generated. The annotation canvas was designed to serve this purpose.

To view the contents of an annotation, users simply click on one of the labels displayed within the video timeline or, more generally, a location along the timeline. This then populates the annotation canvas with a visual representation of the annotation. The semantic annotation is displayed in the form of two distinct tag clouds. The first represents the concepts which are particular to the current position of the playhead in the video. The second consists of concepts that are either related to the whole video or cannot be associated with any particular part of the video.

The size of the tags is determined by the confidence value associated with the underlying interpretation/assertion. Tags in the cloud are interactive components and respond to clicks from the user. If the user clicks on the tag text itself this asserts that the tag is correct and so the confidence increases and so does the tag size. If the delete symbol (in the form of a cross adjacent to the tag) is selected then the confidence is set to zero and the tag is removed from the tag cloud.

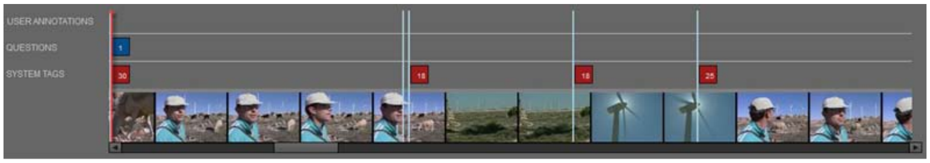


Fig. 6. The Timeline visual component

The Video Player displays the playing video and provides a set of video controls. This component is purposely kept simple whilst providing the necessary control over video playback allowing the user to play/pause and relocate the playhead.

The Timeline Canvas It is important to ensure the look and feel of the interface is familiar to media professionals and to hide, as far as possible, the complexities of the processing and representations that are taking place in the background. Most commercial and research applications that focus on video editing or annotation provide a means of allowing users to easily navigate the video and review the annotations that have been made [8, 10, 16]. This is often in the form of a visual timeline that contains representative keyframes for different shots and helps users in navigating to important sections of video for review, and generally in getting a sense of the video's contents.

Figure 6 shows a screenshot of the timeline that was incorporated into the interface. The content of the video is portrayed using thumbnails generated from still shots from the video at fixed intervals. This approach provides an insight into the content of individual video shots but also gives a clear indication of where the shot boundaries are. In addition to thumbnails, an explicit set of shot bounds are calculated by the KDMA component and displayed as vertical blue lines overlaying the timeline. Shot bounds play an important role in annotation since they provide a natural decomposition of the video.

All annotations that are associated with the content of the video are related to one or more of these shot bounds. Each shot, portrayed within the timeline by shot bounds, may potentially contain a label for semantic annotations, free text annotation and queries. These labels provide an overview of annotation quantity and so, when viewed amongst the other shot bounds in the timeline, provide an indication of the density of annotation throughout the video. If the user clicks on one of these labels they can see the contents of the annotation via the annotation canvas.

A timeline pointer also moves as the video is playing so that the user has some feedback of how the current video playhead position relates to the content of the video and annotation. This pointer is portrayed in the timeline as a vertical red line. The timeline and video player are synchronized so if, for example, the user clicks on the timeline, both the timeline pointer and the video player will jump to that particular time point and the annotation canvas will be updated.

4.4 Intelligent Interaction

An adaptive interface is one that behaves differently in different contexts and for different users. This allows interaction to be more effective and supportive, and less disruptive and intrusive for a given task, individual user or type of user. However, previous research [6] shows clearly that adaptation of interfaces should be limited to dialogue, interactions and content. It should not involve changes to the layout or functionality of the interface as this disrupts learning through spatial awareness. In other words, visual elements of the interface should stay essentially the same and in the same positions, the same controls should be available, and those controls should perform the same functions for different users.

Adaptation should be driven by knowledge about the user (or user type) and their preferences or roles, the state of the task at hand, the state of the interface, a history of the current session, or a combination of all of these [7]. So in order to make the CASAM interface adaptive we must consider two things: the nature of the adaptation, and what will drive it - that is, the process that will decide how the interface behaves for a given user.

Adaptation and user profiles The nature of the adaptation of the interface depends largely on the user, their role, preferences, and the task they are working on. Information about the user is used to decide which course of action to take, and there are three aspects to this: how personal it is, how different behaviours are represented and implemented in the system, and how values or parameters for the implementation are produced.

Users can be described either individually or as types or classes. CASAM adapts its behaviour at the level of user types rather than individual users. The user selects their user type at the login stage. As the system has been developed with journalism in mind there are two roles already defined: journalists and archivists. Each role has different priorities, constraints and resources available and so will be better supported by different interactions with the system.

- **Journalists** may have relatively short periods of time in which to work. They can move backwards and forwards through the video, and might be more interested in context than content. There may or may not be existing annotation for them to work with. Given the time constraints any interruptions or explicit system requests for information, directed towards the user, need to be dealt with swiftly and so the video playback pauses whilst waiting for a user response but will automatically continue after the request is fulfilled.
- **Archivists** will likely have more time than a journalist and might be more interested in content than context - that is, they want to describe the content as comprehensively as possible, rather than in the context of a particular story or issue. The objective of an archivist is to optimise the annotation for retrieval. To support this role the interface provides explicit opportune moments for the user to address requests for information from the system

by pausing at the end of each video segment. This also provides a suitable time for the user to submit free text annotation relevant to that video segment. An archival session is expected to consider multimedia documents with significant existing annotation which will be extended and improved upon. Newly-generated automated annotation may still be added as the session continues.

Adaptation of Dialogue Due to the intended collaborative nature of the CASAM system it is critical that dialogue with the user be effective. Dialogue initiated by the system consists of queries that the system would like the user to answer in order to help it construct a semantic model of the contents of the video. The video is paused when a query is presented (for both the journalist and archivist user types), which constitutes an interruption to the user's task of annotating the video.

Early usability testing indicated that queries became "irritating" and "intrusive" as the annotation session progressed. Some users suggested that the ability to defer a query or to jump directly to the relevant video content would be useful. In general it was apparent that the queries tended to interrupt users' flow of work, whether that was entering text or simply watching the video. In order to introduce an effective dialogue with the user, a thorough assessment of the impact of queries on the user experience was required.

The impact of interruptions in human-computer interaction has received much attention from researchers. Results have found that interrupting users from their primary task can result in a negative impact on their performance [11, 17]. Other studies have found that interruptions can also have a detrimental impact on the time taken to complete a task [13, 5], decision making ability [18], a user's emotional state [1, 15] and increase user error during a task [11]. Researchers have investigated interrupting users at "opportune" moments in order to reduce negative impact. This may include times of lower mental workload or inherent breakpoints in the task [2, 14, 9, 12].

In related work, Creed et al [3] have shown how interruptions which occur in context are preferential in the CASAM system. 'In context' refers to interruptions where the query is directly related to the part of the document that the user is currently annotating. Interruptions 'out of context' were significantly more irritating and perceived to be more mentally demanding. Based on these results, it is clear that system-initiated dialogue needs to avoid the disruptive effects of presenting the user with a query at the wrong time. In order to determine the most opportune moment to present a query, a cost-benefit model was employed.

Cost-benefit model CASAM currently implements a cost-benefit model where each query has a benefit and a cost. The benefit is a measure of the importance of the query to the rest of the system, for example how useful the answer will be in allowing RMI to disambiguate between a number of options. The cost is a measure of how disruptive the user is likely to perceive the query and, by extension, how much impact it will have upon their performance on the annotation task. For example, if the query concerns a different video segment to the one the user is currently viewing, the cost of asking it will be higher than a query concerning the same segment.

The cost-benefit algorithm is driven by a mechanism which requests queries that can be displayed at the current time and playhead position. Cost is calculated using a variety of inputs drawn from the current state of the interface, a record of previous queries, the user's current position in the video and the user type. Four factors are currently considered when calculating cost and each results in a normalised real value in the interval $[0, 1]$ and a total cost is calculated as a weighted sum. These weights differ dependent upon user type.

- **Playhead time difference** is the time difference between the current playhead position (and therefore the video segment the user is currently working on) and the video segment the query refers to. The bigger this difference the higher the cost.
- **Similarity** to the preceding query. If the new question is similar to the previous question this may have an impact on the interruption cost. Queries that contain the same assertions as previous queries are considered to be irritating, whereas queries that contain assertions that are related to previous queries may be considered to build on those queries and so have a lower cost.
- **Video paused** depends on the current play state of the video. If the video is paused the cognitive load of the user is presumed to be lower, and therefore there is a lower cost of interruption than if the video were playing.
- **Repetition** of the same question. The user may opt to defer answering a question. In this case the system may ask that same question again later. A count is kept of the number of times a query has been deferred, and cost rises with this counter.

The weights for each user type were selected based on the results of recent research into the effects of interruptions within the CASAM user interface [3].

5 Discussion and Further Work

CASAM, like all mixed-initiative systems, could benefit from a more complete user modelling approach. Currently the user type is represented by the set of weights used in the cost-benefit model and the level of functionality available in the interface. The set of weights and inputs to the cost calculations could provide a richer adaptive interface. Finding optimal parameter settings for the cost estimation algorithm is a challenge. This could be attempted through extensive user testing or using domain knowledge. There may also be value in adapting these parameters dynamically during the session based on the users prior interaction. It would also be interesting to allow the system to control the playhead position, so that it may ask any query at any time and display relevant sections of the video. User testing would be required to ascertain whether this loss of control is excessively frustrating.

The end goal of adapting the dialogue is to improve the overall annotation performance, in terms of quality and speed. It is insufficient to simply measure quantity, as a greater quantity of annotation does not automatically equate to greater search success for users retrieving items from the archive for re-use, which is ultimately the purpose of the annotation in this case. Similarly, a qualitative comparison to a "gold standard" professional annotation is not necessarily informative as different styles of annotation may, in fact, support search equally well. Therefore, a definition of annotation quality based on search performance is essential both for determining appropriate parameters and for evaluating the success of CASAM and other annotation systems.

As it is unclear how a straightforward user interface for the visualisation and direct editing of semantic relationships might be provided, we have thus far taken the approach of flattening the semantic tree and simply displaying concepts as "tags" which can be confirmed or deleted, and via natural-language queries to modify semantic relationships. Although users are very comfortable with tags as a form of annotation, they expect to be able to edit the tags and to add their own. It is non-trivial to provide this functionality however, as edits could be ambiguous, e.g. whether they are correcting a misspelling of a name or asserting that it somebody else. The addition of new tags would require the new concept to somehow be related semantically to other existing concepts. If it is not practical to expect the user to rectify these problems via the interface, they must be tackled behind the scenes using a probabilistic approach. The generation of natural language queries from a list of assertions framed by the ontology is also non-trivial, especially in the generalised case. The simple approach currently taken sometimes leads to queries that users report difficulty in understanding.

6 Acknowledgements

This work was supported by EU grant FP7-217061.

References

1. P. Adamczyk and B. Bailey. If not now, when?: the effects of interruption at different moments within task execution. In *Proc. SIGCHI conference on Human factors in computing systems*, volume 6, pages 271–278. ACM Press, 2004.
2. B. P. Bailey and S. T. Iqbal. Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Transactions on Computer-Human Interaction*, 14(4):1–28, 2008.
3. C. Creed, C. Bowers, R. Hendley, and R. Beale. User Perception of Interruptions in Multimedia Annotation Tasks. In *Proc. 6th Nordic Conference on Human-Computer Interaction*, pages 619–622. ACM Press, 2010.
4. C. Creed, P. Lonsdale, R. Hendley, and R. Beale. Synergistic Annotation of Multimedia Content. In *Proc. 3rd Intl. Conference on Advances in Computer Human Interactions*, pages 205–208. IEEE, 2010.
5. E. Cutrell, M. Czerwinski, and E. Horvitz. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. In *Human-Computer Interaction, INTERACT 2001*, pages 263–269. IOS Press, 2001.

6. A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human Computer Interaction*. Prentice Hall, 3rd edition, 2003.
7. G. Fischer. User Modeling in HumanComputer Interaction. *User Modeling and User-Adapted Interaction*, 11:65–86, 2001.
8. J. Hagedorn, J. Hailpern, and K. G. Karahalios. VCode and VData: Illustrating a new framework for supporting the video annotation workflow. In *Proc. Advanced Visual Interfaces*, pages 317–321. ACM Press, 2008.
9. S. T. Iqbal and B. P. Bailey. Effects of intelligent notification management on users and their tasks. In *Proceeding of the 26th annual CHI conference on Human factors in computing systems, CHI '08*, pages 93–102. ACM Press, 2008.
10. M. Kipp. Anvil: A generic annotation tool for multimodal dialogue. In *Proc. 7th European Conference on Speech Communication and Technology*, 2001.
11. K. A. Latorella. Effects Of Modality On Interrupted Flight Deck Performance: Implications For Data Link. In *Proc. of Human Factors and Ergonomics Society*, pages 87–91, 1998.
12. G. Mark, D. Gudith, and U. Klocke. The Cost of Interrupted Work: More Speed and Stress. In *Proc. 26th CHI conference on Human factors in computing systems*, pages 107–110. ACM Press, 2008.
13. D. C. Mcfarlane. Coordinating the Interruption of People in Human-Computer Interaction. In *Human-computer interaction, INTERACT'99*, pages 295–303. IOS Press, 1999.
14. Y. Miyata and D. Norman. *Psychological issues in support of multiple activities*, pages 265–284. Lawrence Erlbaum Associates, 1986.
15. C. Monk, D. Boehm-Davis, and JG. The attentional costs of interrupting task performance at various stages. *Proc. Human Factors and Ergonomics Society*, pages 1824–1828, 2002.
16. H. Neuschmied, R. Trichet, and B. Merialdo. Fast annotation of video objects for interactive TV. In *Proc. 15th Intl. Conference on Multimedia*, pages 158–159. ACM Press, 2007.
17. J. Rubinstein, D. Meyer, and J. Evans. Executive control of cognitive processes in task switching. *Journal of Experimental Psychology Human Perception and Performance*, 27(4):763–797, 2001.
18. C. Speier, J. S. Valacich, and I. Vessey. The Influence of Task Interruption on Individual Decision Making: An Information Overload Perspective. *Decision Sciences*, 30(2):337–360, 1999.