

Rules and ontologies in support of real-time ubiquitous application

Marek Hatala*, Ron Wakkary, Leila Kalantari

*School of Interactive Arts and Technology, Simon Fraser University, 2400 Central City,
10153 King George Highway, Surrey, BC, Canada V3T 2W1*

Received 6 May 2005; received in revised form 11 May 2005; accepted 12 May 2005

Abstract

The focus of this paper is the practical evaluation of the challenges and capabilities of combination of ontologies and rules in the context of realtime ubiquitous application. The ec(h)o project designed a platform to create a museum experience that consists of a physical installation and an interactive virtual layer of three-dimensional soundscapes that are physically mapped to the museum displays. The retrieval mechanism is built on the user model and conceptual descriptions of sound objects and museum artifacts. The rule-based user model was specifically designed to work in environments where the rich semantic descriptions are available. The retrieval criteria are represented as inference rules that combine knowledge from psychoacoustics and cognitive domains with compositional aspects of interaction. Evaluation results both from the laboratory and museum deployment testing are presented together with the end user usability evaluations. We also summarize our findings in the lessons learned that provide a transferable generic knowledge for similar type of applications. The ec(h)o proved that ontologies and rules provide an excellent platform for building a highly-responsive context-aware interactive application.

Keywords: Ontologies; Rule-based systems; User modeling; Context aware; Augmented reality; Audio; Museum guide

1. Introduction

Audio museum guides have existed for some time as a means of overcoming the scheduling inflexibility of group tours by museum docents. While beneficial in many respects, the audio guides are limited by their lin-

ear sequence and non-interactive structure. Bederson [3] developed a prototype utilizing portable mini-disc players and an infra-red system to allow museum visitors to explore at their own pace and sequence. As museum visitors approached artifacts on display, relevant audio information would be triggered on the mini-disc player and heard through headphones. Hyperaudio [16] provided visitors with palmtop computers and developed specific user models for adaptive systems within a museum setting. MEG [2] is a portable digital museum guide for the Experience Music Project in

* Corresponding author. Tel.: +1 604 268 7431;
fax: +1 604 268 7488.

E-mail addresses: mhatala@sfu.ca (M. Hatala),
rwakkary@sfu.ca (R. Wakkary), lkalanta@sfu.ca (L. Kalantari).

Seattle that allows visitors 20 hours of audio and video on demand. Visitors make their selections either by use of the keyboard within the PDA device or by pointing the device at transmitters located adjacent to artifacts.

In the previous works, the relationship of the digital content to the artifacts is either pre-planned and fixed, or the digital content is not networked and limited to the local device; in some cases both limits are true. ec(h)o employs a semantic web approach to the museum's digital content, thus it is networked, dynamic, and user-driven. The interface of ec(h)o does not rely on portable computing devices; rather it utilizes a combination of gesture and object manipulation recognized by a vision system.

The dynamic and user-driven nature of ec(h)o requires a highly responsive retrieval mechanism with a criteria defined by psychoacoustics, content, and composition domains. The retrieval mechanism is based on a user model that is continually updated as a visitor moves through the exhibition and listens to sound objects. The criteria are represented by rules operating on the ontological descriptions of sound objects, museum artifacts, and user interests.

One of the main goals of ec(h)o is to achieve an enhanced experience for the museum visitors without inserting an extra layer of technology between the visitor and the museum exhibit. Two mechanisms contribute to an accurate retrieval of sound objects in ec(h)o: the user model and ontology descriptions of objects.

With the development of the semantic web [4] the use of ontologies as a formalism to describe knowledge and information in a way that can be shared on the web is becoming common. Adoption of the standard for the ontology web language (OWL) [21] is propelling this trend toward large scale application in different domains. However, the utility of the ontologies is limited by the processing mechanisms that are smoothly integrated with this form of representation. Therefore there is an effort on the way to formalize the logic layer for ontologies. The semantic web rule language (SWRL) [21] is proposed as an important step in this direction, building on the experience of the previous work on RuleML [5]. Eventually the availability of standardized rule language for the semantic web will make it possible to use both ontologies and rules as a basis for innovative applications that are connected to the semantic web. The understanding of capabilities and implications of this combination will be essential for

successful deployment and adoption of these technologies. This paper aims at addressing some of these issues through the development of a ubiquitous system with some extreme requirements testing the capabilities of the emerging technological platform.

The paper is organized as follows. First we present the ec(h)o architecture and then we describe ontologies used in the ec(h)o. Section 4 describes the user model and Section 5 outlines the retrieval mechanisms for sound objects. Before we show the results of the evaluation in Section 7 we describe the implementation challenges and lessons learned in Section 6.

2. ec(h)o Architecture

The platform for ec(h)o is an integrated audio, vision, and location tracking system installed as an augmentation of an existing museum exhibition installation. The platform is designed to create a museum experience that consists of a physical installation, an interactive layer of three-dimensional soundscapes that are physically mapped to museum displays, and the overall exhibition installation.

Each soundscape consists of zones of ambient sound and "soundmarks" generated by dynamic audio data that relates to the artifacts the visitor is experiencing. The soundscapes change based on the position of the visitor in the space, their past history with viewing the artifacts, and their individual interests in relation to the museum collection. To achieve this type of audio experience the overall system must be integrated with a position tracking system that has a frequent update cycle and a high level of spatial resolution. A pattern of the user's movement can indicate the type of museum visitor [19] as well as user intentions [17].

When the user stops in front of an artifact, she is presented with three sound objects spatially positioned to the left, center, and right. By way of a gesture-based interaction, the visitor can interact with a single artifact or multiple artifacts in order to listen to related audio information. The audio delivery is dynamic and generated by agent-assisted searches inferred by past interactions, histories, and individual interests. The source for the audio-data is digital objects. In the case of ec(h)o, we developed a large sample set of digital objects that originated from the partner

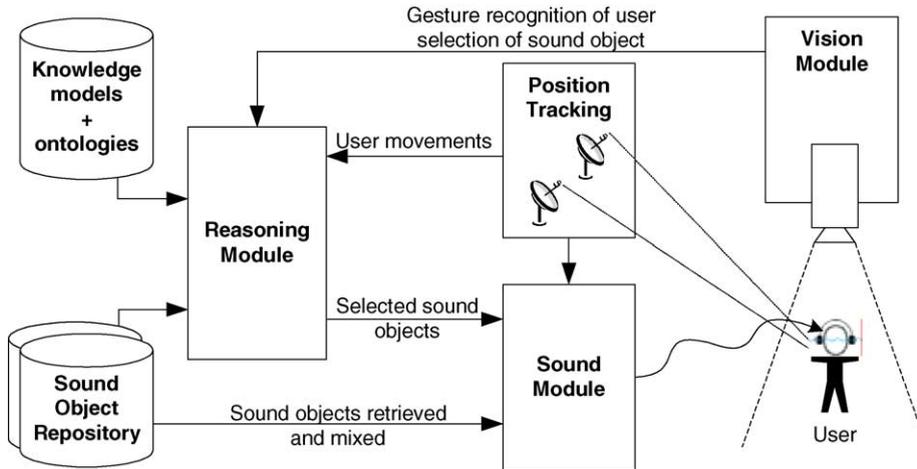


Fig. 1. ec(h)o High level architecture.

museums. These digital objects were used to populate the network of object repositories.

The ec(h)o architecture (Fig. 1) consists of four independently functioning modules: position tracking module, vision module, sound delivery module, and reasoning module. Two main types of events trigger the communication between the modules: the user's movement through the exhibition space and the user's explicit selection of the sound objects.

3. Semantic description of objects

We have identified two types of information as essential for ec(h)o:

- the content description of the user interests (user model), sound objects, and museum artifacts, and
- psychoacoustics and sound characteristics of the sound objects.

3.1. Ontologies for describing content

The ec(h)o interaction model is based on the semantic description of the content of the sound objects. We have developed a sound object ontology describing objects with several properties. As the ability to link to other museum collections is an important feature of ec(h)o, our ontology builds significantly on the standard conceptual reference model (CRM) for heritage content developed by CIDOC [7]. The CRM provides

definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation. To describe sound objects we use CRM TemporalEntity concept for modeling periods and events and Place for modeling locations. We describe museum artifacts using the full CRM model.

The content of the sound object is not described directly but annotated with three entities: concepts, topics, and themes. The concepts describe the domains that are expressed by the sound object such as evolution, behaviour, lifestyle, diversity, habitat, etc. Since the collections in individual museums are different, so are the concept maps describing these collections. A topic is a more abstract entity that is represented by several concepts, such as botany, invertebrates, marine biology, etc. To facilitate the mappings between topic ontologies in individual museums we have mapped the topics to the Dewey decimal classification [8] whenever

Table 1
Content related properties of sound objects

Property	Domain	Range
hasTheme	SoundObject	Theme
hasTopic	SoundObject	Topic
hasPrimaryConcept	SoundObject	Concept_of_interest
hasSecondaryConcept	SoundObject	Concept_of_interest
relatesToTemporalEntity	SoundObject	CRM.TemporalEntity
relatesToPlace	SoundObject	CRM.Place
	MuseumArtifact	
describesArtifact	SoundObject	MuseumArtifact

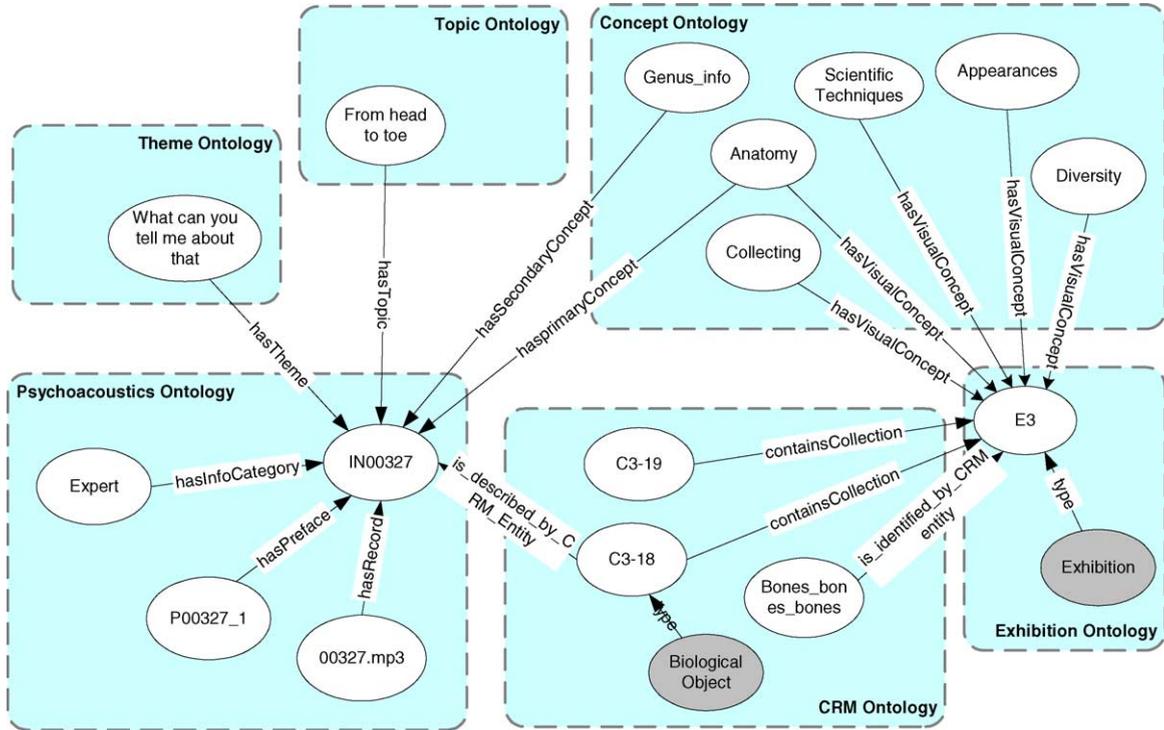


Fig. 2. ec(h)o Content ontologies.

possible. Finally, themes are defined as entities supported by one or more topics, for example, the theme of “bigness” in invertebrates and marine biology.

Table 1 shows content related properties with their domains and ranges.

In Fig. 2 the sound object ‘IN00327’ is annotated with concepts ‘Anatomy’ and ‘Genus Info’, has a topic

```

<rdf:Description rdf:about="&psch;#IN00327">
  <rdf:type><daml:Class rdf:about="&psch;#InfoNarration"/>
</rdf:type>
  <&psch;:hasPrimaryConcept rdf:resource="&concept;#Anatomy"/>
  <&psch;:hasSecondaryConcept rdf:resource="&concept;#Genus_info"/>
  <&psch;:hasTopic rdf:resource="&topic.daml;#From_head_to_toe"/>
  <&psch;:hasRecord
    rdf:resource="http://192.168.0.103/sound_objects/00327.mp3"/>
  <&psch;:hasPreface rdf:resource="&psch;#P00327_1"/>
  <&psch;:describes rdf:resource="&crm;#C3-18"/>
  <&psch;:hasInfoCategory rdf:resource="&psch;#Expert"/>
  <&psch;:hasTheme
    rdf:resource="&theme;#What_can_you_tell_me_about_that"/>

```

‘From Head to Toe’, and supports the theme ‘What Can You Tell Me About That’. The sound object ‘IN00327’ describes the artifact ‘C3-18’ that is modeled as an instance of ‘Biological object’ type in the CRM model described by the ‘Common dolphin skull’ object. The exhibit ‘E3’ from the exhibit ontology holds the in-

formation about the artifacts in the particular exhibit. In addition, ‘E3’ is annotated with concepts ‘Collecting’, ‘Anatomy’, ‘Scientific Techniques’, ‘Diversity’, and ‘Appearances’.

The ontologies for ec(h)o were modeled in DAML + OIL. The DAML + OIL representation¹ of the IN000327 audio object is shown below

In ec(h)o the ontological concepts are transformed into the Jess facts that represent RDF triples (see imple-

¹ For readability we use XML entities to refer to namespaces in this paper. For example, &psch; refers to the namespace <http://echo.iat.sfu.ca/owl/psychoacoustic.daml>, other references are self-explanatory.

mentation section for details). The above DAML + OIL description of the audio object IN000327 is represented with the following facts (with PropertyValue being a fact name used for all RDF triples):

```
(PropertyValue &rdf;#type &psch;#IN00327 &psch;#InfoNarration)
(PropertyValue &psch;#hasPrimaryConcept &psch;#IN00327
  http://echo.iat.sfu.ca/owl/concept.daml#Anatomy)
(PropertyValue &psch;#hasSecondaryConcept &psch;#IN00327
  &concept#Genus_info)
(PropertyValue &psch;#hasTopic &psch;#IN00327
  &topic;#From_head_to_toe)
(PropertyValue &psch;#hasRecord &psch;#IN00327
  http://192.168.0.103/sound_objects/00327.mp3)
(PropertyValue &psch;#hasPreface &psch;#IN00327 &psch;#P00327_1)
(PropertyValue &psch;#describes &psch;#IN00327 &crm;#C3-18)
(PropertyValue &psch;#hasInfoCategory &psch;daml#IN00327
  &psch;#Expert)
(PropertyValue &psch;#hasTheme &psch;#IN00327
  &theme;#What_can_you_tell_me_about_that)
```

For details on creation of content and related ontologies see [23].

3.2. Psychoacoustics and sound characteristics ontologies

The auditory interface of ec(h)o follows an ecological approach to the sound composition. It provides the basic mechanisms of navigation and orientation within the information space. Three areas are taken into account: psychoacoustic, cognitive, and compositional problems in the construction of a meaningful and engaging interactive audible display. Psychoacoustic characteristics of the ecological balance include spectral balancing of audible layers. Cognitive aspects of listening are represented by content-based criteria. Compositional aspects are addressed in the form of the orchestration of an ambient informational soundscape of immersion and flow that allows for the interactive involvement of the visitor.

Table 2
Psychoacoustic properties for the Sound Object

Property	Domain	Range
hasSpectralDensityCenter	SoundObject	<Number>
hasSpectralDensityWidth	SoundObject	<Number>
hasBandwidth	SoundObject	<Number>
relatesToEnvironment	SoundObject	Physical_Environment
relatesToEvent	SoundObject	CRM_Event
hasSource	SoundObject	SourceTypeValue (e.g. AnimalSound, HumanEnvironment-Sound)

Table 2 shows the psychoacoustics ontology that defines the characteristic of the sound objects that are used by the composition rules.

4. The user model

In the core of the ec(h)o's reasoning module is a user model [22] that is continually updated as the user moves through the exhibition and selects sound objects.

Fig. 3 shows an interaction schema of the user model with other modules. There are two main update sources in the system. First, as the user moves through the exhibition, the speed of the movement due to stopping or slowing down at different artifacts provide updates to the user model. The user's behavior type is computed based on the speed and homogeneity of the user's movement. Stopping and slowing down in front of an artifact are interpreted as interest in topics represented by the artifact. The user interests and intentions influence the presentation of soundmarks. For example, soundmark radius and volume is increased for those artifacts that correspond with current user interests. Another example can be the reduction of the number of soundmarks in the exhibition, if the user's recognized intent is to quickly cross the room.

The second source of updates to the user model considers the user's direct interaction when selecting a sound object. In the model, this maps to an increased user interest in topics presented by the sound object and updates the user's interaction history. We describe the user model and retrieval mechanism in detail below.

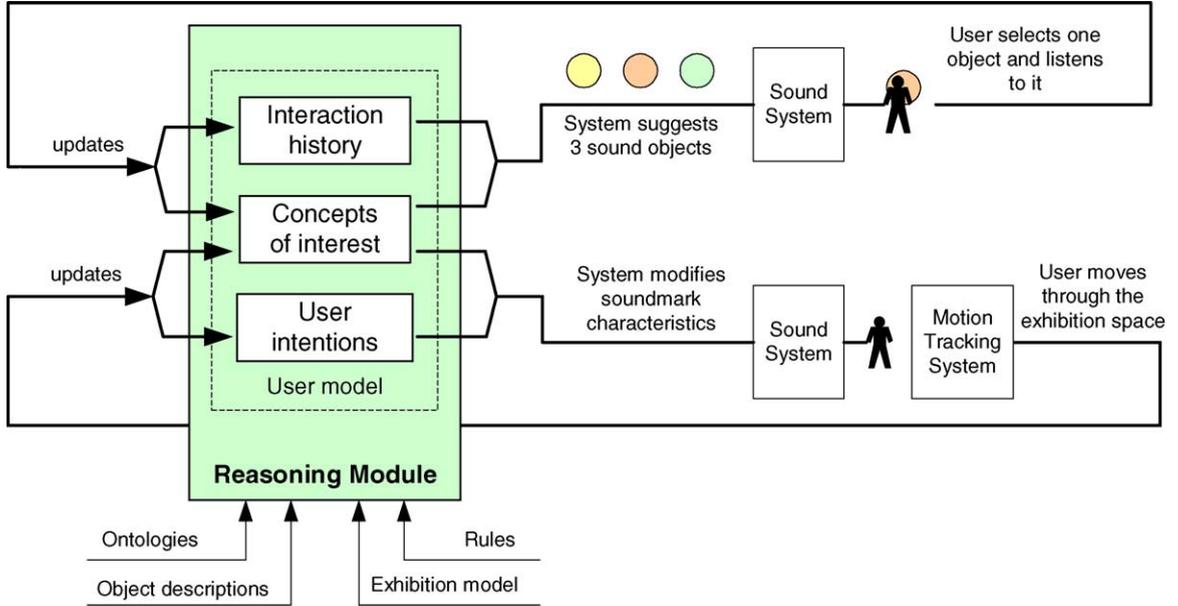


Fig. 3. Interaction of user model with other modules.

4.1. User model components

Interaction history is a record of how the user interacts with the ec(h)o-augmented museum environment. Two types of events are stored in the interaction history: the user's movement and the user's selection of objects. The user's path through the museum is stored as discrete time-space points of locations on the path. A second type of information stored in Interaction History is the user's selections in the form of URLs of sound objects.

User behavior in the museum context is well studied in museum studies [9] and is used in several systems personalizing the user experience [18,19]. In the case of ec(h)o, several categorizations were used; for example, one user may go through almost every artifact that is on his/her way, and another user may be more selective and choose artifacts that have certain concepts. Our categorization of user types is based on Sparacino's work [19] and it classifies users in three main categories: (1) the avaricious type who approaches artifacts in a deliberate and sequenced manner, (2) the selective type who explores certain concepts thoroughly, and (3) the busy type who wants a general idea of the exhibitions by browsing quickly through the museum.

In ec(h)o, the user behavior is not static. It continually updates by considering the location data accumulated in the previous 5 min; in addition to considering topics of previously selected sound objects.

User interests are represented as a set of weighted concepts from the ontology. In ec(h)o each artifact/exhibition is annotated with a set of concepts. The sound objects address a set of particular concepts as well. The system updates the user interests in response to two update channels described above. The interaction of the user and artifacts and sound objects is stored in the Interaction History that together with the user behavior type are used to infer the visitor's interests.

The following rule `concept-evol-choose--1` shows an example of how concepts of interest are updated in the user model. The `?*user-model-concepts*` object accumulates contributions from all activated rules first and indicates that the user model has to be updated. After all contributions are made, the rule `update-user-model---1` (with lower salience value) fires and recalculates the user interests values. It then inserts facts representing values of user interests into the knowledge base. These facts are used in the ranking of sound objects (described in Section 5).

```

(defrule concept-evol-choose-1---1
  (declare (saliency 100))
  (user-group (user ?u) (group 1))
  (chosen-object (user ?u) (context ?e)
                 (object ?in) (time-chosen ?t))
  (object-concept (object ?in) (concept ?c) (level ?n))
=>
  (call ?*user-model-concepts* increaseConceptLevel ?u ?c ?e
      objectSelected ?t (* ?n ?*select-concept*))
  (assert (update-user-model ?u ?e ?t)))

(defrule update-user-model---1
  (declare (saliency 99))
  (user-group (user ?u) (group 1))
  (update-user-model ?u ?e ?t)
=>
  (apply assert-string
    (call ?*user-model-concepts* formModelFactsForUser ?u ?t)))

```

The `?*user-model-concepts*` is a Java object that is more suitable for recalculation of user interests than inference engine constructs (see Section 6.5 for discussion). In our inference engine (Jess), it can be simply bound to a global variable with the following statement:

```

(import ca.sfu.iat.um.*)
(defglobal
  ?*user-model-concepts* = (new ca.sfu.iat.um.UserModelConcepts))

```

4.2. Generalization of user model for semantic web applications

When designing a user model for ec(h)o we considered other application domains where the user model is needed. Another active research area of our lab is eLearning, specifically intelligent support to learners and automatic just-in-time assembly of learning material. A core part of the user model is maintaining user interests that also reappear in other contexts either directly as user interests or as user knowledge, abilities, skills, etc. Recognizing many similarities between requirements from ec(h)o and eLearning domains, we

have designed our user model in a modular fashion that benefits from two easily scalable technologies: ontologies and rule-based systems. Fig. 4 shows the generalized flow of processing that keeps track of user interests with generic parts in bold.

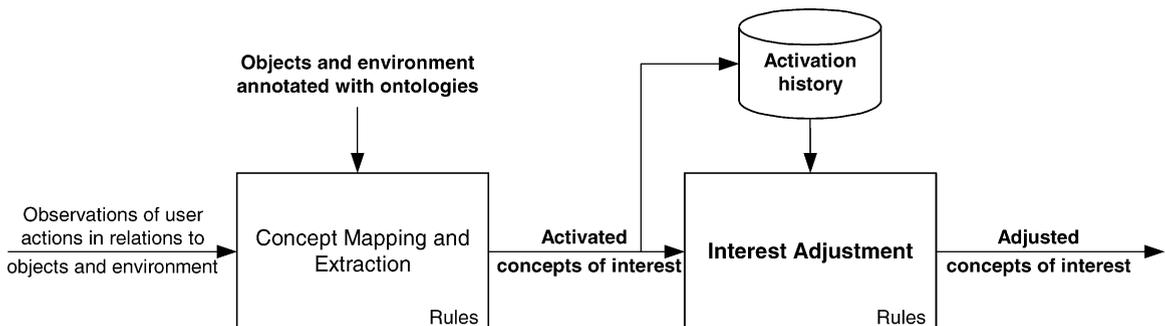


Fig. 4. Part of user model responsible for interest adjustment (generic components are in bold).

with these concepts. For example, when a user selects a sound object annotated with primary and secondary concepts of interest, the system extracts these two concepts and assigns them two different levels of engagement ('activated concepts of interest' link in Fig. 4).

As the name suggests, the Interest Adjustment block is responsible for adjusting the user interest as a reaction to user actions. In our design, this is a generic component that has two parameters: maximum level of individual interest, and a maximum for a sum of all interests. Based on a set of activated concepts and previous values for interest, the algorithm re-computes the values accordingly. Both components are implemented as rule sets and therefore the model can be easily adapted to other applications.

5. Inference-based sound object retrieval

We have identified the following requirements for the retrieval of appropriate sound objects:

1. Content-relevant to the viewed artifact;
2. Content-relevant to the user interests;

```
(defrule c1---1
  (user-group (user ?u) (group 1))
  (replace (user ?u) (context ?e) (object ?in1&:(neq ?in1 nil))
    (sequence ?seq) (time-chosen ?t))
  (user-concept (user ?u) (concept ?c) (level ?i) (time ?t))
  (object-concept (object ?in2) (concept ?c) (level ?n))
  (is-about ?in2 ?e)
  (not (replaced (user ?u) (next-object ?in2) ))
  (test (neq ?in1 ?in2))
  =>
  (bind ?r (* ?i ?n))
  (call ?*object-ratings* addRating ?u ?in1 ?in2 ?r ?t))
```

3. Content invites to exploration of other areas;
4. Content is plausible from the psychoacoustics perspective.

In addition to the criteria for an individual object the following criteria apply to the sequence of the objects offered to the user:

```
(defrule concept-level-c1
  (PropertyValue & psch;#hasPrimaryConcept ?in ?c)
  =>
  (assert (object-concept (object ?in) (concept ?c) (level 1))))

(defrule concept-level-c2
  (PropertyValue & psch;#hasSecondaryConcept ?in ?c)
  =>
  (assert (object-concept (object ?in) (concept ?c) (level 0.5))))
```

5. Provide for exploration of a subject in depth;
6. Provide for the fluidity in experience both in content and sound experience;
7. Provide a mix of informational and entertaining objects.

The retrieval process in ec(h)o can be broken into several steps. The input into the process is user interests, interaction history and semantic descriptions of sound objects. In the process the criteria listed above contribute to overall ranking for each sound object.

The following rule `c1---1` contributes to the rating of object `?in2`. The object `?in2` is a candidate object to replace previously listened to object `?in1` (represented by the `replace` fact). The object `?in2` is a candidate because it matches the concept of user interest `?c` (fact `user-concept`) within the context of the current exhibition `?e` (fact `is-about`). The object rating is a combination of level of user interest in the concept and level by which the concept is represented by the sound object. The rating is added to the `?*object-ratings*` java object (see discussion in Section 6.5).

The `object-concept` facts were created from the semantic representation using rules below. These facts also include different levels for primary and secondary concepts (rules `concept-level-c1` and `concept-level-c2`):

The `?*object-ratings*` is bound to a Java object that simplifies the calculation of object ratings:

```
(import ca.sfu.iat.um.*)
(defglobal ?*object-ratings* = (new ca.sfu.iat.um.ObjectRatings))
```

The composition criteria considers the next object in the context of the objects the user listened to previously. The selection is based on theme, topic, concepts, and described artifacts. An example of such rules is a rule that increases the rating of the sound objects that conti-

```
(defrule guide1---1
  (user-group (user ?u) (group 1))
  (replace (user ?u) (context ?e) (object ?in1&:(neq ?in1 nil))
    (sequence ?seq) (time-chosen ?t))
  (PropertyValue &psch;#hasInfoCategory ?in1 &psch;#Guide)
  (PropertyValue &psch;#describes ?in1 ?a1)
  (PropertyValue &psch;#hasInfoCategory ?in2 &psch;#Guide)
  (is-about ?in2 ?e)
  (test (neq ?in1 ?in2))
  (not (replaced (user ?u) (next-object ?in2) ))
  ..(not (breaks-logical-ordering (user ?u) (object ?in2)))
  =>
  (call ?*object-ratings* addRating ?u ?in1 ?in2 ?*guide-in-rating*
    ?t))
```

nue to provide more information about an artifact described by the previous selected sound object. The rule `artifact2artifact---1` below adds ratings to

```
(defrule calculate-best-object---1
  (declare (salience -20))
  (user-group (user ?u) (group 1))
  (replace (user ?u) (object ?in1) (order ?n) (time-offered ?t1)
    (context ?e) (time-chosen ?t2) (sequence ?seq) (category ?c) )
  =>
  (bind ?best
    (sym-cat (call ?*object-ratings* getBest user1 ?in ?t2)))
  (assert (best-next ?u ?in1 ?e ?t2 ?seq ?best)))
```

the sound object that describes the same artifact as the object being replaced. The rule checks whether candidate object `?in2` describes the same artifact `?a` as previous object `?in1` while `?in2` cannot be an exhibition object but an actual artifact within the exhibition.

```
(defrule artifact2artifact---1
  (user-group (user ?u) (group 1))
  (replace (user ?u) (context ?e) (object ?in1) (context ?e)
    (sequence ?seq) (time-chosen ?t))
  (test (neq ?in1 nil))
  (in-context ?a ?e)
  (PropertyValue &psch;#describes ?in1 ?a)
  (PropertyValue &psch;#describes ?in2 ?a)
  (not (PropertyValue &rdf;#type ?a &crm;#exhibition))
  (not (replaced (user ?u) (next-object ?in2) ))
  =>
  (call ?*object-ratings* addRating ?u ?in1 ?in2 ?*artifact-rating* ?t))
```

Another rule supporting ec(h)o's interaction model is the rule `guide1---1` that favors objects annotated as a guide sound object² after a previous guide object

² Guide objects provide information that is specific to a particular artifact. The guide sound objects are still designed to be independent,

was offered for a particular artifact. This allows system to keep focus on the artifact. As guide objects are related to specific artifacts the rule makes sure that logical ordering between two consecutive sound objects is not violated.

When all the rules contributing to the ratings of sound objects are applied the object with highest rating is selected to replace the object user listened to (rule `calculate-best-object---1`).

The salience value in the rule `calculate-best-object---1` guarantees that the rule is applied after all rules contributing to the ratings of sound objects replacing this particular sound object `?in1`.

however in certain cases it is not desirable to offer some guide objects once the user listened to other guide objects. This is prevented by explicitly specifying such undesirable ordering. Second type of objects are expert objects that provide more generic information applicable across several exhibitions, e.g. sound objects describing relation between evolution and diversity.

For more details of information retrieval aspects inec(h)o see [11].

6. Implementation

The ec(h)o system was fully implemented, deployed, and tested in the setting of the real exhibition space in Nature Museum in Ottawa in March 2004. The system used radio frequency based position tracking system with an update rate of up to 1.6 seconds. The vision and audio delivery systems were developed in our lab in the MAX/MSP environment.

The reasoning module is fully implemented with all features described in the previous section. During the development we embedded the reasoning engine in the Tomcat environment in order to facilitate on-line editing of knowledge models as shown in Fig. 5. However, for the final deployment we removed the reasoning engine from the Tomcat environment for the performance reasons. All communication with the reasoning engine was accomplished through a UDP connection.

6.1. Reasoning engine implementation

The real-time nature of the ec(h)o environment was the driving force for the selection of the implementation platform that would support the reasoning engine. As

shown in Fig. 5, the Jess inference engine is in the center of the reasoning module. We have used DAMLJessKB to load DAML + OIL ontologies into Jess (for details see [13]). DAMLJessKB uses Jena toolkit to convert ontologies into RDF triples which are converted to Jess facts (see examples in Section 3). When converted, ontologies are loaded into the Jess; the rules representing DAML + OIL semantics (provided by DAMLJessKB) infer all the missing relations in the RDF graph. This happens at the start time and prepares the system to respond to the input in a real-time fashion. However, this nice theoretical assumption was challenged by the reality of our implementation, which we summarize in the following sections.

6.2. Memory requirements of ontological representations

Ec(h)o makes use of several ontologies that need to be loaded into the Jess knowledge base. Table 3 summarizes the number of classes, properties, and instances for each ontology used in ec(h)o.

During the loading process the ontologies are converted into RDF triples and the full DAML + OIL semantics is applied, generating complete RDF tree for the knowledge models. Table 4 shows the number of triples for ontology models only and then for ontology models and instances before and after applying semantic rules.

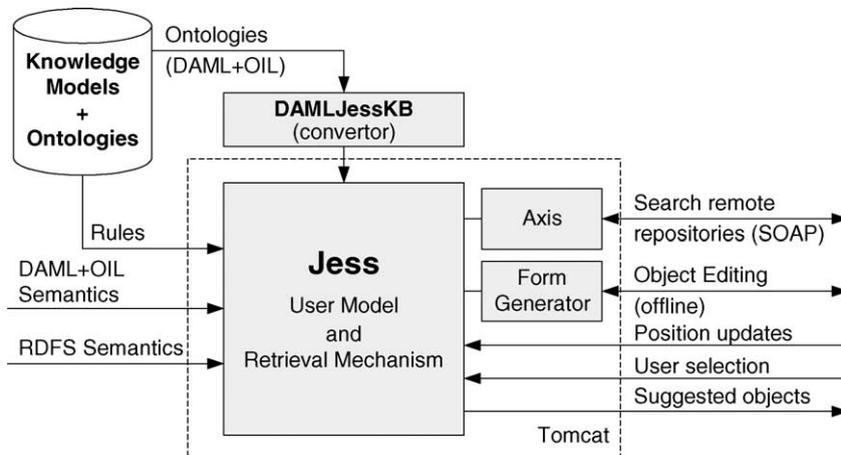


Fig. 5. Implementation schema of the reasoning module.

a fact. First, the system updates user and environment models, then proceeds with the ranking of objects considering updated user and environment models; finally, it applies the interaction criteria to select next recommended objects. To achieve the sequencing we had to prioritize between groups of rules using salience values which consequently had some undesirable effects. We describe particular challenges and lessons learned in the section below.

6.4. Performance

The final implementation of the reasoning engine ran on a Pentium M 1.5 GHz with 768 MB of RAM. The final demonstration served two concurrent users (of maximum four possible). The reasoning engine received input about the location of each user approximately every two seconds. This input caused a short 50% spike in processor activity when the user moved within the same exhibit and a short 100% spike when the user changed exhibits. After receiving input about user selection of a sound object, the processor performance briefly reached 100% and completed the selection of a new sound object below the 1 s limit (this was well below the time the user actually listened to sound objects, which was typically 5–20 s). The memory usage during load time reached above 512 MB and then stabilized around 372 MB (these numbers measure memory used by Java JVM).

The use of a forward chaining inference engine has proved itself to be an efficient mechanism for responding to the dynamic nature of the user input. The system loading time was relatively long as a lot of parsing and initial inference is performed on the ontologies and object descriptions. After the startup phase the amount of inference is limited to updates from the user input, resulting in quick responses.

6.5. Challenges and lessons learned

From the implementation perspective (we will talk about qualitative evaluation in the next section) the reasoning engine had the only criterion: a real time response to other parts of the ec(h)o system. As we developed content incrementally we did most of the reasoning engine design and development with a limited set of 150 sound objects recorded early in the process. As a result some of the challenges

showed up when we scaled up to the full set of over 600 sound objects. Another aspect that challenged us was simultaneous support for multiple users. We discuss some of these challenges that have general implications for similar systems.

6.5.1. Problem 1: rich semantics can cause significant computational delays

6.5.1.1. *Problem.* The rules for selecting sound objects use several criteria for fluency of a dialogue. The criteria depend on ontological annotation of themes, topics, concepts, etc. With richly annotated objects the system was not able to select new sound objects in real-time.

6.5.1.2. *Cause.* Different criteria are represented by individual rules and when fired they contribute a value towards the final score for the objects. Some criteria are satisfied for many sound objects. For example, the criterion that keeps coherency of theme in the dialogue is activated many times as all sound objects are categorized only into seven themes, which are present in the exhibition. The criterion itself has little decisive power but consumes many resources.

6.5.1.3. *Solution.* After we analyzed results from the preliminary user testing we eliminated some of the rules/criteria. This had a minimal impact on the quality of the end user experience and significantly reduced the number of rule activations. In general, the semantic annotation that categorizes an object in a coarse manner should not be used in a generative computation but rather used for filtering out of unsuitable candidates.

6.5.2. Problem 2: concurrency has to be treated explicitly

6.5.2.1. *Problem.* In the case of concurrent users, the reasoning modules waits until sound objects for all users are computed and delivers all of them at the same time. This caused significant latency for individual users.

6.5.2.2. *Cause.* In ec(h)o we had to work with salience values (rules with higher salience value fire before rules with a lower value). In the case of multiple users, the rules interfered with each other. For example, if a second user makes a choice before a computation for the first user is finished then rules with a higher salience

for the second user start firing. This causes computation for the first user to be pending until all the rules for the second user with higher salience values fire. With an increasing number of visitors the latency increased.

6.5.2.3. Solution. We found the solution with help from the Jess community. We categorized the users into groups and assigned an identical set of critical rules for each group. The set of rules is activated only for users belonging to the group, so the users from different groups do not block each other. In general, the same problem can occur when the reasoning engine is exposed as a web service and a multiple access to the service is allowed.

6.5.3. Problem 3: know-your-tool or carefully consider implications of implementation platform

6.5.3.1. Problem. A rule that gets activated many times with a ‘not’ clause positioned early on the list of preconditions takes a long time to fire.

6.5.3.2. Cause. ‘not’ Pattern can only match an absence of a fact. In our case, it is evaluated only when the fact is asserted (then it fails) or when the pattern immediately before the ‘not’ clause on the rule left hand side is evaluated. Therefore patterns following the ‘not’ clause are evaluated at the runtime. Combining this with a large number of candidate facts resulting from the ontology representations causes significant delays.

6.5.3.3. Solution. Position a ‘not’ clause as the last pattern on the left hand side of the rule.

6.5.4. Problem 4: do not use rules for extensive numerical computations

6.5.4.1. Problem. Computing multi-criteria numerical preferences required assertion of extensive number of facts and use of salience values resulting in growing response times for subsequent iterations.

6.5.4.2. Cause. As several criteria are used to contribute preference values to the overall score of each sound objects, we need a mechanism ensuring that all contributions are made before making sound object selection decision. There are two possible approaches: first, add all the contributions as facts and then fire summation rule; or, keep adding contribution to one fact, which means retracting and re-asserting it into the knowledge base. The second approach is more time consuming. Both approaches require use of salience values to make sure all contributions were made.

6.5.4.3. Solution. Build a simple extension in Java (or other language) that will perform the computation and make it accessible through the inference engine extension mechanism (direct call to Java in the case of Jess). This will speed up computation as generating large volume of facts and build up of the Rete network for rule activations will be avoided. The salience will still be

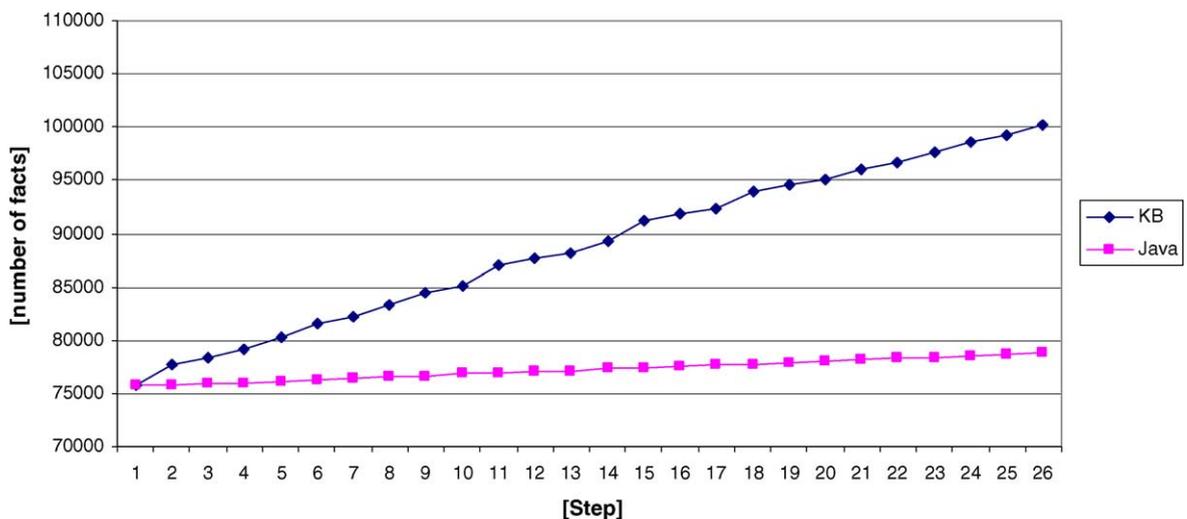


Fig. 7. Number of facts in iteration steps.

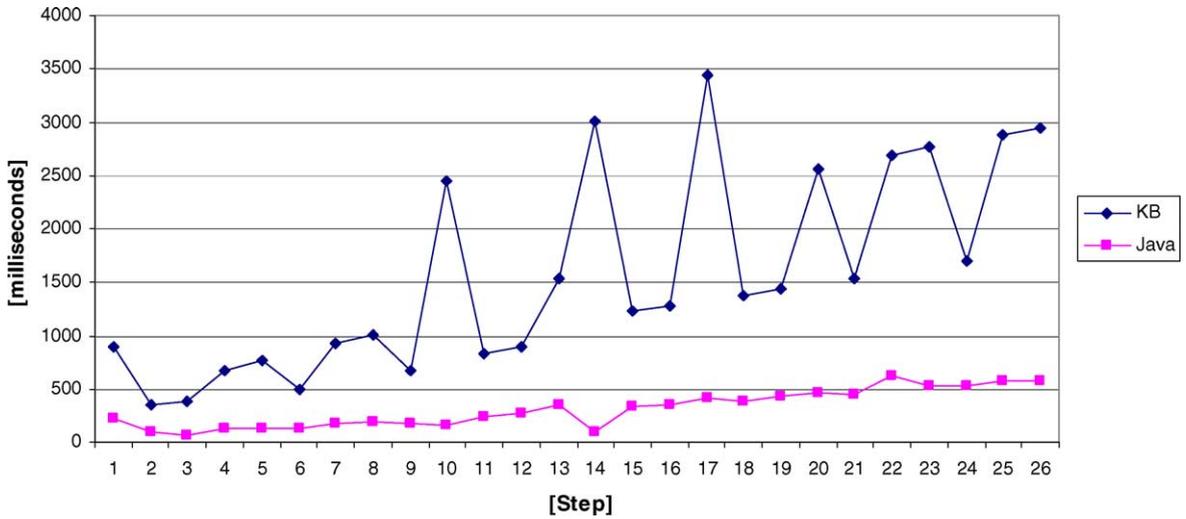


Fig. 8. Response time in iteration steps.

needed to ensure that all contributions were made. The rules in Section 4.1 illustrate the solution. Figs. 7 and 8 show the effect of moving computation from the knowledge base to the external Java module.

7. Evaluation and discussion

ec(h)o is a complex interdisciplinary research project that has to be evaluated from different perspectives. As the evaluation of ubiquitous computing systems is extremely complex [20] we have found Miller’s and Funk’s [14] view of the problem of evaluation of ubiquitous computing systems from the traditional ‘validation’ and ‘verification’ perspective very useful. In validation we evaluate whether the system performs the functions it was built for based on the requirements specification. Verification tests the system against the reality by checking whether the system provides the envisioned benefits. Finally, the evaluation of technical aspects of the system implementation can provide insights to the developers of a similar system.

Following Miller’s and Funk’s approach allowed us to focus our evaluation on the areas where we researched novel approaches in the adaptive ubiquitous systems. We also avoided the evaluation of the aspects of the system that are not well defined or understood and the evaluation results would provide very little value.

Our validation efforts concentrated on the system components for which we either had predicted

outcomes or have established the criteria for such outcomes. Specifically, we have validated the flexibility and responsiveness of the user model and effectiveness of the object recommendation component. We have verified our solution with the targeted end user group through extensive questionnaires and videotaped interviews.

In this section we provide an overview of the evaluation results as those are reported in detail elsewhere [10]. A detail account is given for the evaluation aspects related to rules and ontologies.

7.1. Suitability of ontologies and rules for user modeling

In the context of our work, the user model performs a function of a recommender system [15]. “Recommender systems represent user preferences for the purpose of suggesting items to purchase or examine” [6]. Several types of recommendation techniques have been developed: collaborative, content-based, demographic, utility based, and knowledge-based. Often the researchers combine several techniques to achieve maximum effect. The knowledge-based recommender systems perform favorably with respect to the introduction of new users and new items (so called ‘ramp-up’ problem [12]) which is an important feature for ubiquitous computing environments. The knowledge recommender systems require three types of knowledge [6]: catalog knowledge or knowledge about objects to

be recommended, functional knowledge of mapping between user needs and objects, and user knowledge.

From this perspective we have used ontologies extensively to describe knowledge about objects, environment, and the user. As multiple criteria were used to determine the user interests, a rule-based approach provided us with the flexibility that enabled us to evolve the system through several iterations. Furthermore, to be able to respond to the specifics of the application we have parameterized the influence of inputs from the user and ubiquitous environment such as maximum interest value, object selection, and location change contributions towards user’s interests, etc. The purpose of the parameterization was to fine-tune our generic user model framework. We performed an extensive testing for the suitable combination of parameters in the lab setting with early input from the test users.

The user model uses a spring model to keep interests balanced. The level of interest is represented by the real

number and can range from 0 to 10 (the value was set with respect to other values used for ranking objects). The sum of all interests never exceeds the value of 30. In the model we consider only positive influence from the user interaction that directly increases the level of some of the interests. When this increase causes an imbalance (the sum is above 30), the implemented spring model proportionally decreases values of other interests.

Fig. 9 show the sequence of steps and evolution of interests in each step. In the first step three concepts are selected by the user. The circle icon indicates concepts introduced to the model by the visually represented exhibit concepts (Steps 2, 11, and 15). In the rest of the steps the user selected sound objects. The square icon indicates primary concept and triangle icon secondary concept in the selected sound object.

The rule-based model proved to be very flexible and responsive to the parameters. The representation of the knowledge in the form of ontologies made the

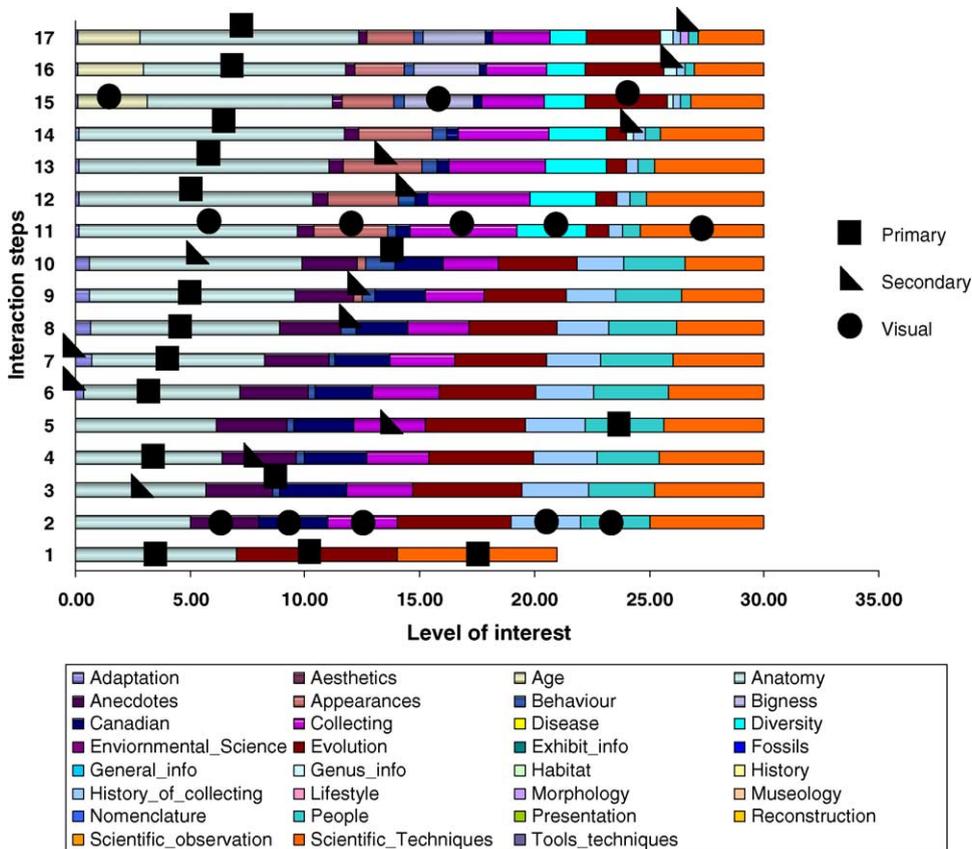


Fig. 9. Evolution of user interests.

design and implementation of the model very easy with the clear way of accessing the knowledge. The use of the DAMLJessKB module accompanied with the DAML + OIL language semantics made the inference in the knowledge base transparent, which enabled us to concentrate on the model implementation instead on navigating and inferring static knowledge.

7.2. End user verification

As Miller and Funk [14] point out the verification evaluates the system from the perspective of provided value. Typically, the qualitative methods are used and end user testing is involved. The qualitative methods are more suitable for novel approaches and new areas of research to verify the potential of those.

In ec(h)o we have conducted in depth usability testing of the system while deployed in the real museum setting. An extensive testing was done with 6 subjects. The subjects were briefly trained on how to use the system (learning phase), and then had an opportunity to ask questions. They used the system on their own for a period of 10–20 min. After this session, they completed a modified version of Ben Schneiderman’s acceptance test. Finally, we conducted and videotaped interviews with the subjects. In addition to those tests, we had one museum expert evaluating the content side of the system in depth.

The overall use of the system was rated relatively high. For example, when asked to rank between 1 and 5 on a Likert scale (5 being best) over five different questions relating to the overall reaction to the system, the averaged response was 3.6. The evaluation scored 4.6 for ease of use and 2.8 for satisfaction. Navigation and engagement of the audio information rated high; for example, appropriateness of the audio experience scored 4.0. This leads us to believe that the system meets or satisfies many of the current advances of electronic guide systems. Participants were explicitly asked to compare the system to experiences with other systems and the prototype ranked favorably.

Difficulties exist in relating sound objects to a specific artifact. In certain cases visitors didn’t mind the ambiguity while others clearly found it frustrating. The results also differ in the ‘attitude’ related questions. Some users had strong feelings about their preferred modes of interaction; others approached the system from the more playful perspective.

It is difficult to draw conclusions from the number of testers we had. The expert reviews were strongly in favor of the approach and the system. The reviews were helpful in catching potential inconsistencies and challenges.

Hatala and Wakkary [10] provides more detailed discussion on the ec(h)o evaluation results from the user modeling perspective.

7.3. Efficiency of ontologies and rules for ubiquitous real-time applications

As the implementation section already presented concrete results and lessons learned from using ontologies and forward chaining rules in ec(h)o, in this section we summarize the outcomes and highlight a potential of used technologies for the realtime applications.

The ec(h)o implementation was based on technologies that were available, stable, and supported by tools in 2003. W3C’s Ontology Web Language has since superseded the DAML + OIL ontology language. This would be our candidate language if we were developing the systems now.

The representation of DAML + OIL (or OWL) ontologies in the forward chaining system knowledge base reflects their RDF representation in the form of triples. This form of representation creates an enormous number of syntactically similar facts resulting in potential performance problems. However, these problems can be overcome by using unordered facts [13]. A major benefit for the real-time systems is that the inference applies ontology language semantics at startup time, inferring the full graph representing all existing relations. During the runtime only relations with newly created instances are inferred resulting in speedy updates to the system. From the developer’s perspective the uniformity of the representation and availability of the full relation graph makes it easier to develop rules referring to the ontologies and properties between objects.

There are many best practices available for writing forward-chaining rule systems. With the large number of syntactically uniform facts some of the recommendations need to be observed rigorously otherwise resulting in a big performance hit. A good knowledge of underpinnings of the inference system is needed (in our case a Rete network and algorithm) particularly about ordering facts in the precondition part of the rules and

using the not clauses in the rules. Also, carefully considering the delegation of certain tasks such as numerical computation to the external modules can improve the performance significantly.

One specific aspect of the multi-user real-time application that we were not able to resolve satisfactorily is the possible collision of rules for individual users. The problem occurs when the salience values are used to sequence processing steps. Our approach grouped users and assigned them their own rule sets so the users from different groups did not collide. A more robust solution would call for the dynamic creation of modules for each user with the full management of these modules to avoid exhausting of the system resources.

Another related effort in the Semantic Web community in the area of rules is Rule Markup Language (RuleML) aiming at interoperability between inference environments. However, we have not considered the RuleML since other requirements such as performance had a priority over the interoperability. We also wanted to benefit from the ability to experiment with and extend our selected inference engine.

8. Conclusions

In this paper we have presented the design and implementation of an augmented audio reality system for museum visitors named ec(h)o. Each visitor experience is tailored to the visitor's interests. The user interests are inferred from the user's movement through the exhibition as well as from the visitor's interaction with the sound objects. The sound objects are retrieved based on their relevance to the user interests, narrative criteria, and psychoacoustic criteria. ec(h)o uses ontologies to describe concepts, temporal and spatial characteristics, and psychoacoustic and sound characteristics of sound objects. In the core of the system is a rule-based inference engine that powers the retrieval mechanism and the user model specifically designed for the applications using rich semantic descriptions.

The system is a result of convergent research streams from research in object repositories, interaction design, auditory display, knowledge representation, and information retrieval. The ontologies combined with the rule-based inference proved to be a powerful implementation platform well suited for this type of the systems. We believe this has enabled us to extend works

cited through the paper in several directions. First, it extends the work of the Alfaro et al. work [1] by building a rich model of the concepts represented by the sound objects. In ec(h)o, the content presented to the user is not pre-processed for possible linkages as in the systems using Rhetorical Structure Theory [24]. Our approach replaces pre-processed linkages with a retrieval mechanism based on composition and interaction criteria formulated in the form of the rules and applied to semantically-annotated independent objects.

The requirements of the real-time ubiquitous application required us to face the challenges stemming from the combination of two powerful technologies: ontologies and forward-chaining rules. We have summarized our findings in the lessons learned that provide a transferable generic knowledge for similar type of application. The ec(h)o proved that ontologies and rules provide an excellent platform for building a highly-responsive context-aware interactive application.

Acknowledgements

Work presented in this paper is supported by a Canarie Inc. grant under the E-Content program. The authors would especially like to thank Mark Graham and his colleagues in the Nature Museum in Ottawa for their enthusiastic support of this project. We would also like to thank our colleagues and participants in several workshops who contributed to the development of the project, namely Kenneth Newby, Dale Evernden, Doreen Leo, Gilly Mah, Robb Lowell, Mark Brady, Jordan Willms, and Phil Thomson.

References

- [1] I. Alfaro, M. Zancanaro, M. Nardon, A. Guerzoni, Navigating by knowledge, in: Proceedings of the Eighth International Conference on Intelligent User Interfaces, 2003, pp. 221–223.
- [2] D. Andolesk, M. Freedman, Artifact as Inspiration: Using Existing Collections and Management Systems to Inform And Create New Narrative Structures. 2001, in: Archives and Museum Informatics Museums and the Web 2001, 2001.
- [3] B.B. Bederson, Audio augmented reality: a prototype automated tour guide, in: Conference Companion on Human Factors in Computing Systems, 1995, pp. 210–211.
- [4] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Sci. Am. (2001).
- [5] H. Boley, S. Tabet, G. Wagner, Design rationale of RuleML: a markup language for semantic Web rules, in: International Semantic Web Working Symposium, 2001.

- [6] R. Burke, Hybrid recommender systems: survey and experiments, *User Model. User-Adapt. Interact.* 12 (2002) 331–370.
- [7] N. Crofts, I. Dionissiadou, M. Doerr, M. Stiff, Definition of the CIDOC Object-Oriented Conceptual Reference Model, July 2001, 2001.
- [8] DDC, Dewey Decimal Classification, <http://www.oclc.org/dewey>.
- [9] D. Dean, *Museum Exhibition: Theory and Practice*, Routledge, London, 1994.
- [10] M. Hatala, R. Wakkary, User Modeling and Semantic Technologies in Support of a Tangible Interface Submitted to UMUI, November 2004, pre-print version available from <http://www.sfu.ca/~mhatala/pubs/>.
- [11] M. Hatala, L. Kalantari, R. Wakkary, K. Newby, Ontology and rule based retrieval of sound objects in augmented audio reality system for museum visitors, in: *Proceedings of the 2004 ACM Symposium on Applied Computing*, 2004, pp. 1045–1050.
- [12] J.A. Konstan, J. Riedl, A. Borchers, J.L. Herlocker, Recommender Systems: A GroupLens Perspective, in: *Recommender Systems: Papers from the 1998 Workshop*, 1998, pp. 60–64.
- [13] J. Kopena, W.C. Regli, DAML Jess KB: a tool for reasoning with the semantic web, *IEEE Intell. Syst.* 18 (2003) 74–77.
- [14] C. Miller, H.B. Funk, Verification Through User value: Or How to Avoid Drinking Your Own Bathwater in Ubicomp Evaluations, in: *Ubicomp'01 Workshop: Evaluation Methodologies for Ubiquitous Computing*.
- [15] P. Resnick, H.R. Varian, Recommender Systems, *Commun. ACM* 40 (1997) 56–58.
- [16] M. Sarini, C. Strapparava, Building a User Model for a Museum Exploration and Information-Providing Adaptive System, in: *Second Workshop on Adaptive Hypertext and Hypermedia at HYPERTEXT'98*, 1998.
- [17] C. Schlieder, T. Voegelé, A. Werner, Location Modeling for Intentional Behavior in Spatial Partonomies, in: *Ubicomp 2001 Workshop on Location Modeling for Ubiquitous Computing*, 2001, pp. 63–70.
- [18] B. Serrell, in: S. Bitgood (Ed.), *The Question of Visitor Styles, Visitor Studies: Theory, Research and Practice*, vol. 7.1, Visitor Studies Association, Jacksonville, AL, 1996, pp. 48–53.
- [19] F. Sparacino, The museum wearable: real-time sense-driven understanding of visitors' interests for personalized visually-augmented museum experiences, in: *Museums and the Web*, 2002.
- [20] M. Spasojevic, T. Kindberg, Evaluating the CoolTown user experience, in: *Ubicomp'01 Workshop: Evaluation Methodologies for Ubiquitous Computing*, 2001.
- [21] W3C, "OWL: Ontology Web Language," 2003. <http://www.w3.org/2001/sw/WebOnt/>.
- [22] W. Wahlster, A. Kobsa, in: A. Kobsa, W. Wahlster (Eds.), *User Models in Dialog Systems, User Models in Dialog Systems*, Springer-Verlag, New York, Inc., 1989.
- [23] R. Wakkary, K. Newby, M. Hatala, D. Evernden, M. Droumeva, in: D. Bearman, J. Trant (Eds.), *Interactive Audio Content: The Use of Audio for a Dynamic Museum Experience through Augmented Audio Reality and Adaptive Information Retrieval, Museums and the Web 2004 Selected Papers*, Arlington, Virginia, 2004, pp. 55–60.

- [24] M. Zancaro, O. Stock, I. Alfaro, Using cinematic techniques in a multimedia museum guide, in: *Museums and the Web*, 2003.



Dr Marek Hatala is an Assistant Professor at School Interactive Arts and Technology and a director of Laboratory for Ontological Research at Simon Fraser University Surrey in British Columbia, Canada. Before this appointment Dr Hatala was a Research Fellow at the Knowledge Media Institute at The Open University, UK and an Assistant Professor at The Technical University of Kosice, Slovakia. Dr Hatala holds his PhD in Cybernetics and Artificial Intelligence.

His main research interest areas are knowledge representation, ontologies and semantic web, user modeling, intelligent information retrieval, organizational learning and eLearning. In the last four years Dr Hatala focused on building a scalable distributed infrastructure for e-learning systems and middleware enabling interoperability between heterogeneous repositories. The infrastructure connects repositories in Canada, US, UK, Australia and Europe. His current research projects look at how semantic technologies can be applied to achieve interoperability in highly distributed and heterogeneous environments, what are the social and technical aspects of building a distributed trust infrastructures, and what role the user and user group modeling can play in interactive and ubiquitous environments.



Ron Wakkary is Associate Professor in the School of Interactive Arts & Technology at Simon Fraser University in British Columbia. His research includes interaction design, and design methods and practice for interactive systems. He has served on the Canadian Culture Online Advisory Board to advise the Minister of Canadian Heritage, and is on the Banff New Media Institute Research Advisory Board. He has lead design and technology research projects for

the Canadian Nature Museum, Nokia Research, Museum of Modern Art, the Guggenheim Museum, and Electronic Arts Intermix. He is currently leading the Am-I-able Network for Responsive and Mobile Environments, a research network funded by Canadian Heritage, and research on the everyday complexity of design funded by the Social Sciences and Humanities Research Council.

Leila Kalantari is pursuing a master's degree at School of Computing Science, Simon Fraser University, British Columbia. During her undergraduate studies at SFU she worked as Research Assistant for Computational Logic Lab, Intelligent Systems Lab and Laboratory for Ontological Research.

