Annotation of Heterogenous Media Using OntoMedia

Michael O. Jewell, K. Faith Lawrence, Adam Prügel-Bennett, and m. c. schraefel

University of Southampton, Southampton, SO17 1BJ, UK
moj|kf03r@ecs.soton.ac.uk

Abstract. While ontologies exist for the annotation of monomedia, interoperability between these schemes is an important issue. The OntoMedia ontology consists of a generic core, capable of representing a diverse range of media, as well as extension ontologies to focus on specific formats. This paper provides an overview of the OntoMedia ontologies, together with a detailed case study when applied to video, a scripted form, and an associated short story.

1 Introduction

The OntoMedia ontology was developed to support the markup of heterogenous media content [1][2]. This includes the annotation of monomedia (drawing on existing technologies such as the CIDOC Conceptual Reference Model[3], ABC[4], and the Functional Requirements for Bibliographic Records[5]) and multimedia (drawing on MPEG and audiovisual media metadata standards).

This paper is divided into two sections. The first provides an overview of the OntoMedia model and examines the core and extension ontologies therein. The second demonstrates the capabilities of the OntoMedia ontology, with the annotation of Philip K. Dick's short story 'We Can Remember It For You Wholesale'[6] and the related film Total Recall. Example queries were carried out to demonstrate the links between the content of the two forms and to illustrate the potential of the system.

2 OntoMedia: An Overview

OntoMedia is based on an Entity/Event system. This is similar to existing systems, and provides a natural divide between spatial and temporal information. We define an Entity as an object or concept, with an Event describing an interaction between one or more Entities. During this interaction, zero or more attributes of those Entities are modified or a new Entity is created. An Entity may have an attribute set to show that it no longer exists but the Entity itself is not destroyed.

The subclasses of the Entity construct fall into three different types. Those related to objects both physical and abstract (Being, PhysicalItem, and AbstractItem), those related to spatial models (Space) and those relating to time (Timeline and Occurrence). The current version of the ontology is available at the project website[1].

The ontology core classes are extended as follows:

[1] http://ontomedia.ecs.soton.ac.uk

Extensions	Where the core provides only the very basic classes, Extensions provides more detailed subclasses. This includes Being, to describe people, and a set of Trait classes to define attributes of entities.
Events	The Events hierarchy extends the Event class from the Core to provide specialised events. This covers Action, Gain, Loss, Travel, and also additional properties for these extra types. These are based on the most common types of events found in literature.
Fiction	In order to provide specialist classes for fiction representation, this set of classes includes Character, which builds upon Being, and other properties to denote spoiler information and accuracy.
Media	The Media class in Core only defines a basic MediaItem, so it is necessary to define more detailed classes to describe specific media items. This hierarchy includes audio, image, photograph, textual, and video subclasses.
Miscellaneous	Finally, the Miscellaneous set provides classes which may be used by any or all of the other classes. This includes colour information, name representation, and geometry.

Graphical interfaces for OntoMedia are under development, including Meditate (see Section 3.2), to aid ontology extension and entity creation. This program works directly with OWL and RDF files.

3 Annotating Multimedia: A Case Study

3.1 Screenplay Annotation

To ease the annotation of screenplays into a machine-readable form, a simple format was designed. Named SiX (Screenplays in XML), the format is designed to wrap around existing content. So, given a script, very little alteration is required to render it readable. Four ‘core’ tags are provided: transition, location, dialogue, and direction. Transition tags denote a change in scene, and can be cuts, fades in, fades out, dissolves, or blackouts. Locations are also straightforward, with two attributes (‘time’ and ‘pos’) specifying the time period (day or night) and the position (interior or exterior) respectively, and the enclosed text giving a more detailed description.

The dialogue tag, as expected, denotes speech within the script. It has four attributes: ‘paren’, ‘speaker’, and ‘ctd’. Paren, short for parenthetical, allows for a description of the manner in which dialogue should be spoken (e.g. ‘nervously’); speaker identifies the character delivering the dialogue, and ctd (when set to 1) specifies that the dialogue is a continuation from a previous event. An optional ‘voiceover’ flag is also allowed, which specifies that the text is read by a narrator. Finally, the direction tag, which does not take any attributes, simply describes an action taking place in the script.

To provide a suitable amount of metadata relating to the script, SiX allows for Dublin Core annotation inside an <sc:info> block. As such, it is possible to denote the creators, creation date, a description of the script, and (most importantly) the title of the work. To ease readability of SiX documents, a custom XSL was created to render them into a style conforming to the Oscar requirements for submitted screenplays. This includes margins, font sizes, and case requirements.

Listing 3.1. A section from Total Recall in SiX format

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no" ?>
<sc:script xmlns="http://www.w3.org/1999/xhtml"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:sc="http://mikesroom.org/script">
  <sc:info>
    <dc:title>Total Recall</dc:title>
  </sc:info>
  <sc:location time="day" pos="int">HILTON - CORRIDOR/SERVICE ELEVATOR - 6TH FL.</
      sc:location>
  <sc:dialogue speaker="Lori">Doug...you wouldn&apos;t hurt me, would you, honey?
      </sc:dialogue>
  <sc:direction>She sees his expression.</sc:direction>
  <sc:dialogue speaker="Lori" ctd="1">Sweetheart, be reasonable...We&apos;re
      married.</sc:dialogue>
  <sc:direction>Lori stealthily reaches behind her back for a concealed gun and
      pulls it on him.</sc:direction>
  <sc:direction>Quaid shoots Lori in the forehead, leaving a clean, small hole
      between her eyes.</sc:direction>
  <sc:dialogue speaker="Quaid" paren="rising,␣to␣Melina">Consider that a divorce.
      </sc:dialogue>
</sc:script>
```

Listing 3.2. Binding an OntoMedia representation to a SiX screenplay resource

```
<omsi:ScriptItem rdf:ID="Scene1">
  <omsi:uri>http://ecs.soton.ac.uk/˜moj/script/totalrecall.xml</omsi:uri>
  <omsi:has-subregion rdf:resource="#Loc1" />
  <omsi:has-subregion rdf:resource="#Act1" />
</omsi:ScriptItem>
<omsi:ScriptItem rdf:ID="Loc1">
  <rdfs:label>HILTON - CORRIDOR/SERVICE ELEVATOR - 6TH FL.</rdfs:label>
  <omsi:uri>http://ecs.soton.ac.uk/˜moj/script/totalrecall.xml#xpointer(element(
      scene1/1))</omsi:uri>
  <omsi:has-expression rdf:resource="#Corridor1" />
</omsi:ScriptItem>
```

Once the screenplay is annotated (see Listing 3.1), it is possible to link directly from OntoMedia to the script representation. This is achieved with the use of ScriptItem objects, which extend the MediaItem class mentioned previously. To provide a flexible means of referencing the marked screenplay, xpointers are stored in attributes of the objects, so references may be specific (pointing to an element with a given id) or relative (pointing to the Nth element of a file) and a script may be held in multiple files. Furthermore, as with other MediaItems, a ScriptItem may have subregions. As such, it is possible to have one ScriptItem representing a scene, and that containing other ScriptItems representing individual script elements.

To tie these items to OntoMedia Expressions, the has-expression property is employed. Listing 3.2 shows two ScriptItems, one covering the whole screenplay file, and a second tying a Location to its script representation. It is, of course, possible to bind from a ScriptItem to an occurrence, which is essential for the linking of action or dialogue to the corresponding script information. Note that the URLs used in these examples are available online.

3.2 Character Annotation

To ease the annotation of the (otherwise complex) character instances, the Meditate application was designed to act as an easy-to-use interface between the user and the RDF. By taking the options available to the user directly from the OWL ontology definition and exporting the created entities in RDF, Meditate removes the complexity of editing RDF files.

Meditate provides the ability to mark up a large amount of information regarding characters, including the links between them. As well as basic information about the character such as names and gender the entity also contains links to related entities, in this case the Being which represents Arnold Schwarzenegger and the Character which represents "Douglas Quail" from "We Can Remember it for you Wholesale"[6] are linked via the Character of "Douglas Quaid" from Total Recall.

As Meditate allows for the saving and loading of RDF and OWL to both local and remote repositories, it therefore acts as an intuitive interface for the editing of information which can then be used by and in conjunction with other applications.

3.3 Event Annotation

OntoMedia contains three elements which are essential for the content within a medium: Timeline, Event, and Occurrence. All of these are necessary to fully describe a sequence of events, although not all events need actually occur (a character may *want* an event to occur, but it may not actually exist on a timeline). A medium may also have several timelines, where one timeline may be another character's view, or consist of events which occur in a dream.

In the case of the Total Recall segment, we are annotating a fight sequence. Douglas Quaid, the male protagonist, is being dragged unconscious to an elevator by his captor, Lori. Once they reach the elevator, Quaid's ex-lover (and the female protagonist) Melina arrives in the elevator, and the fight commences. This involves instances where Lori has the upper hand, where Melina has the upper hand, and, finally, where Quaid takes control of the situation and kills Lori. There are also two instances where both Lori and Melina lose items - one where Lori's gun is kicked away, and one where Melina's knife is shot away.

As such, four subclasses of Event are utilised - Action, Transformation, Gain, and Loss. In this instance, Action is used for a scene in which nothing occurs plotwise, but there is action (a fight in this case); Loss/Gain are used to represent a character losing or gaining an item; and Transformation to represent the alteration of a character's state. As such, it can be easily inferred that Lori has the upper hand in the first event, as she is specified as the subject entity, and that the gun kicked away by Lori in one event is retrieved by Quaid in another. Finally, it is clear that the character Lori is killed, as she changes state from Alive to Dead.

Once the events are declared, it is a simple matter to create the timeline and add the required occurrences. Occurrences are very simple - they specify the position of the events on the timeline relative to the other occurrences. This also allows for events to be used multiple times, or for events to coincide. Listing 3.3 demonstrates how the Timeline and Occurrences are created.

Listing 3.3. Defining the occurrences of events

```
<ome:Timeline rdf:ID="Total_Recall.Timeline" />
<ome:Occurrence rdf:ID="Act1Occ">
  <ome:involves rdf:resource="&base;Total_Recall_Douglas_Quaid_Character_102" />
  <ome:involves rdf:resource="&base;Total_Recall_Lori_Character_145" />
  <ome:precedes rdf:resource="#Act2Occ" />
  <ome:timeline-ref rdf:resource="&base;Total_Recall.Timeline" />
</ome:Occurrence>
```

3.4 Example Queries

To test the flexibility of OntoMedia, the annotated scene from Total Recall was imported into a Sesame triple store. The following are a few examples of the RDQL queries and the results which were obtained.

Locating Specific Events In this case we want to find all cases where Lori loses something, defined as the location in the script where it occurs.

```
SELECT ?uri
WHERE (?item rdf:type omsi:ScriptItem) (?item omsi:uri ?uri)
      (?item omsi:has-expression ?expr) (?expr rdf:type ome:Occurrence)
      (?event ome:has-occurrence ?expr) (?event rdf:type ome:Loss)
      (?event ome:has-subject-entity ?char) (?char rdfs:label "Total␣Recall.Lori")

Result:

"http://ecs.soton.ac.uk/~moj/script/totalrecall.xml#xpointer(element(scene1/9))"
            #Act8Eve2
```

Querying Across Media As Total Recall was (partially) based on the book ‘We Can Remember It For You Wholesale’, we can specify this using OntoMedia. This example shows how we can find links across media types.

```
SELECT ?char2
      WHERE   (?character ome:is-shadow-of ?char2)
      (?character rdfs:label "Total␣Recall.Douglas␣Quaid")

Result:

http://ontomedia.ecs.soton.ac.uk/ontologies/ontomedia/entity_store/TotalRecall#
     Wholesale_Douglas_Quail_Character_1
```

4 Conclusions

While OntoMedia is still under active development, the ontology in its current form is extremely powerful. The core allows for the representation of a wide range of media, and the extension classes allow for the addition of domain-specific knowledge without compromising the transparency of the framework. The provision of cross-linking between media types introduces a rich scope for inference, with applications from recommender systems to desktop search tools.

While it is beneficial to have a flexible ontology, it is also essential that the tools be available to ease the annotation process. The creation of Meditate and SiX are first steps towards this goal. There are also efforts underway to combine OntoMedia with other applications. For example, Entity Stores provide multi-user editing functionality as well as versioning control, and work is in progress to combine OntoMedia[7] with the EPrints[8] document archiving software and SEMEDIA's DBin[9] platform. This will provide a means for users to submit fiction to an archive, together with links to domain information.

As the Entity Store does not differentiate between present-day or historical characters, it can also be used for cultural heritage material such as myths and legends. This allows the links between mythological characters and motifs to be explored and searched both within the mythology itself and between the mythology and popular culture.

OntoMedia provides a novel approach to the annotation of heterogenous media. As tools become more user-friendly and an increasing amount of multimedia is annotated, it will be exciting to see the emergent patterns through its application.

References

1. Lawrence, K.F.: OntoMedia - creating an ontology for marking up the contents of fiction and other media. In: Proceedings of 1st AKT Doctoral Colloquium, AKT (2005)
2. Lawrence, K.F., Tuffield, M.M., Jewell, M.O., Prugel-Bennett, A., Millard, D.E., Nixon, M.S., monica schraefel, Shadbolt, N.R.: OntoMedia - creating an ontology for marking up the contents of heterogeneous media. In: Proceedings of Ontology Patterns for the Semantic Web Workshop (ISWC 05). (2005) Available online at urlhttp://eprints.ecs.soton.ac.uk/11153/.
3. Crofts, N., Doerr, M., Gill, T., Stead, S., (eds), M.S.: Definition of the CIDOC CRM conceptual reference model. Reference document, International Council of Museums (2005)
4. Hunter, J.: Enhancing the semantic interoperability of multimedia through a core ontology. IEEE Transactions on Circuits and Systems for Video Technology (2003)
5. Saur, K.G.: Functional requirements of bibliographic records: final report. Technical report, IFLA Study Group on the Functional Requirements of Bibliographic Records, M"unchen (1998)
6. Dick, P.K.: We Can Remember It for You Wholesale. In: The Preserving Machine. Ace Books (1969)
7. Lawrence, K.F., schraefel, m.c.: Bringing communities to the semantic web and the semantic web to communities. In: Proceedings of WWW2006. (2006)
8. Gutteridge, C.: New developments in EPrints. In: CERN workshop on Innovations in Scholarly Communication (OAI4), CERN, Geneva (2005)
9. Tummarello, G., Morbidoni, C., Puliti, F., Piazza, F.: The DBin semantic web platform: An overview. In: WWW2005 Workshop on the Semantic Computing Initiative. (2005)