Conferentie informatiewetenschap 2003, Technische Universiteit Eindhoven, 20 november 2003 : proceedings

WGI TU/e technische universiteit eindhoven SIKS

Conferentie Informatiewetenschap 2003

Technische Universiteit Eindhoven
20 november 2003

Proceedings

edited by P. De Bra

De negende Interdisciplinaire Conferentie Informatiewetenschap 2003 is georganiseerd door de Technische Universiteit Eindhoven in opdracht van de Werkgemeenschap Informatiewetenschap en in samenwerking met de Onderzoekschool SIKS.

De conferentie Informatiewetenschap heeft tot doel het bijeenbrengen van onderzoekers, deskundigen, probleemeigenaren en geïnteresseerden op het vakgebied "informatiewetenschap". De conferentie is **het** ontmoetingspunt bij uitstek voor leden van de werkgemeenschap (en andere geïnteresseerden) omdat het een uniek forum is waar onderzoekers op het gebied van de informatie zelf en onderzoekers op het gebied van de technologie om met informatie om te gaan hun werk presenteren en bediscussiëren.

De conferentie is geopend door prof. dr. Theo Huibers (KPMG en Universiteit Twente), met een keynote "Information Retrieval, Wat Vindt U?". De tekst van deze lezing is niet in deze proceedings opgenomen.

Programmacommissie

Antal van den Bosch (Universiteit van Tilburg)
Crit Cremers (Universiteit Leiden)
Paul De Bra (Technische Universiteit Eindhoven)
Peter Doorn (Nederlands Instituut voor Wetenschappelijke Informatiediensten NIWI)
Lynda Hardman (Centrum voor Wiskunde en Informatica)
Ad Van Heijst (Van Heijst Consulting)
Geert-Jan Houben (Technische Universiteit Eindhoven)
Theo Huibers (KPMG en Universiteit Twente)
Kees van der Meer (Technische Universiteit Delft)
Aldo de Moor (Universiteit van Tilburg)
Paul Nieuwenhuizen (Vrije Universiteit Brussel)
Eric Postma (Universiteit Maastricht)
Guus Schreiber (Vrije Universiteit)
Egbert De Smet (Universiteit Antwerpen)
Gerrit van der Veer (Vrije Universiteit)
Arjen de Vries (Centrum voor Wiskunde en Informatica)
Theo van der Weide (Katholieke Universiteit Nijmegen)

Sequence and Emphasis in Automated Domain-Independent Discourse Generation

Martin Alberink Telematica Instituut
P.O. Box 589 NL-7500 AN Enschede The Netherlands
+31 53 4850485 Martin.Alberink@telin.nl

Lloyd Rutledge CWI
P.O. Box 94079 NL-1090 GB Amsterdam The Netherlands
+31 20 5924127 Lloyd.Rutledge@cwi.nl

Mettina Veenstra Telematica Instituut
P.O. Box 589 NL-7500 AN Enschede The Netherlands
+31 53 4850485 Mettina.Veenstra@telin.nl

ABSTRACT

For humans to gain comprehensive views of large amounts of repository contents, they need to have insight into the relations among information objects. It is a challenge to automatically generate presentations of repository contents, through, for example, search results, which reveal such relations to readers. Such presentations must reflect properties of information objects such that large sets of information objects appear as a coherent whole. An approach to this is generation of discourse structures that convey such properties of information objects in presentations. Semantic Web technology provides a conceptual basis for generation of discourse in Web-based information environments.
This paper describes automatic generation of sequence and emphasis in presentations of information objects. It shows generation of object sequences and emphasis in accordance with a user input of relevance of information attributes in our Topia architecture. The resulting presentations allow users to encounter information objects in decreasing order of relevance. This makes it easier to identify relevant information objects among many others, as well as to observe their relations with the other information objects.

Categories and Subject Descriptors
H.5.4, H.5.1 [Information Interfaces and Presentation (e.g., HCI)]: Hypertext/Hypermedia architectures, navigation; Multimedia Information Systems Hypertext navigation and maps, Evaluation/methodology; I.7.2 [Document and Text Processing]: Document Preparation Hypertext/hypermedia, Markup languages, Multi/mixed media, standards.

General Terms
Algorithms, Documentation, Design, Experimentation, Standardization.

Keywords
Discourse, Narrative, Coherence, Semantics, Sequence, Order, Emphasis, Hypermedia, Cluster, Semantic Web, RDF.

1. INTRODUCTION

Search engines on the Web typically generate presentations of retrieval results as plain lists of links to information objects, possibly sorted according to relevance or hyperlink connectivity. Such presentations do not easily allow users to assess sets of retrieved information objects as a whole, since this requires that users inspect the retrieved objects one by one. This inhibits users from readily identifying the information objects that are relevant for their information need. Structuring sets of repository contents in coherent presentations, taking preferences of individual users into account, would supposedly facilitate users orientation in presentations of large amounts of information objects. Semantics of electronic content on the Web encoded in Semantic Web technology [7] provide a basis for deriving relations among information objects in Web-based information environments. Such relations, when included, could add to coherence in presentations of information objects.

Our focus is on automated generation of coherent presentations of database contents in order to allow users to find their way in large information collections. We aim at enhancing coherence in presentations of sets of information objects by transforming semantics encoded in RDF into constructs of common discourse that are meaningful for human users. Well-known discourse, such as narrative, conveys relations among information objects in addition to the information itself, such as by means of story lines, sequences, emphasis and focalisation [3]. Automatic production of such rich discourse typically produced by human authors remains elusive. Instead, we aim at generating simple but commonly encountered discourse constructs that can be based on attributes and relations, a typed of semantics supported by the Semantic Web. This papers focus is on automated generation of two such basic discourse constructs in presentations of information objects: sequence and emphasis on information objects in presentations. Sequences show interrelations among a set of objects in a semantic dimension, such as time, place or causality. Emphasising objects conveys a distinct property of such objects with respect to the other elements or a special relation with the other elements. Our presumption is that the resulting presentations of retrieval results enable users to assess the contents and relevance of information objects faster and with less effort compared with the common lists of search results. This, we hypothesize, assists users in deciding on navigation and exploration directions while traversing the information space, in order to help users grasp the contents of information repositories and discover what they find relevant or useful [11].
Section 2 of this paper discusses the approach in this paper in relation to other research. Section 3 describes the Topia (Topic-based Interaction with Archives) project [18] that produced the results described in this paper. Automated generation of sequence and emphasis as discourse constructs in web environments is described in sections 4 and 5 respectively. Section 6 explains involvement of a user statement of relevance of relations in the automated generation of sequence and emphasis in hierarchical presentations of search results. This section shows that the resulting presentations direct readers to the information relevant for them in the search result, while preserving directions to the other retrieved objects. Section 7 shows that the resulting presentations are capable of structuring retrieval results in different perspectives. Sections 8 describes future work on the topic in this paper and section 9 wraps up this paper with a summary and conclusions from this work.

2. RELATED WORK

A number of research projects discussed in this section focus on automated discourse generation in presentations of content stored in digital systems. Their approaches differ in three senses: the balance between human-specified and computer-inferred semantics for discourse generation, the types of discourse constructs that transformation of semantics results in, and the way of presenting discourse structures by conveying relations among information objects. Sections 2.1 through 2.4 position related work in the range from almost completely human specified discourse structures to discourse with high-level human specification only. They also discuss the position of sequence and emphasis in these different approaches.

2.1 Fixed discourse frameworks

Underlying frameworks of automatically generated discourse structures are in the range from nearly fixed to largely flexible. At the one extreme are nearly complete and rigid presentation structures that only leave room for objects to be inserted. Presentations of retrieval results of search engines fall in this category. Such presentations typically contain hyperlinks to retrieved items in straight lists. Application of sequence and emphasis to lists of retrieval results can convey relations among information objects and relations between information objects and information needs of users.

2.2 Template-based discourse

Templates that specify discourse structures are a step provide more flexible discourse frameworks than lists of retrieval results. Gaps in such templates allow insertion of information to fill in the contents of the story. Computer-generated sequence and emphasis in such presentations are bound to the information in each gap. It is important that the information filling the gaps is coherent in its connection with the template. The Artequakt project has a template-based approach focusing on discourse of textual biographies using narrative templates [1]. Sequence and emphasis in textual information is contained in the text itself. For text that originates from natural language generators, the coherence of the text as well as its connection with the template are important sequence criteria.

2.3 Semantics-based discourse

Geurts approach focuses on generation of discourse based on domain knowledge, straight from semantic information [10]. Discourse of specific types can be generated, such as biographies and curricula vitae. Such discourse requires semantics-based sequences and emphasis, since it should be in accordance with their usual contents and structure. Automatically generated sequences in such presentations should be in accordance with user expectations in order to make such presentations coherent.

2.4 Semantics-driven discourse

At the other extreme along the line of discourse framework flexibility is discourse without human-specification of the discourse structure. Such discourse results from characteristics of semantics that abstract from the meaning of the semantics itself but are based on the occurrence of the semantic relations only. Our Topia architecture generates presentations with such discourse structures and applies sequence and emphasis in the discourse. Sequence and emphasis are capable of adding a fraction of the semantics that human authors can generate. They are however universal across application domains. Further in this paper, section 6 explains how sequence and emphasis can be generated such that they are in accordance with relevance of objects for individual users. Section 7 shows that this principle can produce discourse that show information in different perspectives.
Section 3 explains how the Topia architecture derives relations as well as the discourse and presentation structures.

3. TOPIA ARCHITECTURE

The research described in this paper is part of the development of the process architecture of the Topia project. The Topia architecture automates generation of presentation structures of retrieval results with discourse constructs [18]. Figure 1 shows its four phases. The information objects in Topias repository are 740 artefacts from the art collection of the Rijksmuseum Amsterdam [17]. Attributes of the artefacts are encoded in about 64,000 RDF triples.

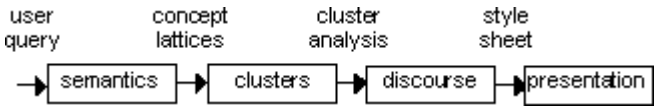


Figure 1. Topia architecture overview [18]

Users access the Topia repository by specifying queries. After retrieving a set of artefacts together with their attributes in the first stage, the second stage generates a concept lattice: a structure of clusters of information objects and the attributes they have in common in a subsumption graph [9]. The third stage transforms clusters and subsumption relations in a concept lattice into a conceptual presentation with discourse constructs. The final stage specifies the layout, the presentation of recurrent themes and the interaction with users in an HTML or SMIL presentation, generated by an XSLT style sheet.

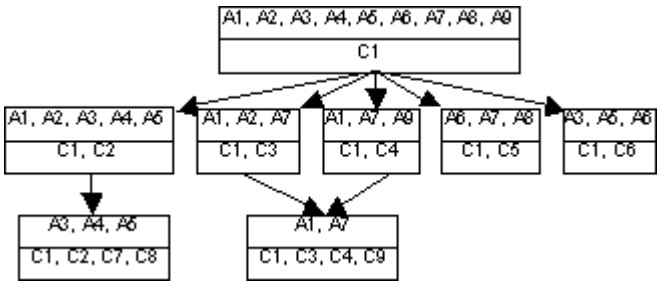


Figure 2. Cluster graph of concept lattice from Table 1 [18]

Table 1. Artefacts mapped against properties in a concept lattice for query on "water" [18]

(C1) "Water"	(C2) Genre: Water, ice and snow	(C3) Genre: Dutch landscapes	(C4) Genre: Field meadows	(C5) Genre: Buildings in landscapes	(C6) Artist: Jacob van Ruisdael	(C7) Genre: Tree forests	(C8) Genre: Riverscapes	(C9) Artist: Paul Joseph Constantin Gabriel
"A watercourse at Abcoude" (A1)	X	X	X					X
"Watercourse near 's-Graveland" (A2)	X	X						
"Mountainous landscape with waterfall" (A3)	X				X	X	X	
"A water mill" (A4)	X					X	X	
"Landscape with waterfall" (A5)	X				X	X	X	
"Water mill" (A6)				X	X			
"Windmill on a polder waterway, known as 'In the month of July'" (A7)		X	X	X				X
"A waterside ruin in Italy" (A8)				X				
"The battle of Waterloo, 18 june 1815" (A9)			X					
Concept Size	**5**	**3**	**3**	**3**	**3**	**3**	**3**	**2**

The concept lattices generated in the second stage of the Topia architecture not only contain all individual artefacts in a retrieval result, but also all clusters of artefacts in a retrieval result that have one or more attributes in common. Each cluster of artefacts together with their common set of attributes is a concept in the concept lattice. Concept lattices subsume concepts under other concepts that contain their smallest supersets of artefacts, in a directed graph [9]. As an illustration, Table 1 shows the retrieval result of the query specifying the string "water" in the title of artefacts. The rows are the titles of the retrieved artefacts and the columns are attributes of one or more of the retrieved artefacts. The crosses in the table indicate the occurrence of the corresponding attribute for the object concerned. Figure 2 partly shows the concept lattice that results for this retrieval result, generated by the Topia architecture. The concepts are the pairs of adjacent bars with the objects printed in the upper bar and the attributes in the lower. The set of common attributes expresses what the relation is among the set of objects in each concept. For example, Figure 2 shows that artefacts A3, A5 and A6 have C1 and C6 as common attributes.

<?xml version="1.0" encoding="UTF-8" ?>
- <sections size="8">
- <section>
<title>material is "Oil on canvas"</title>
- <sections size="4">
- <section>
<title>artist is "Jan Willem Pieneman",
genre is "Battles, Group, Group portraits,
Historical scenes",
theme is "Struggle and Strife",
title is "The Battle of Waterloo, 18 June 1815"
and year is "1824"</title>
- <sections size="1">
- <section>
<title>The Battle of Waterloo, 18 June 1815</title>
<date>1824</date>
<artist>Jan Willem Pieneman</artist>
<image>SK/Org/SK-A-1115.org.jpg</image>
</section>
</sections>
</section>
- <section>
<title>theme is "Netherlands and the Water"
and title is "Landscape with Waterfall"</title>
- <sections size="1">
<section />
</sections>
</section>
- <section>
<title>place is "Den Haag"</title>
- <sections size="2">
<section />
</sections>
</section>
- <section>
<title>title is "Mountainous Landscape
with Waterfall"</title>
- <sections size="1">
<section />
</sections>
</section>
</sections>
</section>
- <section>
<title>genre is "Water, ice and snow"</title>
+ <sections size="3">
</section>
- <section>
<title>place is "Amsterdam"</title>
+ <sections size="2">
</section>
- <section>
<title>genre is "Buildings in landscapes"</title>
+ <sections size="2">
</section>
- <section>
<title>artist is "Jacob van Ruisdael"</title>
+ <sections size="3">
</section>
- <section>
<title>material is "Oil on panel"</title>
+ <sections size="3">
</section>
- <section>
<title>genre is "Dutch landscapes"</title>
+ <sections size="2">
</section>
- <section>
<title>genre is "Fields, meadows"</title>
+ <sections size="2">
</section>
</sections>

Figure 3. Conceptual presentation generated by the Topia demo

Subsumption edges imply a relation between the clusters of artefacts and attributes in the concepts they connect: traversing subsumption edges in an upward direction leads to a more general concept, since such a concept has more objects and fewer attributes than the one traversed from. Likewise, traversing subsumption edges in a downward direction leads to a more specific concept.

The third stage of the Topia architecture generates hierarchical conceptual presentations by flattening the directed acyclic graph structure of concept lattices. Hierarchically organised structures are commonly used backbone structures, such as in books subdivided in chapters, sections and paragraphs, to facilitate orientation by human readers. The conceptual presentations specify the clusters of information objects and relations among the objects by means of the common attributes. Figure 3 shows the conceptual presentation of the retrieval result of the query "water", while Figure 4 shows the presentation of the concept lattice on the screen.

Figure 4. Presentation generated by the Topia demo

Sections 4 and 5 focus on automated generation of sequence and emphasis respectively from universal aspects of semantic annotations. The sections also describe the support that web standards offer.

4. SEQUENCE

Sequences of objects in presentations convey to readers an order of objects along a certain dimension. Consequently, readers of presentations expect sequences of objects to be meaningful so that they have to be in accordance with logical and, if possible, useful sequence criteria. This section discusses organisation of sequences of objects in presentations at the four phases in the presentation generation process, namely semantics, discourse, presentation structure and style.

4.1 Semantics

Semantics imply domain-independent sequences of concepts in information repositories in multiple ways. First, sequences follow directly from explicitly ordered sets of objects. The RDF recommendation contains a <seq> class for explicitly ordered collections, while RDFs <bag> class supports an unordered collection of objects and RDFs <alt> class supports collections of objects that are equivalent in some sense. Second,

sequences follow from the occurrence of a relation between subsequent objects, such as chains of objects with identical relations. Semantics encoded in RDF triples allow derivation of such sequences by examining the attribute-object pairs of subjects. Third, sequences follow from numeric characteristics, such as weights indicating relevance of objects. Semantics encoded in RDF allow identification of numeric quantities, since the data type specification in XML schemas reveals whether items are of a numeric type. Fourth, inference rules allow derivation of relations between objects that order objects as sequences, such as for generation of rich narrative sequences. The first three sequence criteria are universal, since they are independent of the semantics themselves of the attribute instances.
Section 4.2 discusses sequence as a discourse construct for relating information objects in presentations.

4.2 Discourse

Many common ways of presenting a set of elements imply a notion of sequence [19]. It is important that sequence criteria make sense to users. For sequences of objects to be comprehensible and meaningful for humans, they should be arranged according to similar characteristics. Complexity, specificity, and causal relations between subsequent objects are general sequence criteria that need specification. The specification determines their semantics. Sequences of objects can result from mapping characteristics of these objects to other characteristics that are sequence criteria, for example through inference rules. As an example, events can be related to their period if periods are expressed as numbers, such as years, which allows a chronological sequence of events. The meaning of the resulting sequences depends on the sequence criterion, so that it is possible that sequences are not useful for readers.
Hierarchical presentation structures, such as Topias conceptual presentation structures, contain subsumption of multiple concepts under other concepts. A depth-first traversal of a hierarchical object structure is a sequence of specificity with specialisation and generalisation steps to lead readers through hierarchies in a comprehensible way. Concepts subsumed under a certain concept have equal position in the hierarchy. In the Topia conceptual presentations, the sequence of concepts subsumed under a concept is according to the relevance criterion explained in section 6. The sequences of artefacts within concepts are according to year of creation, a manually chosen numeric criterion [18]. However, the relevance criterion is as well applicable to individual artefacts in a concept.
Common objects of subsequent clusters can be the transition from the one cluster to the next. Maximisation of this type of transition for a given set of clusters can be a criterion for sequence of clusters. For this it should be possible to arrange the presentation ordering of the contacts of the two clusters such that the common object is the last item in the first cluster and the first item of the next. Such transitions save presentation space since the common objects are presented once for both clusters. It acts as a conceptual transition, a segue, between the clusters. Segues improve the aesthetics of the presentation and help convey the relation between the groups.

4.3 Presentation structure

In earlier work, we propose presentation structures in hypermedia for the sequence nucleus type in Rhetorical Structure Theory [15] conveying sequence. These presentation structures are bookshelf order (the order of stacked bookshelves), temporal order and next-buttons for navigating to the next node in a sequence [19]. In this earlier work, we also suggest presentation structures that contain hardly any notion of sequence in a set of objects compared to common presentation structures, namely random arrangements of objects, patchworks and grid structures. Scattering objects by these structures reduces an implied notion of sequence. Such presentations devoid of an implication of sequence avoid the risk of presenting information in sequences that have no meaning to users. Alternatively, criteria that presumably help users assess the content of presentations can be a basis for sequences of objects. This section discusses presentation structure devices for representation of sequence in three aspects of hypermedia presentations: space, time and link structure.

4.3.1 Space

Conveyance of sequences in space requires that the objects in the sequence be positioned in space with respect to each other, such that usual reading directions of humans imply the sequence. The relative distance between subsequent objects conveys the relative distance in the attributes or relations that are the basis for the sequence. Two-dimensional media can express two-dimensional sequences by putting objects in tables for conveying sequences according to two criteria. CSS has properties for supporting the positioning of objects required for the above-mentioned structures.

4.3.2 Time

Time-based presentations suggest a sequence running from the beginning of the time series to the end. This inherent sequence in time-based presentations strongly implies sequence. The sequence can however be adjusted by flashbacks and flash-forwards, resulting in presentation sequences that can disturb the expected ordering. Time-based presentations can convey development of real events in time. In addition, time-based presentations can convey ordering of space, such as in guiding tours [20]. Time-varied transitions convey the relative distance between subsequent objects, as well as the beginnings and ends of sequences that are put in concatenation. SMIL-enabled web-based presentations can contain the above-mentioned features in progressions in time.

4.3.3 Links

Sequences in navigation structures guide users through one or more paths of nodes in sequences specified by the navigation structure. Links in such navigation paths can represent relations between subsequent objects, spatial relations or separations of nodes applying to different events in time [20]. The hyperlink construct in the HTML standard supports these techniques.

4.4 Style

Sequences are typically presented in lists of ordered items. The CSS property list-style-type conveys sequence, or lack thereof, to the user. Most of its values prescribe numeric systems, typically numbers that precede the display of the element's children. The numeric system values correspond with the <ol> element in HTML, specifying an ordered list. These numeric systems emphasise that the displayed items fall in a sequence. The remaining values, such as disc and circle, correspond instead with the <ul> element, specifying an unordered list. They potentially communicate that the list is not necessarily a sequence.

5. EMPHASIS

Emphasis on objects in presentations indicates to readers that such objects have one or more properties, such as relevance, that distinguish the objects from other objects. Consequently, viewers of presentations expect emphasised objects to be worthy of note in some sense.
This section discusses derivation, from semantics, domain-independent distinguishing features, which are expressible in presentations by using emphasis. The discussion concerns the four phases in the presentation generation process, namely semantics, discourse, presentation structure and style.

5.1 Semantics

Semantics imply domain-independent distinguishing features of concepts in information repositories in multiple ways. First, distinguishing features follow directly from annotations that explicitly express that concepts are distinct with respect to other concepts, or distinct for specific users. Second, distinguishing features follow implicitly for concepts with attributes or a combination of attributes that not many other concepts have. Similarly, distinguishing features follow for concepts with attributes or a combination of attributes that are relevant for specific users. These latter cases of implicit distinguishing features are universal, since they are independent of the semantics themselves of the attribute instances. Appropriate presentation of concepts with such distinguishing features is dependent on the degree and nature of the features that distinguishes such concepts.
Semantics encoded in RDF triples [13] allow derivation of distinguishing features of RDF subjects by examining the attribute-object pairs of subjects for particularity with respect to other RDF subjects.
Section 5.2 discusses emphasis as a discourse construct for relating information objects in presentations.

5.2 Discourse

Distinct discourse characteristics of information objects suggest distinguishing features of such objects, and emphasises such objects with respect to other information objects. Examples of distinct discourse characteristics are central or extreme representations of information objects or groups of information objects, additional discourse characteristics and annotations. Variations of intensity, position, distance or direction of information objects in presentations convey such distinct discourse characteristics, emphasising the objects

concerned.
Regularity in discourse characteristics suggests a thread running through a presentation tying it together. Such regularity can be repetition of objects or specific types of objects, objects with consistently applied specific discourse characteristics, and rhythm, being a fixed structure of repetition. Such threads are conceived as prominent themes, express emphasis on the objects involved and thus allow focalisation of presentations. Broken regularity, such as absence of objects or discourse characteristics at some positions in otherwise regular structures, suggests exceptions.
Concepts in concept lattices are themes characterised by the attributes that the objects in a concept have in common. Regular discourse structures convey such themes. The subsumption structure of concept lattices is also a regular structure, since downward traversal invariantly results in specialisation and upward traversal in generalisation.
Concepts with many objects or attributes compared to other concepts are distinct concepts, as well as concepts with objects and attributes that are relevant for users. A relatively dense interconnection structure of concepts in concept lattices is also a distinguishing feature of the concepts involved. Putting such distinct objects at central or extreme positions in discourse structures emphasises such objects for users.

5.3 Presentation structure

Presentation structures specify the relations among objects in presentations while abstracting from the physical aspects of presentations. Presenting distinct objects in a way that is different in some sense from the presentation of other objects emphasises such objects. In hypermedia presentations, putting distinct objects at prominent positions, such as at a central or extreme position emphasises such objects, as well as association of additional objects such as text, images or symbols with such objects. Regular structures in one of the dimensions of hypermedia convey themes. This section discusses presentation structure devices for representation of emphasis in three aspects of hypermedia presentations: space, time and link structure.

5.3.1 Space

From a layout point of view, putting distinct objects at a central position, such as in the middle of the screen, or at extreme positions, such as on top of the screen, emphasises such objects. Alignment of objects along a spatial dimension conveys a stratification of emphasis on objects. Examples are distribution of objects in a lattice structure, indentation for indicating levels in hierarchical structures and the organisation of books, where titles of chapters are on top of pages and footnotes at the extreme bottom. In addition, alignment of objects conveys a regular structure of themes. In the hierarchical conceptual presentations generated by the Topia architecture, spatial grouping of branches conveys the fact that they are a theme, corresponding with a concept. Alignment of concepts and artefacts in the orientation bar conveys a stratification of emphasis that is related to the number of objects in concepts. CSS elements support positioning and alignment of objects in HTML presentations.

5.3.2 Time

Putting distinct objects at the beginning or end of a time sequence emphasises such objects. Increased presentation duration of objects in time-based presentations also emphasises objects. Variable pacing allows conveying a stratification of emphasis, typically by slowing down the pace proportionally to emphasis. Flashbacks and flash-forwards emphasise objects or events and allow repetition and regular structures. Temporal grouping, rhythm and fixed-length pauses also convey regularity. Players or browsers that support SMIL enable web-based presentations with the above-mentioned features in progressions in time.

5.3.3 Links

Objects linked to from many places in navigation structures emphasises such objects. Such objects can be central nodes that act as home or start pages of, for example, web sites such as portals. Furthermore, objects that have links to other objects have emphasis with respect to objects without links to other objects. In addition, names of links can express emphasis since they can contain an identification or annotation of the link. These techniques are all supported by the hyperlink construct in the HTML standard.

5.4 Style

The classical type of technique for emphasising objects is highlighting them in order to give emphasised objects distinct presentation characteristics with respect to other objects. Techniques for highlighting are setting objects size, use of different fonts and colours, flashing objects, use of icons such as arrows and frames around objects. A feature such as frame size conveys the intensity of the emphasis, and colour possibly the type of emphasis.
Style features such as colour are applicable to individual objects and do not inherently constrain the presentation of other objects. Style features allow addressing individual objects for emphasising. However, possible unintended effects of a combination of style features in presentations should be avoided, while in addition style can affect presentation of information and its presentation structure [16]. As examples, background colours should not mask colours in the media items or conflict with them, and application of many different fonts may inhibit readability.

6. USER CONTROL

In presentations of hierarchical structures, a meaningful sequence of a set of concepts that are subsumed under a concept provides readers with a means of relating the subsumed concepts to each other according to the applied sequence criterion. If the sequence criterion is relevance of objects, readers reading the sequence from beginning to end encounter each of the concepts in the sequence before all other concepts that are less relevant. A difficulty is that it is hard to tell beforehand what makes concepts relevant for users. We base the sequence of concepts subsumed under a concept on relevance for users, and consider a number of criteria that are optional relevance criteria for individual users. These criteria are the portion of the retrieval result covered by the objects, the amount of information available about the objects, and the relevance of the available information for individual users. We now explain how these relevance criteria relate to characteristics of concepts.
The first relevance criterion mentioned, being the portion of the total number of retrieved objects in concepts, is proportional to the number of objects in concepts. Consequently, we consider the number of objects in concepts as a measure of the concepts relevance.
The second relevance criterion, being the amount of information available about the objects, is proportional to the number of attributes of concepts. Consequently, we consider the number of attributes of concepts as another measure for the concepts relevance.
The third relevance criterion, being the relevance of the available information for individual users, requires that a specification of the relevance of attributes for users be available. Since users goals vary, different users consider different attributes as relevant. A way of letting users specify relevance of attributes is by requesting an assignment of positive numbers to attributes as relevance weights, such that higher numbers correspond to higher relevance of attributes. Since higher numbers of the previously mentioned relevance criteria also correspond to higher levels of relevance, a measure of the total level of relevance of a concept that follows from the three individual relevance criteria can be calculated according to the following formula.

$$R_{concept} = N_{objects} \times \sum_{i=1}^{N_{attribute}} W_i$$

In this formula, Rconcept is the relevance of the concept, Nobjects is the number of objects in the concept, Nattributes is the number of attributes in the concept and Wi is the weight of attribute i in a set of Nattributes attributes.
Multiplying the number of objects with the sum of the weights assigned to the attribute types results in their having equal effect on the resulting concept relevance, without their having to be of equal order of magnitude. Adding the number of objects to the sum of weights results in their having equal effect on the outcome only if they are in the same order of magnitude. Since the order of the number of objects in concepts generally increases with the total number of objects in the database, this would entail a need to bring the weights into accordance with the number of objects in concepts.
A set of weight values containing the values zero and one only allows users to designate attributes as either relevant or irrelevant without further distinguishing between the relevance levels.
The formula shows the calculation of the concept relevance when all three relevance criteria mentioned are involved. Leaving some of the relevance criteria aside requires adjustment of the formula. Excluding the first relevance criterion, being the number of objects in concepts, implies that the factor Nobjects must be removed from the formula. Excluding the second relevance criterion, being the number of attributes in concepts, implies that an additional division by the number of attributes in the concept must follow

calculation of the resulting Rconcept. Excluding the influence of differently valued weights implies that the number of attributes Nattributes replaces the summation factor.
A sequence of presentation of siblings in decreasing order of concept relevance in hierarchical conceptual presentations results in readers encountering siblings in decreasing order of relevance. Emphasising concepts with relevance levels that exceed a certain threshold level, such as zero, allows users to identify the objects in presentations with the specified relevance level at a glance.
The Topia architecture puts siblings in hierarchical conceptual presentations in a sequence according to relevance as explained in this section. With their query, users specify the level of relevance of the types of attributes that occur with the retrieved information objects. Figure 5 shows the specification form. The form shows the weights as well as a direction for users for applying the weights. Users specify one of six levels of relevance for each of these attribute types, or tick the extreme left column for specifying attribute types that should not be included in the presentation.
Attributes in the Topia repository have a type and a value. Topia allows user specification of the relevance of only the attribute types that occur in the retrieval result. Conceptually, users could as well be allowed to specify weights of attribute values. However, attribute types typically have many attribute values, resulting in a large amount of attribute values that occur in retrieval results. Letting users specify relevance for all of these requires considerable efforts. RDF encoded databases allow automated extraction of the attribute types and values of retrieved objects.

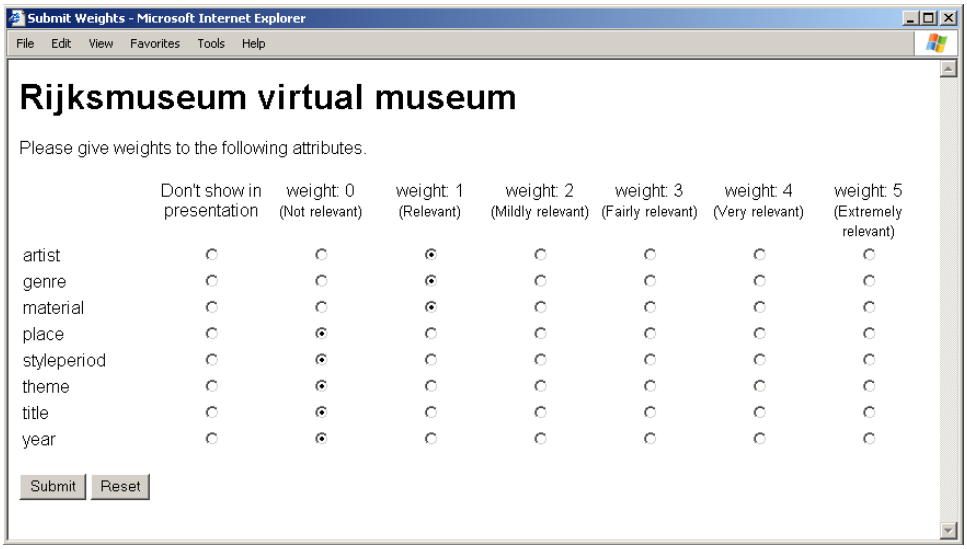


Figure 5. User specification of relevance of attribute types

To process the specified levels of relevance, they are assigned the integers from zero to five. Higher numbers in this range correspond to higher relevance levels, as shown in the table. The values of weights in the form are illustrative and not critical for a good performance of the sequence principle. In fact, users could be allowed to specify the weight values freely, allowing users to apply a weight distribution different from a set of successive integers. For calculating relevance of concepts, the Topia architecture applies the mentioned formula in order to involve all three stated relevance criteria.
A hierarchical list of concepts conveys the retrieval results, as described in section 3. Since people read from top to bottom, presenting the sequenced concepts from top to bottom requires a presentation device, so that at each hierarchical level, users encounter concepts in decreasing order of relevance. Emphasised concepts, with a relevance exceeding the threshold level, appear as blue links, while the non-emphasised are ghosted out.
Section 7 shows that sequences of siblings according to relevance of clusters for users as explained in this section allows focalisation of presentations to specific points of view.

7. DIRECTING DISCOURSE

Section 6 showed that sequence and emphasis in presentations position a set of information objects at a point in the story space. For users to obtain discourse that shows specific perspectives of sets of information objects requires an appropriate statement of relevance of attributes. This section shows how a statement of relevance of attributes results in discourse that give a corresponding perspective of a set of retrieved information objects. The discussion focuses on one of the relevance criteria only, being the attribute weights, since it is the only relevance criterion that relates to the contents of information objects.
Increasing the weights of specific attributes moves concepts with these attributes to the front of sequences

they are part of, allowing users to encounter such concepts first. Consequently, in order to put discourse in perspective, attributes that are characteristic of the required perspective must have higher weights than others in order to give the corresponding clusters high relevance. To illustrate this with the Topia architecture, we consider a user who wants artefacts about the theme water and specifies a query “water” in the artefact title. Among useful perspectives for readers of the retrieval results are the perspective of the art domain on the one hand and the perspective of time and place on the other hand. Considering the attributes that occur in the retrieval result at the extreme left in Figure 5, the following weight configurations are in accordance with the two perspectives.

1. Perspective of art domain: attributes artist, genre and material have weight value 1, other attributes have weight value 0.
2. Perspective of time and place: attributes place and year of creation have weight value 1, other attributes have weight value 0.

Figure 6 shows a presentation in the art domain perspective resulting from the weight configuration stated in item 1. Concepts that have attributes of type artist, genre or material appear above other concepts in the presentation sequence.

Figure 7 shows a presentation in the perspective of time and place resulting from the weight configuration stated in item 2. Concepts that have attributes of type place or year appear above other concepts in the presentation sequence.

In addition to users themselves, discourse domain experts can be involved in specifying the weight configuration of attributes for discourse with specific perspectives. Dynamic RDF encoded databases do not allow retrieval of an up-to-date set of attributes of information objects before the time of retrieval. Consequently, it is not known beforehand what attributes are available, which of the attributes relate to the required perspective and how they should be weighted to ensure a proper position of objects and attributes in the resulting discourse of the required type. A classification of attributes in the repository gives discourse domain experts a means for specifying the relevance of classes of attributes in presentations with specific perspectives.

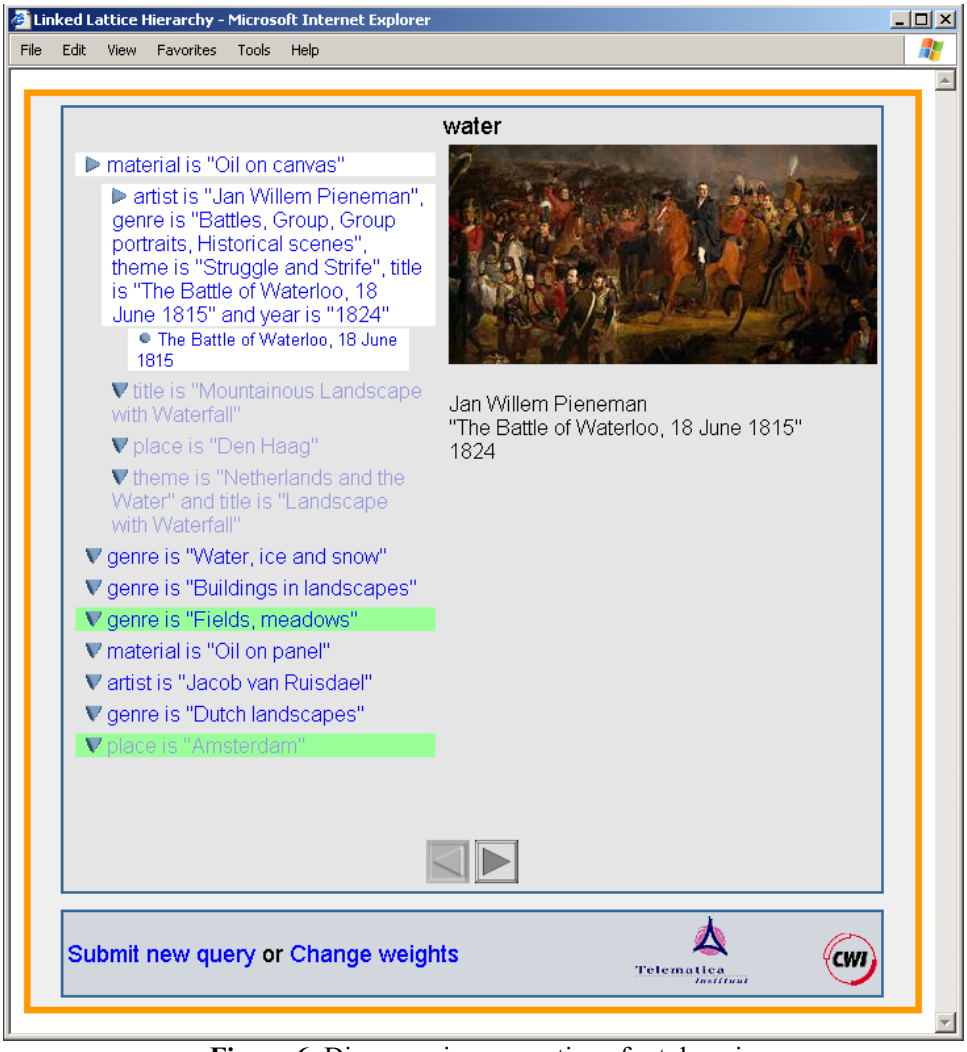


Figure 6. Discourse in perspective of art domain

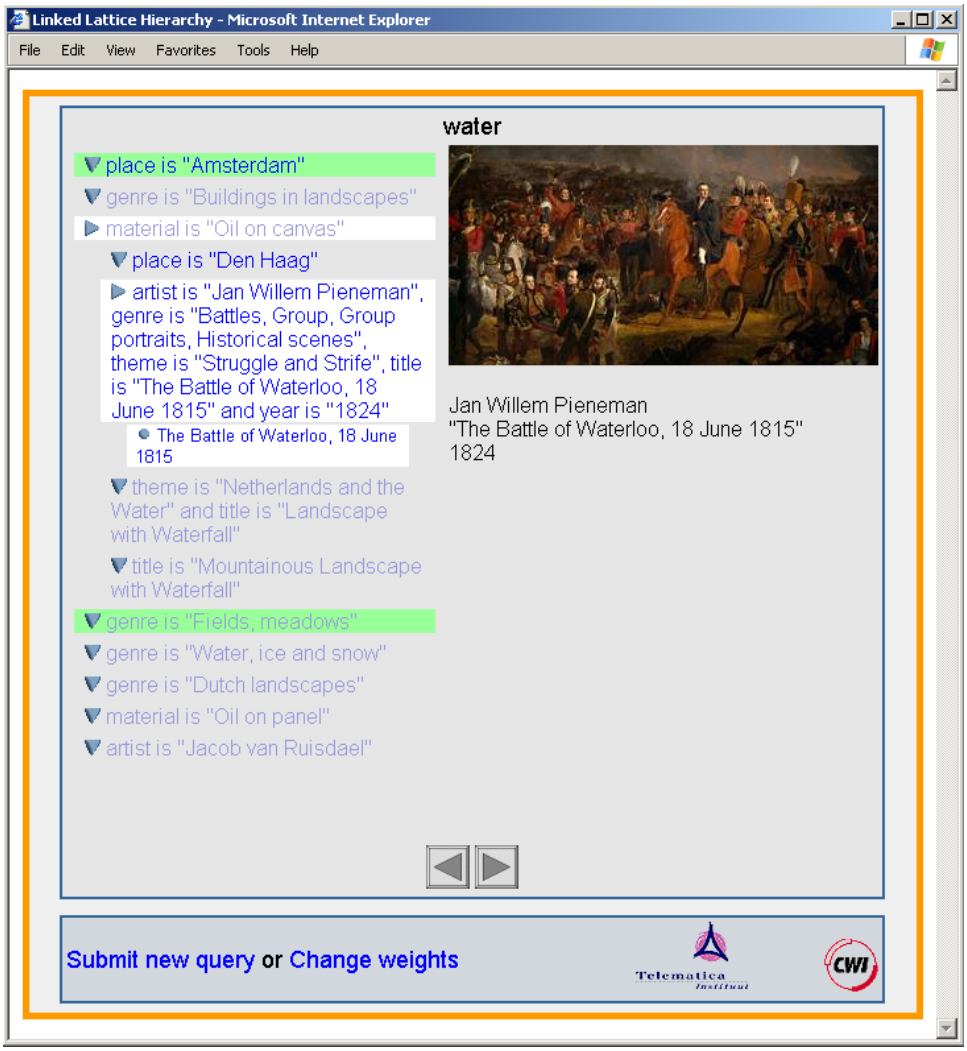


Figure 7. Discourse in perspective of time and place

8. FUTURE WORK

The work presented in this paper bases automatically generated sequence and emphasis on the relative number of objects and attributes of concepts and on relevance of attribute types for individual users. Another domain-independent criterion for sequences and emphasis is the subsumption structure in concept lattices. The subsumption structure occurring in concept lattices depends on the occurrence and distribution of attributes among the retrieved information objects. Sequences of concepts can be based on their number of child concepts or parent concepts, while emphasis on concepts can be based on a high number of parent concepts or child concepts. Analysis of concept lattices reveals the presence of distinct structures such as central concepts or intensively interconnected clusters of concepts, which can be emphasised. Presenting such relevant and prominent characteristics of concept lattices by means of discourse constructs to convey patterns in the retrieval result will be a topic of future research.

Topias current implementation generates concept lattices based on exact match of attributes of information objects. Extension of the exact match criterion with measures based on proximity of attributes can potentially increase the number and quality of clusters. Clustering techniques exploiting proximity of attributes have found their application in data mining for partitioning sets of objects [5]. The type of clustering technique determines the properties of the resulting clusters and hence the type of coherence among objects in clusters. In order to let users experience the objects in the resulting clusters as semantically close, the required distance measure between attributes for clustering should be accordingly. In spite of the required tuning, density-based numeric clustering techniques take the distribution of numbers in the retrieved data set into account for generating clusters of objects with relatively small numeric distance between the objects. Such techniques can be particularly useful for clustering numeric properties, such as the year of creation of artefacts.

Vector space models of information objects in an attribute space have found common application to express

similarities between information objects for information retrieval purposes [21]. Vector space models are a conceptual basis for clustering objects based on non-numerical attributes and for calculating clusters similarity to user queries. Discourse constructs such as sequence and emphasis can express such cluster characteristics in presentations. Future work will extend the applied clustering techniques and focus on their presentation in discourse constructs.
Another application of sequences is for conveying themes as threads through concept lattices. Such themes can concern subsequent clusters of attributes that have a specific identical attribute, but that do not occur under the same concept in the concept lattice. The user statement of relevance of attributes can be extended to a user statement of themes to be presented as paths along subsequent clusters in presentations. We will focus on automatic generation of such themes by means of sequence and emphasis and possibly other discourse constructs.
RDF databases are flexible because of their support for integration and inference rules without having to redefine the database structure. Consequently, attributes that occur in retrieval results cannot be determined earlier than at time of retrieval. It will be interesting to think about development of semantic structures that let domain discourse experts specify generation of perspectives of presentations by means of discourse constructs, in the absence of an exact knowledge of the attributes that occur in retrieval results.

9. SUMMARY AND CONCLUSION

This paper focuses on the automated derivation of two discourse constructs, being sequence and emphasis, from semantic annotations. The results of this work are a continuation of the Topia project, which generates discourse structures from clustering of semantic annotations. Other approaches focus on human-authored narrative templates for specifying sequence and emphasis. We present requirements for automated domain-independent generation of sequence and emphasis in the four phases of our processing chain, being analysis of semantic annotations, clustering, discourse structure generation and hypermedia generation. We also present an overview of the support that web standards, including the Semantic Web standard, offer for this. Principles for discourse generation that are independent of specific domain semantics allow automatic generation of narrative presentations from the contents of multiple repositories in web environments, irrespective of their application field.
Domain-independent criteria for sequence and emphasis follow from two sources of information. First, such criteria can be derived from attributes of information objects. Hard-coded sequences, numerical attributes and chains of information objects with identical relations between subsequent objects are sequence criteria that can be derived automatically. The occurrence of relatively large clusters of information objects that have identical attributes is a criterion for emphasis, as well as occasional attributes of objects with respect to those of other objects. A second criterion for sequence and emphasis is relevance of information objects for individual users. We present a relevance criterion that takes both types of criteria into account. The latter, subjective, criterion is according to a user-specified expression of relevance of information objects, stated by assigning relevance weights to attribute types that occur in the metadata repository.
This paper demonstrates application of the presented relevance criterion in the Topia architecture, in order to generate sequenced and emphasised clusters of objects in presentations of artefacts from the Rijksmuseum Amsterdam collection. RDF encoded annotations allow derivation of the actual set of attributes that occur with the retrieved objects at time of retrieval. Finally, we show that the user statement of relevance is a basis for generating presentations that put the retrieval result in specific perspectives.

ACKNOWLEDGMENTS

Funding for work on this paper came from the Topia project of the Telematica Instituut and CWI. Lynda Hardman and Frank Nack of CWI provided many helpful comments for improvement. Stanislav Pokraev of Telematica Instituut helped clarify the discussion of XML and RDF technology for this work. We thank the Rijksmuseum Amsterdam for their permission to use their Websites database and media content. We also thank IBM for sponsoring the project.

REFERENCES

1. Alani, H., Kim, S., Millard, D.E., Weal, M.J., Hall, W., Lewis, P.H., and Shadbolt, N.R. Automatic Ontology-based Knowledge Extraction from Web Documents, IEEE Intelligent Systems, 18(1) (January-February 2003), 14-21.
2. André, E., The Generation of Multimedia Documents, in. Dale, R, Moisl, H. and Somers, H. (eds.), A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text, Marcel Dekker Inc., 2000, 305-327.
3. Bal, M. Narratology: introduction to the theory of narrative, second edition. University of Toronto Press, 1997.
4. Bateman, J., Kamps, T., Kleinz, J. and Reichenberger., K. Towards constructive text, diagram and layout generation for information presentation, Computational Linguistics 27(3), 2001, 409-449.
5. Berkhin, P. Survey of clustering data mining techniques, http://www.accrue.com/products/rp_cluster_review.pdf
6. De Bra, P. Pros and Cons of Adaptive Hypermedia in Web-based Education. Journal on CyberPsychology and Behavior, Vol. 3, No. 1, Mary Ann Lievert Inc., 2000, 71-77.
7. Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M. and Horrocks, I. The Semantic Web: The roles of XML and RDF, IEEE Internet Computing, 15(3), 2000, 63-74.
8. Buckingham Shum, S., Uren, V., Li, G., Domingue, J. and Motta, E. Visualizing Internetworked Argumentation, In: Kirschner, P.A., Buckingham Shum, S.J. and Carr, C.S. (eds), Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making, Springer-Verlag: London, 2003, 185-204.
9. Ganter, B., and Wille, R., Applied Lattice Theory: Formal Concept Analysis. Preprints http://wwwbib.mathematik.tudarmstadt.de/Math-Net/Preprints/Listen/pp97.html, 1997.
10. Geurts, J., Bocconi, S., van Ossenbruggen, J., and Hardman, L. Towards Ontology-driven Discourse: From Semantic Graphs to Multimedia Presentations, technical report INS-R0305, http://ftp.cwi.nl/CWIreports/INS/INS-R0305.pdf, 2003.
11. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Yee, K. Finding the flow in web site search. Communications of the ACM, Vol. 45, No. 9, 2002, 42-49.
12. Kamps, T., Diagram Design : A Constructive Theory, Springer Verlag, 1999.
13. Lassila, O. and Swick, R.R. (eds), Resource Description Framework (RDF) Model and Syntax Specification. World Wide Web Consortium (W3C) Recommendation, February 22nd, 1999.
14. Little, S., Geurts, J. and Hunter, J., Dynamic Generation of Intelligent Multimedia Presentations through Semantic Inferencing. In: Proceedings of the Sixth European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2002), Springer, September 2002, 158-189.
15. Mann, W., Mattheissen, C., and Thompson, S. Rhetorical Structure Theory and Text Analysis. Information Sciences Institute Research Report, ISI/RR-89-242, 1989.
16. Van Ossenbruggen, J. and Hardman, L. Smart Style on the Semantic Web. In: Semantic Web Workshop, WWW2002, http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-55/ossenbruggen.pdf, 2002.
17. Rijksmuseum Amsterdam, Rijksmuseum Amsterdam Website. http://www.rijksmuseum.nl
18. Rutledge, L., Alberink, M., Brussee, R., Pokraev, S., Van Dieten, W. and Veenstra, M. Finding the Story Broader applicability of Semantics and Discourse for Hypermedia Generation. ACM Hypertext, 2003. (to appear)
19. Rutledge, L., Davis, J., Van Ossenbruggen, J. and Hardman, L. Inter-dimensional Hypermedia Communicative Devices for Rhetorical Structure. In: Proceedings of the International Conference on Multimedia Modeling 2000 (MMM00), Nagano, Japan, November 13-15, 2000, World Scientific, 89-105.
20. Rutledge, L., van Ossenbruggen, J., Hardman, L. and Bulterman, D. Structural Distinctions Between Hypermedia Storage and Presentations. In: Proceedings of ACM Multimedia (pages 145-150), ACM Press, 1998.
21. Wong, S., Raghavan, V. Vector Space Model of Information Retrieval: A Reevaluation. In: Rijsbergen, C.J. van (Hrsg.), Research and Development in Information Retrieval, Cambridge University Press, Cambridge, UK, 1984, 167-186.

User Interaction in Modern Web Information Systems

Peter Barna - Geert-Jan Houben
Technische Universiteit Eindhoven
PO Box 513, NL-5600 MB Eindhoven, The Netherlands
{p.barna, g.j.houben}@tue.nl

Abstract

Modern Information Systems based on Web technologies (Web-based Information Systems - WIS) typically generate hypermedia presentations according to the user needs. Hera is our model-driven methodology specifying the design cycle and the architecture framework for WIS. To avoid additional expensive programming the functionality of generated hypermedia presentations is limited to following links. However, modern e-commerce applications require more sophisticated user interaction. In this paper we discuss extensions of the Hera methodology regarding the design of such interactive WIS. We explain the main ideas of the extension on the example of a virtual shopping cart application, as a typical pattern appearing in e-commerce applications.

1. Introduction

Many information systems today use the Web as a platform. Their remote clients interact with the system through Web browsers. Increasing demands of E-commerce require richer functionality of such Web-based Information Systems (WIS). This rich functionality goes together with richer means of user-interaction compared to just following links. For the sake of conciseness in the rest of the paper we call such WIS (though not completely correct) *interactive WIS*.

Due to the specific nature of the Web, a number of methodologies have been developed particulary for WIS design ranging from earlier methodologies as RMM [8], through the object-oriented approaches as OOHDM [13] and UWE [11], to methodologies as WebML [2]. Some of the methodologies do not support the design of interactive WIS (e.g. RMM considers only static navigation specification), and some do (e.g. object-oriented methods or WebML).

In our perception, a WIS is an information system generating hypermedia presentations delivered by means of the Web to users. Hera [5] is a model-driven WIS design methodology that specifies a number of design steps and their outputs in terms of models. The models describe different facets of the system and are used during the process of hypermedia presentation generation. Every concrete model is built from primitive concepts that are defined and hierarchically organized in a *schema* for the model. An analogical example in an object-oriented structure modelling method (e.g. UML class diagram) is a concrete class structure where a schema for these models defines terms as "class", "association", "specialization", etc. and their relationships.

Hera supports the generation of adaptable and adaptive hypertext presentations. Adaptability is adjusting presentations to features known before the generation process (for example the characteristics of the hardware platform: a presentation looks differently on a WAP phone and on a PC). Adaptivity is conditional inclusion of page fragments and conditional link hiding where both are based on dynamically changing features: a user model is dynamically updated during the browsing. Although both mechanisms contribute to the usability of presentations for concrete users, they do not make presentations interactive (in the sense we have defined earlier).

Even though Hera was not explicitly aimed for the design of interactive WIS, we consider possibilities of its deployment for these applications. In this paper we investigate the application of Hera for authoring of interactive WIS. A possible application is demonstrated on a simple example of a poster sales process in an on-line museum poster shop, where we show the combination of specifying interaction and navigation structure.

In section 2 basic principles of Hera's methodology and framework are briefly described. The requirements for the on-line poster shop example are specified in section 3, and the proposed design focusing on the

navigation and interaction aspects is explained in section 4. Possible consequences for the extension of the Hera architecture and its implementation are discussed in section 5.

2. Hera

In Hera a WIS accepts a request from the user, collects necessary data from distributed data sources, integrates the data, and using a process of data transformations forms a hypermedia presentation in concrete end format (e.g. HTML).

The Hera methodology defines a set of design steps that need to be taken to build a set of models and data transformation procedures. The models specify views on certain aspects of the transformation process, particulary the structure of data in different stages of the process. Concrete models are constructed from primitive concepts that are defined in so called model schemas.

According to Hera model data processed in a WIS is in the RDF (Resource Description Framework) [12] format serialized in XML. The benefit of using RDF is its more explicit semantics compared to XML. For definitions of models and model schemas we use RDF Schema (RDFS) [1]. For query specifications within the system we use RDF Query Language RQL [9].

2.1 Methodology

Typical WIS design methodologies distinguish the following phases:

- **Requirement Analysis** gathers and forms the specification of the user requirements.
- **Conceptual Design** defines the conceptual model for the problem domain.
- **Navigation Design** builds the navigation view of the application.
- Some methodologies include **adaptation design**, where the adaptation model is built and all associated mechanisms are defined.
- **Presentation Design** defines the appearance and layout of generated presentations.
- **Implementation** realizes the WIS itself.

The Hera methodology redefines the following phases and models (see Figure 1):

- **Conceptual and integration design**. The main outputs of this phase are the Conceptual and Integration Models (CM, IM). Since we consider a distributed and heterogeneous (in format and content) data repository, CM gives a unified semantic view on the repository (to facilitate further design steps), whereas IM specifies semantic links from concepts in particular sources to concepts in CM.
- **Application and adaptation design**. In this phase the designer creates the Application Model (AM), and a set of models for adaptation (e.g. set of adaptivity rules for adaptivity, a specification of the user/platform profile, and initial user model). AM is built on top of CM and describes the overall structure of generated presentations including navigation. Adaptation of the generated presentation is based on static features (known before the generation process, e.g. user/platform properties), or the dynamic features (changing during browsing) based on a user model. For adaptivity we can use the AHA! engine [3] (based on AHAM reference model [4]) is used.
- **Presentation design**. In this phase the designer specifies the layout and rendering of presentation units in the presentations.

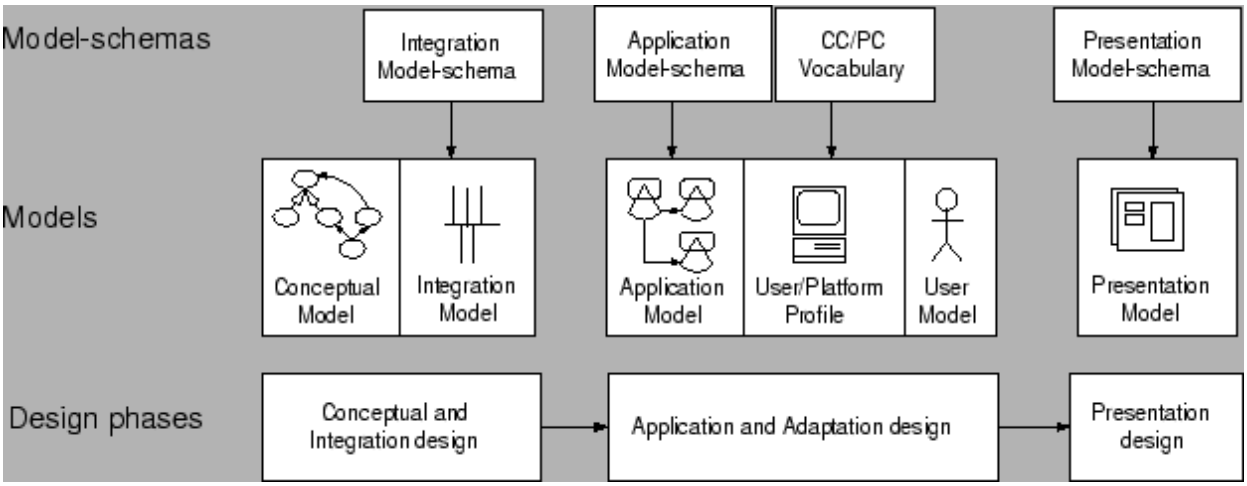


Figure 1: Design phases and models in Hera

2.2 Models

Models in Hera specify different facets (views) of the system and presentations generated by this system. In the following paragraphs we shortly explain models we use later in the example: conceptual and application models.

2.2.1 Conceptual Model

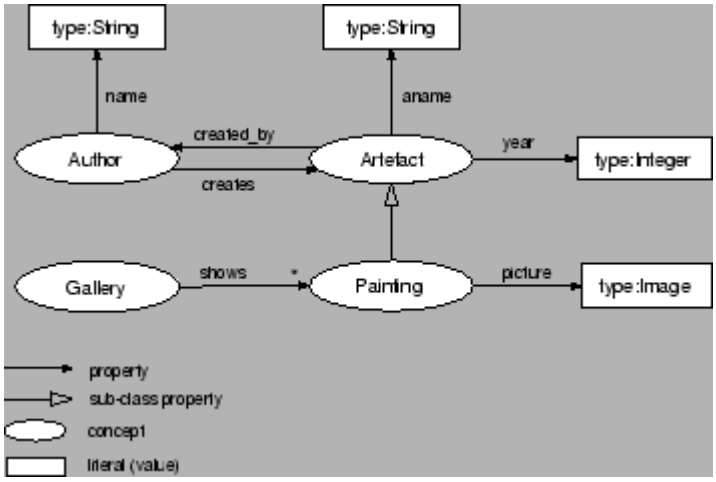


Figure 2: Example of a CM

The conceptual model describes the semantics of the data repository (problem domain) by means of concepts and their properties. The properties have as values other concepts, or concrete values (literals). There is a special property, the sub-class property, that represents concept inheritance. The schema for CM is actually a RDFS data model with added properties *cardinality* describing multiplicity of other properties, and *inverse* representing reversal of properties. An example of CM represented in RDFS graphical notation is in Figure 2.

The ovals represent concepts, and the rectangles represent literals (values). Concepts can have arbitrary properties with ranges of the types concept or literal. For the `sub-class` property the range concept inherits all properties of its domain concept, for instance, the `Painting` concept has also the `aname` property.

An example of inverse properties are the `created_by` and `creates` properties. An example of a property with multiple cardinality is the `shows` property (note the star).

2.2.2 Application Model

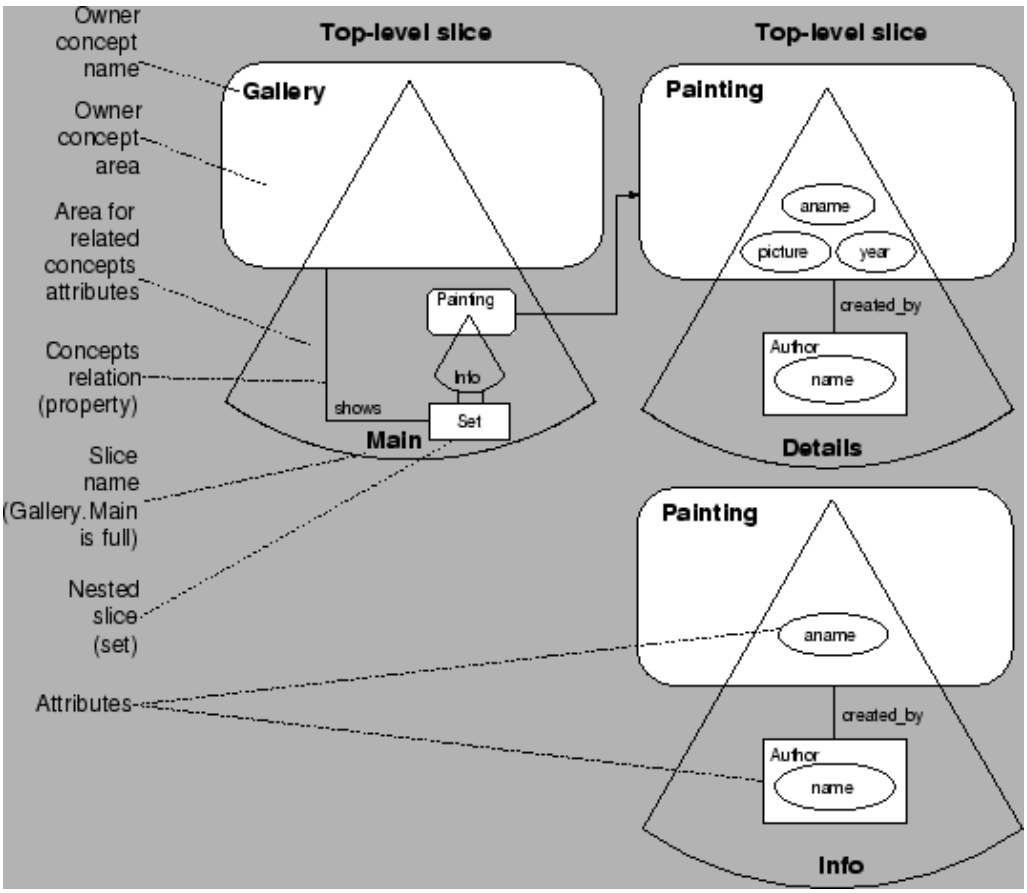


Figure 3: Example of an AM

The application model describes groupings of concept attributes (from CM) to semantically meaningful units and relationships between the units. Such units are called slices. A slice contains selected attributes of a so called owner concept, but can also contain attributes of related concepts. Slices can be aggregated (one slice can contain another slices), or referenced. Slices roughly represent page fragments in generated presentations (of course without spatial, temporal, or rendering details), and top-level slices (not contained in another slices) represent pages. References represent links.

An example of an AM based on the CM from the previous paragraph is in Figure 3. The top-level slice `Gallery.Main` has the nested slice `Painting.Info` that will be rendered as a set of its instances (the `shows` property in CM has multiple cardinality). From a concrete slice instance of `Painting.Info` there is a reference (link) to concrete instance of `Painting.Details` showing complete information about the given painting.

2.3 Data Transformations

When the user queries the data repository and wants to obtain the desired information in form of a hypermedia presentation, his query is re-distributed over data resources, and the data transformation process is performed in the steps (Figure 4):

- **Integration and data retrieval**, where the required data is collected from different data sources and transformed into a CM instance. The IM is used for the query re-distribution and data integration.
- **Presentation generation**, where the CM instance is transformed into an AM instance, and then into a final presentation in concrete format (e.g. HTML, WML, or SMIL) using PM. During generation of an AM instance, the adaptability conditions are evaluated and appropriate AM elements are included into the presentation. Moreover, the user model is instantiated.

All data transformation procedures in Hera are specified in XSLT [10] sheets. The procedures do not depend on concrete models. More details of Hera can be found in [5,7].

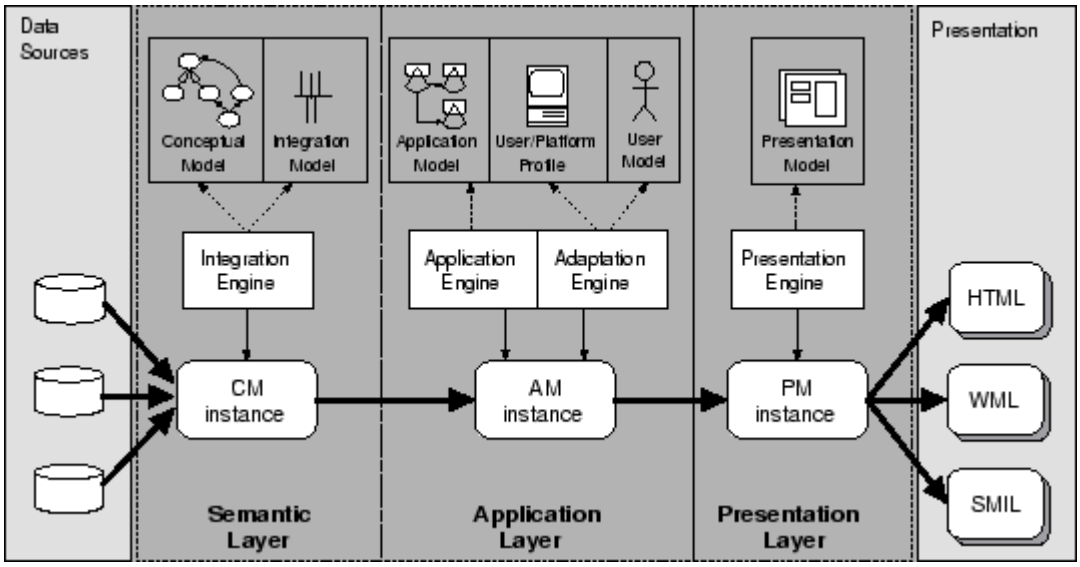


Figure 4: Data transformations in Hera perspective

3. Poster Shop Example

The example is an extension of a museum site with on-line shop selling posters related to paintings from the museum. Figure 5 shows details of how we envision the structure of application pages for the sales process, and specifically the shopping cart. The application should allow searching for desired posters based on their subjects. The user can put selected items into the shopping cart, can report and update the content of the cart, and finally can confirm the purchase.

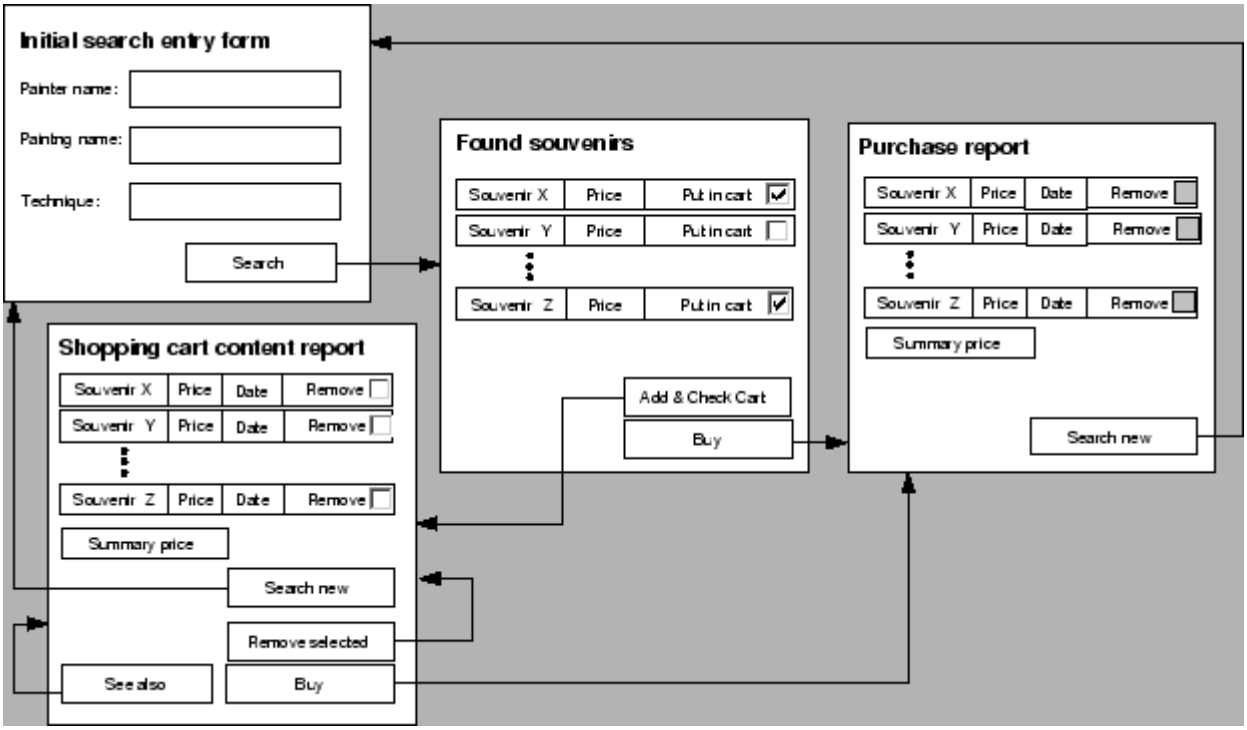


Figure 5: Envisioned application structure

A typical user scenario is:

- The user sees the `Initial search entry form`, where he can enter names for a painting, painter, or technique to find posters related to paintings matching the criteria.
- By pressing the `Search` button the `Found posters` page is rendered with a list of posters matching entered criteria. The user marks the items he wants to put into his virtual shopping cart. Marked items can be bought by pressing the `Buy` button, or the content of user's shopping cart can be

viewed and updated by pressing the `Add & Check Cart` button.
- If the content of the cart was bought, the `Purchase report` page is displayed with the option of a new search.
- If the `Shopping cart content report` was displayed, the user can remove items from it (by marking them and pressing `Remove selected`), go to a new search to add other items (by pressing `Search new`), or to perform the purchase (by pressing `Buy`).

The described system is very simple and far from complete (e.g. no payment processing is considered), but it serves our purposes of demonstrating how this interaction can be specified and implemented well.

4. Interaction Design in Hera

In this section we design the example application using Hera and we point out how we can extend the Hera models and architecture to facilitate interaction design. Since the desired application structure is captured in AM, we focus on the AM specification.

4.1 Conceptual Model of the Poster Shop

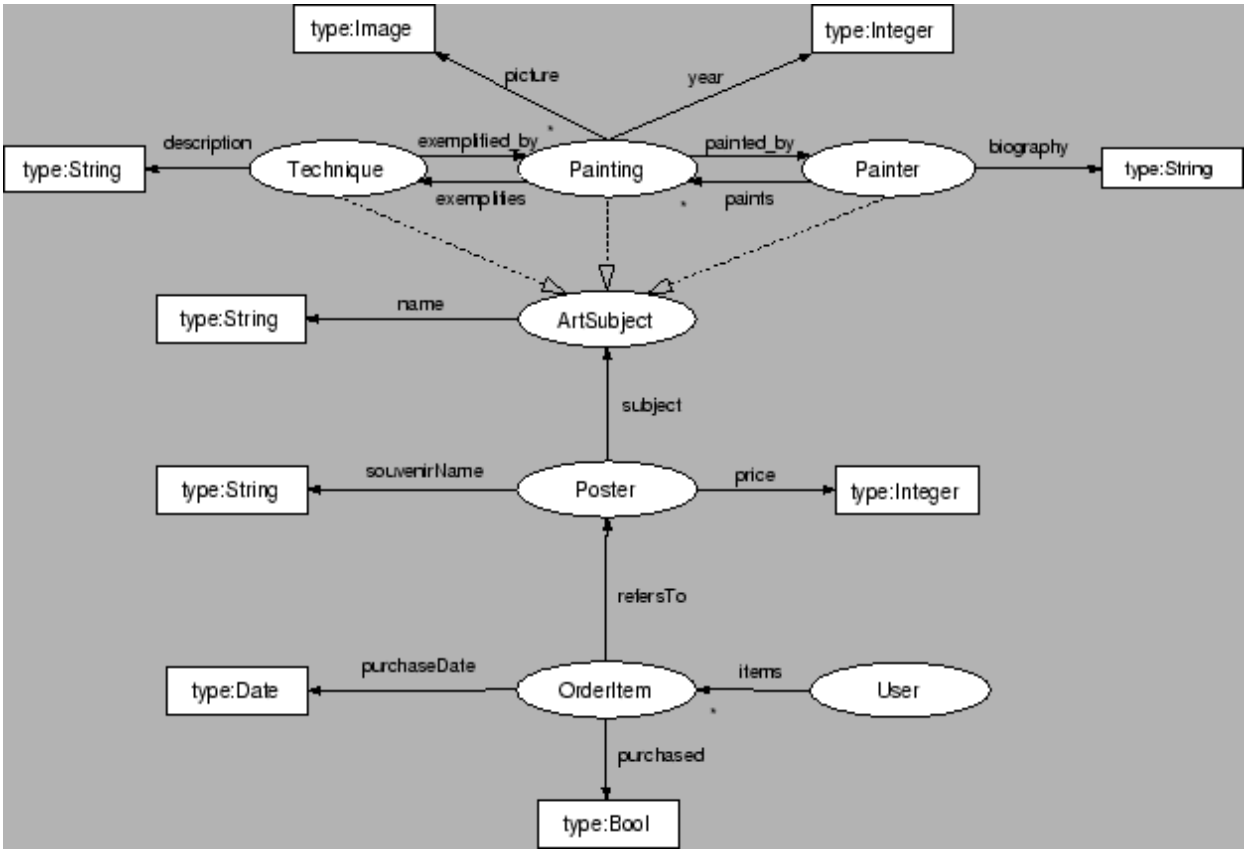


Figure 6: The CM of the poster shop

The CM of our example (see figure 6) contains the main concepts `Poster`, `ArtSubject` (covers paintings, techniques, and painters), and `OrderItem` (represents an item in the shopping cart). The `purchased` attribute of `OrderItem` determines wether the item was purchased or it is only in the shopping cart.

4.2 Application Model of the Poster Shop

The AM in Figure 7 outlines the envisioned application structure by means of the slices:

- `Poster.List` contains a list of (found) posters, where the user can choose the posters he wants to put into the shopping cart; pressing the `checkCart` button opens the cart report (`OrderItem.CartReport`); pressing the `buy` button performs the purchase and displays the purchase report represented by the slice `OrderItem.PurchaseReport`. The `Poster.List` slice corresponds to the `Found posters` page from Figure 5.

- `OrderItem.CartReport` contains a list of items in the shopping cart; the buttons there allow removing of items from the cart, displaying related items to these in the cart (posters with the same subjects), and performing the purchase. The `OrderItem.CartReport` slice corresponds to the `Shopping cart content report` page from Figure 5.
- `OrderItem.PurchaseReport` contains a list of purchased posters; the button `Search` opens the initial search page. The `OrderItem.PurchaseReport` slice corresponds to the `Purchase report` page from Figure 5.
- `Poster.ListItem` (nested in `Poster.List`) is an item in the list of found posters.
- `OrderItem.ListItem` is an item in the list of items (in a cart or purchased).

For the interaction we have extended the original definition of AM specification in two dimensions:

- The slices in AM can contain new elements that were not used in AM specifications before (e.g. button). They capture events caused by the user, or serve for data entering and output. The specification of these elements represents a *structural* extension of AM specification.
- Naturally, data and events provided by the user must be processed on-line. In addition, the data content of slices may depend on previously collected data. All this processing must be specified in some way. Therefore, it is clear that we need also a *functional* extension of slices that goes beyond simple navigation specification. This could include the manipulation of system/session state data, for instance the content of a shopping cart.

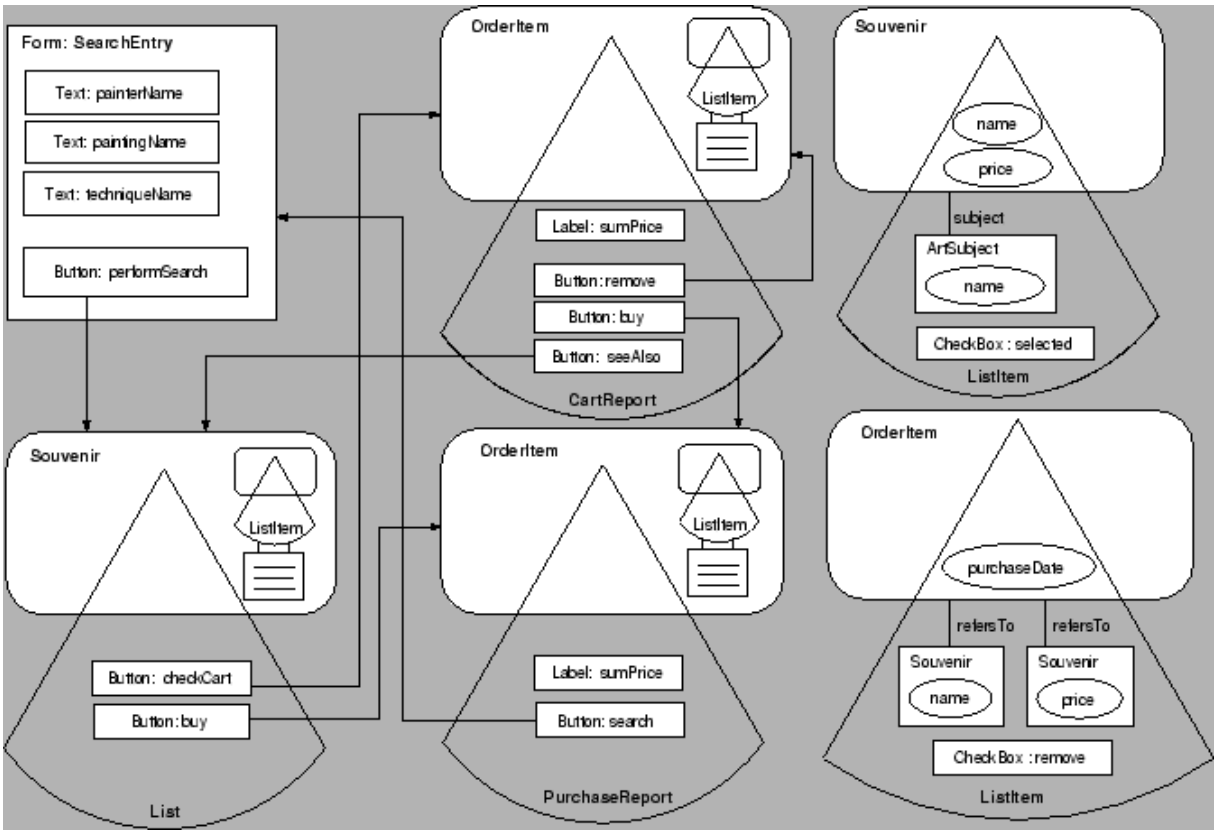


Figure 7: The AM of the poster shop

4.2.1 Structural Extension

As illustrated in Figure 7 we add to the AM definition new elements, called controls:

- **Button**. An example is the `Search` button in the `SearchEntry` form. It is obvious, that the button should be associated with operations to perform (as we explain further) when the button is pressed. The `performSearch` button should together with the `Poster.List` slice provide the list of matching souvenirs.
- **Checkbox**. An example is the `selected` checkbox in `Poster.ListItem`, where the user determines which items are put into the shopping cart. The checkbox control has assigned to it a boolean value that can be changed by the user.

- **Text entry field**. This field is needed for example for entering search criteria in the `SearchEntry` form; the value of a text entry field control is a string and can be changed by the user and read by the application.
- **Label**. Unlike the regular data slice attribute, a label is calculated from values of other attributes or controls. An example of the label is `SumPrice` in the `OrderItem.CartReport` slice. Labels are associated with assignment operations (e.g. an RQL query returning a single value).

4.2.2 Functionality Extension

Before we show how to specify the behavior of controls we need to realize the following:

- Due to user interaction that may influence the data content of slices we need to select the data content dynamically just before a slice is reached via navigation. The link, or link anchoring control should provide the condition selecting the data instances for the target slice. We call this **dynamic slice instantiation**.
- Thus generated hypermedia presentations are not static anymore, the system should maintain its **state information**: in our example the content of the virtual shopping cart. The system should be able to read and update this type of information.

Obviously, the structural diagram in Figure 7 does not describe the functionality related to controls, or the functionality related to data retrieval/update. For the sake of simplicity we show the functional specification only for the `Poster.List` slice. Let us assume that the `checkCart` button is pressed by the user.

- **The users's shopping cart** is updated according to the currently selected and unselected items in the souvenirs list:
  - **Selected posters are added to the shopping cart**. New instances of `OrderItem` are created. In RQL it can be:

    ```
    INSERT INTO OrderItem (refersTo, purchaseDate, purchased)
       FROM   X:ListItem
       VALUES {X}.root, null, 'false'
       WHERE  X.selected='true'
    ```

    The nested slice `ListItem` is referenced here in a similar way as a concept, and its attributes are referenced as concept attributes (`selected`). The `root` expression acting as a slice ``property'' returns the slice root concept (e.g. `Poster` for `Poster.ListItem`).
  - **Unselected souvenirs are removed from the shopping cart**. All `OrderItem` instances corresponding to the unselected posters with (the `purchased` property set to `false` means that the items are in the basket and are not purchased yet) are deleted. Using RQL we can express it:

    ```
    DELETE   X
       FROM  {X:OrderItem}refersTo{Y:Poster},
             {Z:ListItem}root{Y}
       WHERE Z.Selected='no', X.purchased='false'
    ```

- **The `OrderItem.CartReport` slice is instantiated and displayed**. The condition determining instances of the `OrderItem.ListItem` should be specified: all instances with the `purchased` value equal to `false`.

  ```
  REF( purchased='false' )
  ```

  Taking into account the structure of `OrderItem.CartReport` the system will execute this constructed RQL query:

  ```
  SELECT X.purchaseDate, Y.name, Y.price
  FROM   {X:OrderItem}refersTo{Y:Poster}
  WHERE  X.purchased='false'
  ```

  The structure of the `SELECT` clause is based on the attributes of the target slice `OrderItem.CartReport` with nested `OrderItem.ListItem`.

Since the principle is similar for the buy button, we do not show the specification for it. The specification can be serialized into an RDFS file. The sample pattern of such a file is:

```
<rdfs:Class rdf:id="Slice.Poster.List">
<rdfs:SubClassOf rdf:ID="#Slice">
...
<rdfs:Class rdf:ID="checkCart">
  <rdfs:SubClassOf rdf:resource="#Button"/>
  <rdfs:Class rdf:id="checkCart-processing">
    <rdfs:SubClassOf rdf:resource="#Processing"/>
    <rdfs:Class rdf:id="InsertItem">
      <rdfs:SubClassOf rdf:resource="#Operation"/>
        <AMS:OpBody>
          INSERT INTO OrderItem (refersTo,purchaseDate, purchased)
          FROM    X:Slice.OrderItem.ListItem
          VALUES  {X}.root, null, 'false'
          WHERE   X.selected='true'
        </AMS:OpBody>
      </rdfs:Class>
    </rdfs:Class>
  ...
  </rdfs:Class>
</rdfs:Class>
```

The X argument is bound to the OrderItem.ListItem slice, and the RQL command using it creates an instance of the OrderItem concept. The owner here is not the name of an slice attribute, but refers to the owner concept of the slice OrderItem. The AMS is a namespace of the AM schema, which defines among others also the Slice, Button, Processing, and Operation concepts.

5. Architecture of interactive WIS

The architecture of WIS should be refined to allow this kind of interaction. Mainly, there is a need of an execution engine that would perform dynamic slice instantiation based on query processing and data retrieval, and data updates. The engine should provide:

- Reference operations: the system processes the operation and returns a slice instance(s) in the form of the element (e.g. HTML page) to be displayed next, where it needs to be performed:
  - construction of the RQL query from the target slice structure (SELECT and FROM clauses of an RQL query from the slice structure and the construction of the WHERE clause from the condition) and from the selecting condition that is the part of the reference operation,
  - execution of the query, and
  - translation of the raw data into slice instance(s) and then into the presentation unit (in whatever format, e.g. HTML).
- Data manipulation operations, and other RQL queries: the system evaluates all references in queries and executes them.
- External calls (e.g. for on-line payment, but also same other more complex functions as complete shopping carts), typically static calls of web services (using SOAP and WSDL) and possibly calls of dynamically discovered (loosely coupling) web services using UDDI, and perhaps DAML-S, etc.

5.1 Implementation Issues

To make the system versatile, the engine performing dynamic generation of presentation pages (based on dynamic slice instantiation) and processing user events/data would include all engines from figure 4. The prototype system we developed runs as a servlet under a host web server (Apache Tomcat). This servlet processes the Get (HTTP client asks for another page that should be provided) and the Post (the client sends an event to the server and values of controls are read) HTTP messages.

As a response to the Get message the system:

- determines the next slice from AM that will be rendered as the next page,
- performs a data query constructed from the selecting condition associated with the reference operation and from the structure of the target slice,
- retrieves data (creates a CM instance) and creates an instance(s) of the target slice (an AM instance), and
- transforms the AM instance into the presentation page in an appropriate format (e.g. HTML)

As a response to the `Post` message the system will collect the data provided by the user interacting with concrete controls. The system reads data values of controls that are passed to the `Post` message as arguments. Instances of controls are bound with concrete slice instances during page generation via forms and nested hidden arguments (`<form/>` in XForms and HTML).

6. Conclusion

The ideas proposed here point to extension of our models, schemas, and architecture regarding the design of interactive WIS. We have demonstrated the need of interaction in typical e-commerce applications. With this paper we have established the direction of the research and for now omitted exhaustive schema specifications and complete sets of controls and corresponding mechanisms. We have indicated how these ideas can be implemented on the basis of our prototype.

Another aspect that is a subject of our intensive research and is not sufficiently covered yet, is automation of the design process of such interactive presentations. There are several possible ways that we investigate, for instance re-use CM and AM patterns, or semi-automated generation of AM from CM and formalized goals/tasks of the system.

References

1. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft (2002
2. Ceri, S., Fraternali, P., Matera, M.: Conceptual Modeling of Data-Intensive Web Applications. IEEE Internet Computing, vol. 6, number 4, pages 20-30 (2002)
3. De Bra, P., Aerts, A., and Houben, G.J., Wu, H.: Making General-Purpose Adaptive Hypermedia Work. In Proc. WebNet World Conference on the WWW and Internet, AACE (2000)
4. De Bra, P., Houben, G.J., Wu, H.: AHAM: A Dexter-based Reference Model for Adaprtive Hypermedia. In Proc. The Tenth ACM Conference on Hypertext and Hypermedia, ACM Press (1999)
5. Frasincar, F., Houben, G.J., Vdovjak, R.: Specification Framework for Engineering Adaptive Web Applications. In Proc. The Eleventh International World Wide Web Conference, Web Engineering Track, ACM Press (2002)
6. Gervais, M.P.: Towards an MDA-Oriented Methodology. In Proc. 26th Annual International Computer Software and Applications Conference, IEEE Computer Society (2002)
7. Hera web page and software prototype. Available on the URL http://wwwis.win.tue.nl/~hera
8. Isakowitz, T., Stohr, E., Balasubramanian, P.: RMM: A Methodology for Structured Hypermedia Design. Communications of the ACM, volume 38, number 8, pages 34-44 (1995)
9. Karvounarakis, G., Alexaki, S. Christophides, V., Plexousakis, D., Scholl M.: RQL: A Declarative Query Language for RDF. In Proc. The Eleventh International World Wide Web Conference, ACM Press (2002)
10. Kay, M.: XSL Transformations (XSLT) Version 2.0. W3C Working Draft (2002)
11. Koch, N., Kraus, A., Hennicker, R.: The Authoring Process of the UML-based Web Engineering Approach. In Proc. First International Workshop on Web-Oriented Software Technology (2001)
12. Lassila, O., Swick, R.R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation (1999)
13. Schwabe, D., Rossi, G., Barbosa, S.D.J.: Systematic Hypermedia Application Design with OOHDM. In Proc. The Seventh ACM Conference on Hypertext (1996)

CHIME: Service-oriented Framework for Adaptive Web-based Systems

Vadim Chepegin[1], Lora Aroyo[1,2], Paul De Bra[1], Geert-Jan Houben[1]

[1] Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O.Box 513, 5600 MB Eindhoven, The Netherlands

[2] Department of Computer Science
University of Twente
P.O.Box 217, 7500 AE Enschede, The Netherlands

Abstract

In this paper we present our view on how the current development of knowledge engineering in the context of Semantic Web can contribute to the better applicability, reusability and sharability of adaptive web-based systems. We propose a service-oriented framework for adaptive web-based systems, where the main goal is to help the semantic enrichment of the information search and usage process and to allow for adaptive support of user activites. In other words, our aim is to provide flexible information access, presentation and update to a broad range of users (individual and groups) in a personalized way within the context of pursuing a user's goals and performing tasks. We take an ontological approach to enable a shared understanding of concepts throughout the system and to provide semantic relationships between the information resources and the user's knowledge of (or interest in) them. We argue that the future of adaptive web-based systems lies in the modularity of the architecture and the openess to interoperate with other applications or compontens. To achieve this we adopt the concept of the UPML framework for semantic web service integration. Our ideas are illustrated in the context of the Token2000 project for Cultural Heritage in Interactive Multimedia Environments (CHIME) and show how combining elaborate AI strategies with the simplicity of hypermedia interaction can result in more easily applicable knowledge-based systems, or in more reasoning enhanced adaptive hypermedia systems.

1. Motivation

The more the corpus of information accessible through Internet grows, the more crucial it becomes to enhance the ways of finding, accessing and retrieving the right piece of information at the right time as part of our overall problem solving activities. This moves the focus primarily towards the provision of tools to support users to cope with the complexity of the information space and the dynamically changing user demands. Ideally we need a number of independent services which, when combined "on the fly", can support any type of activity of any type of users (on the various levels of granularity of their problem solving activities). Traditionally the field of Artificial Intelligence (AI) implements and successfully applies elaborate modeling approaches within the context of knowledge-based systems (KBS) in order to support users in performance their tasks [1, 9, 15, 25 and so on]. Although lately a lot of research effort is concentrated to decrease the complexity of the KBS and to open them up to various application domains, their design and implementation is still rather application dependent and their maintenance is a sophisticated and time consuming task. This subsequently obstructs their popularity and wide applicability. At the same time, the simple concept of adaptive hypermedia systems wins more and more interest in a short time. Their simple reference architecture [3, 7], aimed at a quick adaptive response, appeared to be very suitable for the web environments. On the other hand they lack the notion of solid knowledge and reasoning, which weakens their position within the context of adaptive systems. The simple modeling approaches appear not to be enough to assess the user's knowledge and to provide accurate adaptation [6]. Their current notion of "user's knowledge" does not cover various knowledge facets, which are important for the assessment of the user's knowledge level.

An ultimate goal within the current ubiquitous software environments, where the users are mobile, mainly web-based and interact simultaneously with various applications, is to allow for reasoning-based adaptation accross applications, see for example [24]. For this we need the simple concept of hypermedia and improve

the adaptation strategies using methods and techniques from AI-based systems. In order to achieve the interoperability across applications we need to offer an open and modularized architecture, which will be able to interact, exchange data and share components. The provision of semantically rich descriptions of the components' functionality and their internal formats is important in order to allow for interoperability among system components. Finally, maintaining a generic sharable (dynamic) user model is needed to serve as a communication point for the different systems [11, 12]. The biggest challenge here lies in the sharing, synchronization and interpretation of the user model. This way the user's behaviour within each system will be permanently evaluated and more detailed and richer user models will be achieved in order to allow for enriched adaptation and personalization of the content.

2.Background

If we look at the cultural heritage domain and specifically the one of Dutch national museums, we quickly realize that the most artifacts are inaccessible to the general public and experts distributed around the world. Museums own many more artifacts than they can show in their main exhibition at any one time. Large investments are being made to "capture" the artifacts digitally, and projects have been carried out to give broader access to the digitized material. There are two limitations to the current approaches, however. First, they focus only on a single type of user, e.g. novice user in the Rijksmuseum ARIA system or expert user in the Rijksmuseum AdLib database. Second, the system is unidirectional, i.e. "experts" input information into the system and "users" query this information. An important aim of the CHIME project is to offer ways to remove these restrictions and to allow information to be presented to a broad range of users in a suitable way and to allow users to add their own information to the repository, while respecting the integrity of the original historical sources. This allows a decentralized approach to the enrichment of the information in the repository by all its users and to the benefit of all its users. To achieve this we need to focus on providing adaptation to the different users' goals and characteristics, so that we can minimize the time and optimize the efficiency of achieving the goal for each user. Within the scope of CHIME project we focus on (1) tailoring the presentation of cultural information extracted from existing repositories to different types of users; and (2) allowing users not only to query the information database but also to add their own remarks (relevant multimedia data, such as figures, video material, photos, newspaper articles, spoken commentary) to the repository. In other words the central themes of the project are supporting different user performing different tasks, and providing functionality with respect to querying as well as modifying the repository. A central role in this is played by the *Modeling of the User* and the *Modeling of the Content* within a multi-task context (e.g. presentation generation, material searching, etc.). During the past decades we have been observing the success of different types of software systems, which adaptively support users in various activities, e.g. Expert Systems (ES), Intelligent Tutoring Systems (ITS), Information Retrieval Systems (IR), Adaptive Hypermedia Systems (AHS), Web-Based Information Systems (WIS) [4, 5, 14, 20, 21, 26, 27]. The observed problem is that most of the AI systems for user modeling and ITS are built in a very application dependent manner and the process of their developement is time consuming and not oriented towards sharing [17]. On the other hand IR systems propose useful and precise techniques to retrieve data, but they do not consider the application of user features. Finally, AHS and WIS are primarily targeting the adaptation and personalization to the user needs and goals, but they lack the sophisitcation of the the IR and UM techniques and the precision of the user input. Thus, we argue that an interdisciplinary approach would be most beneficial, alllowing us to combine elaborate knowledge acquisition and user modeling techniques from AI and user driven design from HCI and to apply them within the context of adaptive web-based systems.

Currently research in the area of Semantic Web, originating primarily from the knowledge engineering and artificial intelligence fields, with a special focus on ontologies and Web Services, provides a number of standards and accompanying solutions which can be used to achieve the above mentioned requirements. On the one hand we have the notion of ontology, which plays a role in facilitating the sharing of meaning and semantics of information between different software entities. A number of representational formats have been proposed as W3C standards for ontology and metadata representation. The most current advances with OWL exploit the existing web standards (e.g. XML, RDF and RDFS) and add the primitives of description logic as powerful means for reasoning services. The next step in this process of opening the AHS architectures is made by applying a Web Service perspective on the system components. Web Services (also known as software agents) make use of the above mentioned semantics and offer means for flexible composition of services (system components) through automatic selection, interoperation of existing services, verification of service properties and execution monitoring. They appear to be a useful solution for achieving the modularization. We can reach reasonable automatization and dynamic realization of the main aspects of web services (e.g. web service location, composition and mediation) by extending them with rich formal

descriptions of their competence (in standardized languages such as RDF or OWL). This way we can allow adaptive web-based systems to reason about the functionalities provided by different web services, to locate the best ones for solving a particular problem and to automatically compose the relevant web services for dynamic application building.

Within the context of the CHIME framework we exemplify how the use of ontologies and Semantic Web open standards can be benefitial for the improvement of the adaptation and the interoperability among internal and external system components. We also aim at (1) enhancing the interaction between system agents and providing richer semantics for the adaptive support of various user types, and (2) standardization of user modeling and adaptation in order to enable shareability and interoperability among various adaptive web-based systems. We take an interdisciplinary perspective and show how to enhance existing adaptive hypermedia systems with elaborate AI reasoning methods in order to improve the user's adaptation. This is the first step towards defining a new class of Intelligent Hypermedia Environments (IHE) as a crossing of AHS and ITS.

The remainder of this paper is organized as following. In Section 3 we introduce our architectural considerations for CHIME framework. In Section 4 we position our research in the context of related projects and intiatives, and finally in Section 5 some conclusions and future work are presented.

3. CHIME Architectural Considerations

The CHIME system can be viewed both as an Adaptive System and as a Hypermedia System, which both belong to the broader class of Knowledge-based Systems (KBS). The KBS perspective gives us a basis to develop CHIME as an adaptive (hypermedia) problem-solving environment. In order to achieve this we target a *modular* system architecture of *reusable components*, which supports the *shareability* of CHIME components as well as the use of *third-party components* within the CHIME system. We aim at *standardizing the protocols for message exchange* both between internal and external components. Another important requirement for our architecture is to allow for *scalability* in terms of multi-user support within a multi-task context.

In order to achieve the *modularity* we propose a multi-agent architecture, where both human and software components are considered to be agents. In order to support the *shareability* of CHIME components we follow the current Semantic Web notion of software agents in terms of Semantic Web Services. They allow discovery, configuration and management of agents. Next to this, the ontological engineering offers methods and technologies for adding semantic descriptions to content, functionality and dataflows, in order to allow for the discovery, configuration and management of internal and external agents. The basic idea is that by augmenting encapsulated system modules with rich formal descriptions of their competence we can further improve and also automate many aspects of the system management. Furthermore, by introducing dynamic shareable user model we also enable the inter-system interactions, shareability and reusability of modules. By applying open standards for the realization we secure the interoperability and the wide applicability of the adaptive web-based systems. Exsiting web service frameworks, e.g. the Internet Reasoning Service (IRS-II) introduced by Motta et al. [16], show how we can support the publication, location, composition and execution of heterogeneous semantic rich web services. It uses **UPML** (Unified Problem Solving Method Developments Language) for the specification of reusability in knowledge-based systems by defining how we can build elementary components and how these components can be integrated into one whole system [8]. The IRS-II approach enables us to support *capability-driven service invocation* (e.g. find a service that can solve problem X) because of the explicit separation of *task specifications* (the problems which need to be solved), *method specifications* (the ways to solve problems), and *domain models* (the context in which these problems need to be solved).

In Figure 1 we illustrate our idea in the context of high-level services in the CHIME architecture, adapted from (Motta et al., 2003). It alters the well known adaptive hypermedia reference models, e.g. AHAM [3] and the Munich Model [13], by introducing the notion of services and semantic description of functionality in terms of ontologies. The CHIME architecture distinguishes between the following components at the highest level of abstraction:

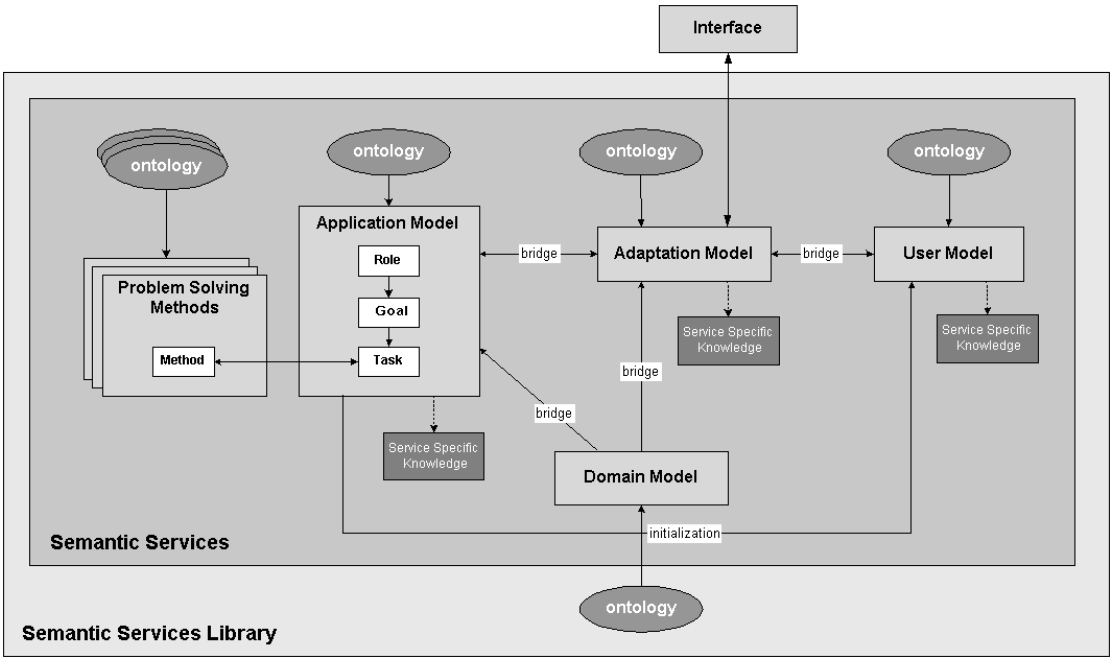


Figure 1: modular CHIME architecture (adapted from Motta et al. 2003)

- **Domain Model Service**, which is responsile for the explicit storage and description of the domain knowledge in terms of concepts of a Domain Ontology;
- **User Model Service**, which is an active agent following the user (inside and possibly also outside the system) in order to collect and further analyze data about the user's activities; this allows for inference of new knowledge about the user;
- **Adaptation Model Service**, which is an agent responsible for the application of rules to plan and perform the adaptation;
- **Application Model Service**, which contains a generic description of the user tasks in the context of Role-Goals-Tasks-Methods (Problem Solving Methods) chains. The Problem Solving Methods (PSMs) provide abstract descriptions of reasoning processes which can be applied to solve tasks in a specific domain. This way a clear distinction is made between tasks and methods. As a result flexible mappings between services and problem specifications can be made. Dynamic, knowledge-based service selection is also enabled in this way [16].
- **Bridges**, in accordance with the UPML framework connector definition [8], specify mappings between the different model services within CHIME framework.
- **Ontologies** play the role of shareable dictionaries in order to define and unify the system's terminology and properties to describe the knowledge of each CHIME service. Each service is specified by means of a corresponding ontology. This way a common ground for knowledge sharing and interoperability between CHIME agents (services) is provided. The choice of using open standards (e.g. XML, RDF, OWL) helps standardizing and formalizing meaning and enables the reuse and interoperability in the context of WWW. Finally, it all leads to very modularized architecture with an enhanced service maintenance.

A central role in CHIME is played by the Application Model Service. In the interaction with the application each user is represented by a particular role (e.g. guest, expert, student, teacher). This role defines for her a corresponding behavior in terms of the goals to achieve. In order to accomplish these goals the user applies appropriate tools (applications or agents), which provide one or several corresponding problem-solving methods (PSMs). Each of these applications maintains additional information about the user-system interactions (e.g. in the form of tables with description of topology and initial probabilities of Belief Networks) in order to be able to monitor and further reason over each step in the entire process. For instance, when the user works with a selected application, which offers a particular PSM, every action she performs through the user interface is communicated to the Adaptation Model Service, which is responsible for selecting the adaptation strategies on the basis of the User Model, the Domain Model and the Application Model. When the decision about the next step is made the Adaptation Model Service sends this information together with the information about the user's actions to the User Model Service. The User Model Service updates the User Model with the new values. The user information is stored and a reasoning engine infers new knowledge from it and makes predictions concerning the user's future behavior. This new knowledge is

sent back to the Adaptation Model Service, which makes a decision about "the best" next information item to be presented to the user. The interface component presents this information to the user and her feedback is translated back to the Adaptation Model Service. This completes the main system loop, which repeats as long as the user interacts with the system.

4. Related Work

In this section we give a brief overview of the related research and studies which served as an inspiration to our approach.

The distributed notion of WWW influences among other areas also the software development process in the direction of intelligent software brokering. A major contribution in this field are the results achieved within the context of the **i-brow**^{3} project. It aims at providing intelligent reasoning services on the Web by integrating research on heterogeneous databases, interoperability, ontologies and Web technology within KBS. The objective to increase the level of support on the global information infrastructure and to hide the technological complexity of the underlying system is achieved by the provision of intelligent brokering services. As a result of this, an Internet Reasoning Service (**IRS-II**) has been proposed. It is a Semantic Web Services framework, which allows applications to semantically describe and execute web services [16]. Thus, a framework (**UPML**, similar to the **CML** developed in the CommonKADS project [23]) has been developed to describe modular and reusable architectures and components to facilitate their semi-automatic reuse and adaptation. It partitions the knowledge into ontologies, domain models, task models, and problem solving methods (PSMs) and connects them via bridges. Each of knowledge model types is supported by corresponding ontologies. The work on IRS-II has focused on (1) the integration of the UPML framework with current web service standards, and (2) the enabling of developers to semantically describe code (currently Lisp and Java) of web services. In the context of CHIME, the key design decision we made was to associate each PSM with exactly one web service although a web service may map onto more than one PSM since a single piece of code may serve more than one function. Problem-solving methods provide reusable architectures and components for implementing the reasoning part of knowledge-based systems. We adopt the ideas of i-brow^3 and IRS-II about semantic web services also within CHIME, and thus apply the notion of reasoning services within distributed web-based systems and use the UPML language to specify the components and their relations.

Another approach in this direction is presented in **SOAR**. It offers a general cognitive architecture for developing systems that exhibit intelligent behavior [22]. SOAR defines a single framework for all tasks and subtasks (problem spaces), a mechanism for generating goals (automatic subgoaling) and a learning mechanism (chunking). Next to this a single representation of permanent (productions) and temporary knowledge (objects with attributes and values) is given. All decision computations are made through the combination of relevant knowledge at run-time. A desired state for CHIME is to use all the available knowledge for each task that the system encounters. Unfortunately, because of the complexity of retrieving the relevant knowledge this goal is not our focus, as with the increase of the knowledge body, the tasks become more diverse, and the requirements in system response time become more stringent. The best that can currently be obtained is an approximation of complete rationality and we consider the SOAR design as an investigation of one such approximation.

An important part of the modeling of user tasks, goals, roles and cognitive processes, is played by the learning theories. Two of the most applied ones are **ACT-R** proposed by John Anderson [2] and **Constraint-Based Modeling** (**CBM**) proposed by Stellan Ohlsson [19]. They are both based on the distinction between declarative and procedural knowledge, and the view that learning consists of two main phases. In the first phase the declarative knowledge is encoded and in the second it is turned into more efficient procedural knowledge [20].

Another aspect of modeling cognitive processes is given by existing modeling languages like Hank and UserML. **Hank** is a relatively new cognitive process modeling language, which is designed to be easy to grasp by non-programmers and powerful enough to build models of non-trivial psychological theories [18]. A central role in the modeling is played by the user model, which requires a protocol for encoding the information about the different users, and also makes it possible that any given adaptive system should be able to benefit from others sharing the same user model and that user modeling agents should follow you around [12]. The **User Modeling Mark-up Language (UserML)** offers an XML-based exchange language which is based on an ontology that defines the semantics of the XML vocabulary (UserOL). It provides a

modularized approach for module connections (via identifiers and references to identifiers) which allows for a graph structure representation [10]. It can be used as a protocol language between a User Model service and other services as well as the language for internal representation of user in the User Model.

5. Further Work and Conclusions

The next step in this research context is to select languages for the agent communication and to specify conventions for agent interaction. This will be considered with other participants of our common project. The further development of the CHIME system will involve application of the principles and techniques of already established examples of multi-agent systems. For example, AgentBuilder offers a good environment and tools for constructing intelligent software agents and agent-based systems. Next to this the Internet Reasoning Service (IRS-II) offers a flexible framework for the integration of web services. We can probably also find a good examples of architectures and infrustructures of web-services in IBROW, UPML, and so on.

Acknowledgements

The research work presented in this paper has been performed within the context of the CHIME Token2000 project. We would like to express our thanks to Guus Schreiber, whose comments about the CHIME architecture were very constructive and useful. We would like to also mention Mark van Assem, Lynda Hardman, Katya Falkovych and Frank Nack for there contribution on the general CHIME context and goals.

References

1. Aroyo, L., De Bra, P. (1999). Agent-oriented Architecture for Task-based Information Search System, Proceedings of the "Interdisciplinaire Conferentie Informatiewetenschap", pp. 94-98, Amsterdam, 1999.
2. Anderson, J.R. (2000). *Cognitive Psychology and its Implications.* Fifth Edition. Worth Publishers, NY. - p.241.
3. De Bra, P., Houben, G.-J., Wu, H. (1999) *AHAM: A Dexter-based Reference Model for Adaptive Hypermedia*. In *ACM Conference on Hypertext and Hypermedia*, pp. 139-146, February 1999. URL: http://delivery.acm.org/10.1145/300000/294508/p147-de_bra.pdf
4. De Bra, P., Aerts, A., Smits, D., Stash, N. (2002) *AHA! The Next Generation*, In *ACM Conference on Hypertext and Hypermedia*, pp. 21-22, June 2002. URL: http://delivery.acm.org/10.1145/520000/513347/p21-de_bra.pdf<
5. De Bra P., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D., Stash, N. (2003) "AHA! The Adaptive Hypermedia Architecture". In *ACM Conference on Hypertext and Hypermedia*, pp. 81-84, August 2003. URL: http://delivery.acm.org/10.1145/910000/900070/p85-eiron.pdf
6. Brusilovsky, P. and Maybury, M. T. (2002). From adaptive hypermedia to adaptive Web. In P. Brusilovsky and M. T. Maybury (eds.), *Communications of the ACM* **45** (5), Special Issue on the Adaptive Web, 31-33.
7. Halasz, F. and Schwartz, M. (1994). The Dexter Hypertext Reference Model. Communications of the ACM, Vol. 37, nr. 2, pp. 30-39, 1994.
8. Fensel, D., Motta, E., Benjamins, V.R., Decker, S., Gaspari, M., Groenboom, R., Grosso, W., Musen, M., Plaza, E., Schreiber, G., Studer, R., and Wielinga B. (1999) "The Unified Problem-solving Method development Language UPML". ESPRIT project number 27169, IBROW3, Deliverable 1.1, Chapter 1., URL: ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/upml.pdf
9. Haake, J., and B. Wilson (1992). "Supporting Collaborative Writing of Hyperdocuments in SEPIA." *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative* Work, Toronto, Ontario, 1-4 November 1992.
10. Heckmann, D., Kruger, A. (2003) A User Modeling Markup Language (UserML) for Ubiquitous Computing". In *User Modeling 2003 Conference*, pp. 393-397.
11. Kay, J, R.J. Kummerfeld and P Lauder, (2002) Personis: a server for user models, De Bra, P, P Brusilovsky, R Conejo (eds), *Proceedings of AH'2002, Adaptive Hypermedia 2002*, Springer, 203 - 212.
12. A. Kobsa (2001): Generic User Modeling Systems. *User Modeling and User-Adapted Interaction* 11(1-2), 49-63. URL: http://www.ics.uci.edu/~kobsa/papers/2001-UMUAI-kobsa.pdf

13. Koch N., Wirsing, M. (2002). The Munich Reference Model for Adaptive Hypermedia Applications. De Bra P, P. Brusilovsky, R. Conejo (eds), *Proceedings of AH'2002, Adaptive Hypermedia 2002*, Springer, 213 - 223.
14. Koedinger, K. R., Aleven, V., Heffernan, N. T. (2003). Toward a Rapid Development Environment for Cognitive Tutors. 12th Annual Conference on Behavior Representation in Modeling and Simulation. Simulation Interoperability Standards Organization.
URL: http://nth.wpi.edu/pubs_and_grants/03-BRIMS-063.doc
15. Leake, D., Hammond, K., Birnbaum, L., Marlow, C., and Yang, H. (1999). Task-based knowledge management. In *Exploring Synergies of Knowledge Management and Case-Based Reasoning, Proceedings of The American Association of Artificial Intelligence (AAAI-99) Workshop*. Orlando, Florida: AAAI Press. - Pp. 35-39.
16. Motta, E., Domingue, J., Cabral, L., and Gaspari, M. (2003) "IRS-II: A framework and Infrastructure for Semantic Web Services". In *2nd International Semantic Web Conference 2003 (ISWC 2003)*, 20-23 October 2003, Sundial Resort, Sanibel Island, Florida, USA - in press.
URL: http://www.cs.unibo.it/~gaspari/www/iswc03.pdf
17. Murray, T. (1999). Authoring intelligent tutoring systems: an analysis of the state of the art, *International Journal of Artificial Intelligence in Education*, 10, 98-129.
18. Mulholland, P. and Watt, S. N. K. (2000). Learning by building: A visual modelling language for psychology students. *Journal of Visual Languages and Computing*, **11** (5), 481-504.
19. Ohlsson, S. (1994) "Constraint-Based Student Modelling. *Student Modelling: The Key to Individualized Knowledge-Based Instruction*". pp 167-189, Springer-Verlag, 1994.
20. Mitrovic, A., Koedinger, K.R., Martin, B. (2003) "A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling". In Proceedings of 9th International Conference on User Modelling 2003, USA, pp.313-322.
21. Page, L., Brin, S. (1998). The anatomy of a large-scale hypertextual web search engine. *Proceedings of the 7th Intl. WWW Conf.*, 107-117, 1998.
22. Rosenbloom, P.S., Laird, J.E., Newell, A. (editors.) *The Soar Papers: Research on Integrated Intelligence*. MIT Press, 1993.
23. A. Th. Schreiber, B. J. Wielinga, J. M. Akkermans, W. Van de Velde, and R. de Hoog. (1994) CommonKADS: A comprehensive methodology for KBS development. *IEEE Expert*, 9(6):28–37, December 1994.
URL: http://www.cs.vu.nl/~guus/papers/Schreiber94f.html
24. Slaney, M., Subrahmonia, J., Maglio, P. (2003). Modeling Multitasking Users. In *P. Brusilovsky et al. (Eds.): UM 2003, LNAI 2702*, pp. 188.197.
25. Sycara, K. and Zeng, D. (1995). Task-Based Multi-Agent Coordination for Information Gathering. In Working Notes of the AAAI Spring Symposium Series on Information Gathering from Distributed, Heterogeneous Environments. Stanford, CA, 1995.
26. Szolovits, P., Long, W.J. (1982). The development of clinical expertise in the computer. In P. Szolovits, editor, *Artificial Intelligence in Medicine*, pages 79-117, Westview Press, Boulder, Colorado, 1982.
27. Weber, G. and Brusilovsky, P. (2001) ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education* **12** (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems, 351-384.

Design criteria for preservation repositories

Frans Dondorp and Kees van der Meer
Delft University of Technology, DIOSE Betake Research Group, Faculty EEMCS
PO Box 5031 2600 GA Delft
{F.P.A.Dondorp, K.vanderMeer}@ewi.tudelft.nl

Abstract

What are the requirements for repositories aimed at long term preservation of digital information objects, containing static objects (documents) and dynamic objects (programs)? It is recognized that preservation efforts should be independent of current technology in order to survive technology obsolescence. This requirement is hard to meet.
In this paper current preservation efforts (projects and techniques) and relevant standards are discussed in relation to this requirement. A view on authenticity of digital objects is presented that leads to the requirement of dependence on the designated community that is to be recognized in the design phase when building repositories.

Keywords: *longevity, preservation, standards, authenticity*

1. Introduction

A landmark in preservation was the publication in 1993 of the book 'Preserving the present' [1]. At that time, regarding the problem of preservation of digital information objects, records of knowledge and memory, neither relevant questions nor possible answers were known. 'Preserving the present', based on research on what organizations were doing then to preserve their electronic documents, was meant as a first guide to what they should do. The publication of that book was extremely useful to draw attention to the problem of digital preservation.
Electronic data on the US census of 1970 were no longer readable and proved to be lost beyond repair. The e-mails on the financial support of the State of the Netherlands to the shipbuilding industry were untraceable and probably deleted. The same for student allowances. Old electronic documents could not be reproduced in their original lay-out. The electronic Domesday books of 1986 was nearly inaccessible - quite a difference from the historic original of 1086.
Before publication of this book, only few people had realized that there was a relation between these phenomena. Preserving the present alerted people to the problems of preservation: the fact that reuse and readability of electronically recorded information is not guaranteed even in the near future. It is a subject in the heart of the science of information.

Ten years after 1993, it is a good moment to look at the state of the art on preservation by presenting design criteria for repositories. The topic of preservation repositories has become very important. The value of digital records has grown enormously. So has the amount of electronic information, as is suggested by Varian and Lyman [2]. Two categories of digital records exist: static and dynamic. Information provided by static objects is stable: it does not change over time. Traditional textual documents in digital form are static objects. Dynamic objects on the other hand contain (possibly machine-specific) instructions to be executed and may provide interactive user interfaces. Programs are dynamic objects, as are documents containing scripts or macros. A growing part of the information that has to be preserved is dynamic.
Moreover, the developments, changes and improvements on functionality in programs for electronic information objects take place at a rapid pace. Compared to the speed of the progress of the industrial revolution the speed of change in the electronic revolution is inconceivable. Astonishingly, the increase in the 'speed of write' (Harnad's term) does not by itself lead to durable thinking on preservation of electronic information objects.
The different aspects of ageing of rendering equipment, program libraries, operating systems, data carriers and hardware needs collaboration of experts from different fields of expertise. This need for collaboration makes the problem not easier to manage.

The problem of digital archiving (or preservation of digital objects in general) can be formulated in design criteria for repositories, as well as functional requirements to the preservation process once such repositories are realized. The repositories contain (static) documents and (dynamic) program derivates, software. The

repository and the preservation process should be independent of computing platform, media technology and format paradigms (stated by Dürr and Lourens in [3]) to the highest possible extent while providing adequate preservation of valuable information objects for as long as possible under heavy economic constraints. Thus, standards need to be developed, used and maintained, and general concepts for information value including selection and authenticity (evidential value) need to be defined. These design criteria are hard to meet.
In this paper examples are given of projects in the area of digital document preservation. The more complicated object class of programs is discussed, relevant standards are listed and a discussion on authenticity is presented. From these pieces of the puzzle a generalization follows and conclusions are drawn as to which (abstract) design criteria have to be met in creating repositories suitable for long term preservation of digital objects.

2. Document preservation

What are organizations doing now to construct a repository for 'until Doomsday or five years - whichever comes first' (Rothenberg)? Several examples exist in which a repository was realized where the design questions were in our opinion thoroughly considered and written down in a detailed way.

E-mail

E-mail messages can be created, received or maintained in the transaction of business or the conduct of affairs and, in that case, may have to be preserved as evidence. The need to preserve e-mails has made itself felt for several years. Fortunately, not all of the about 1000 million e-mails that are produced each year have to be preserved. The well-documented David project [4] reports that the old structure of electronic e-mail archive may appear disorderly due to the sheer quantities of files; this draws attention to the metadata necessary to access the e-mail archive. Apparently, in relation to policy on records management, the construction of a folder structure for an archive to be transferred and the assignment of useful file names is a point of attention. Finally, different governments have given different answers to the question whether paper copies or electronic copies of e-mails should be preserved in the archive. The attachments are a different kind of element; there are also differences to how to deal with the electronic attachments.

For use in Dutch government agencies, the Digital Preservation Testbed has designed and developed a solution to preserve e-mail [5]. This approach aims to provide a practical means to either automatically preserve e-mail when it is sent, or preserve received e-mails at any time. The approach embeds a component in MS Outlook that converts an e-mail message into XML. This XML document is passed to a Web service that formats the XML file into HTML and forwards the XML file to a repository for storage. The HTML is passed back at Outlook and ultimately forwarded to the SMTP server responsible for sending. In this way, outgoing e-mail is automatically stored in XML and centrally formatted using a standard style sheet. Upon sending the user is required to enter metadata that is stored with the object.
The storage of outgoing e-mail is straightforward. All parts of the SMTP message are represented in the XML file that is stored. For received e-mails, the parts are separated into elements. Attachments and possible HTML body content is saved as separate files to which references are included in the XML file. A logfile is also included, as is the original SMTP message (a textual dump of all fields, including header information and encoded binary attachments), in this approach called 'transmission file'. The Testbed approach is a step towards storing messages in a standardized manner, using strict regulations on accompanying metadata, required trace information (logfiles) and redundant inclusion of attachments (both in encoded form (in the transmission file) and in decoded form as saved binaries). Using XML as storage format, the message body part for non-HTML formatted mail is preserved according to the opinion that XML is a future-proof format for textual objects. Preservation of binary attachments is a problem, as these objects can either be static or dynamic. Emulation might be necessary, as will be discussed in the section on program preservation. HTML formatted body content is saved to file, thus making it susceptible to obsolescence. Conversion to XHTML+CSS would make it more durable (as HTML is in danger of becoming extinct and XHTML is an XML application), but requires an extra conversion step that might be done at a later stage.
Basically the approach boils down to a migration technique. Redundancy is used as a safety net: the original message is included in the archive. If the XML packaging technique becomes outdated or gruesome, it can all be done again in some different form.

For web archiving a similar design could be used. The differences would be the transmission file (now a textual dump of a HTTP response) and the composition of metadata, as other contextual information is relevant. Binary attachments can be considered to have the form of embedded content such as Flash movies

that require a viewer to be rendered in the future. On a functional level the approach can be copied from the one proposed by the Testbed for preservation of e-mail. Once again the HTML body content is a problem (and once again conversion to XHTML+CSS might be considered).

Nedlib

Nedlib was the project of libraries, computer science organizations and publishers to design and set up requirements for a deposit system for electronic publications. The Guidelines have been published in the Nedlib report series [6]. This project aimed at preservation of publications for national libraries. What proved to be the major issues in this at this state-of-the-art project? They proved to be the vocabulary (a list of terms was issued!), the applicable standards were the subject of a thorough investigation, the strategy of emulation for maintenance purposes, the use of the OAIS model (see the section on standards), the metadata and its relations to the OAIS model, and of course the realization of a long-term deposit system. Interestingly, the results lead to an operational Deposit system, of which the results have been published, allowing refinement of the original ideas [7].

Cedars

The Cedars project [8] was carried out in 1998-2002 to establish best practices for digital preservation for UK Universities. Like the Nedlib project it was well thought out, had sufficient mass, and was based on research rather than assumptions; it lead to fundamental insight on the practice of preservation. The parties in Cedars (universities) form collections. Collectioning means selection. Selection means that information objects can be excluded for reasons of content (outside scope) or other reasons. This could be stated in a Service Level Agreement (SLA) regarding the types of information objects to be kept in the collection. Selection provides the Cedars organizations with a 'degree of freedom' the Nedlib partners do not have: as deposit libraries, these have the duty to preserve all information objects that form the national intellectual heritage. The emphasis in Cedars was on managerial aspects; technicalities seem to be treated as rather subordinate. The Cedars way of working is based on the OAIS model. The considerations on collection management and costs are valuable. A demonstrator has been built.

E-archive

The e-archive project of Delft, Utrecht and Maastricht [9] can be seen as an extension to the Cedars project. Its aim is to realize a workbench of electronic publications for decades. Again, the OAIS model is adhered to. The publications are put in an XML container, containing a standard identification, the original bitstream, the necessary viewer, zero or more conversions of the original bitstream, and various kinds of metadata. In this project, the business model of the e-archive with two times appraisal, requirements on data management and access, and a cost model are worked out in detail.

Generalization

The list of projects described is meant to be extensive nor complete. This short summary suffices to illustrate the general direction in which these efforts are going: towards a standardized 'archive architecture' based on the OAIS model, incorporating XML applications (such as XHTML) when possible. The aim apparently is technology independence through standardization: a generally applicable architecture using a standardized format for archival content. As these projects are built on a foundation of standards, the choice of standards to incorporate is the crucial cornerstone and therefore the weak spot.

3. Program preservation

The problem of preserving dynamic objects is a subproblem of preserving many object types: for e-mail it is hidden in the attachments and for web pages it is included by scripts and embedded players (such as Flash and Shockwave). Documents containing scripts or macros can also be regarded as dynamic objects: advanced techniques used in wordprocessing can turn a document in a object that is very hard to preserve.

In archiving digital objects, programs are by far the most complicated ones. Preserving such a 'dynamic object' requires the preservation of the runtime environment in which it is to be executed. This environment is crucial to the 'rendering' of a dynamic object.
A problem with this requirement is that it tends to be recursive: to preserve the program, all underlying layers

(operating system and hardware) have to be preserved as well. An executable compiled to run under MS Windows on an Intel platform requires both components to be preserved if the executable is required to run in the future. These components cannot be replaced by others: the executable will contain platform-specific machine code and OS-specific function calls. Preserving one Windows machine to preserve all Windows programs will not work as programs designed for Windows XP will not run on Windows 95 and programs compiled for Windows NT on a DEC Alpha will not run on an Intel machine. The recursion can be drawn further: how about peripheral equipment, network, documentation and required skills? What if a user, other than an experienced computer scientist, is confronted with a thirty years old machine under emulation, which was even then operated by trained personnel?
Two types of programs need to be distinguished. Programs that are enablers to the rendering of data ('viewers') are different types of objects than interactive objects (games for instance). The difference can probably best be illustrated by the degree of dependance on a specific computing platform when 'rendering' the information contained in the object.
A PDF document for example is a static object containing data to be rendered. To do so, a specific computing platform is not required: just a program that can interpret the data correctly. This viewer is a dynamic object of the relatively undemanding viewer kind: creating an emulator to preserve it does not compare to the cost of rewriting the viewer altogether. The virtual machine approach can also be used for this class: as relatively undemanding programs, a simple computing platform can be designed for which emulators can be created at low cost and for which such viewers can be programmed. Once available, access to these viewers (and thus to the data they can render) can be provided by creating the simple emulator. In this way, the cost of emulation can be reduced drastically. The UVC approach discussed further on has a similar design.
To play a level of Quake, more is needed than a graphical image produced on screen: the playing experience needs to be replicated, including sound, video effects (possibly requiring specific video hardware), input devices and speed of game play. Rebuilding such a game is a gruesome operation that might easily compare to the complexity of emulation of the computing platform. To preserve highly interactive objects such as games, emulation is probably the only solution. Virtual machines are no option here: as these programs are very demanding, a virtual machine would have to be so complex that it compares well to an emulator for the original computing platform.

Emulation is an essential strategy in preserving dynamic objects. Even though the costs are high, emulation may be feasible if a large amount of programs running on a specific computing platform need to be preserved. Only a single emulator would be required. This emulator is an extremely complex program. The computing environment in which the game was originally run has to be replicated in such detail that the game can be played in the same way as it could one generation ago. One may question whether Pacman, the well-known old computer game, is fun to play on a modern machine with a 2 GHz CPU. One can state that playing against a figure that moves with the speed of light on your screen is not how the game was intended.
This technique is mostly applied for games. For many platforms no longer in existence (mainly home computers and game consoles) emulators are freely available, quite often created by gaming enthusiasts. The success of these emulators is often referred to as a suggestion that emulation is a feasible approach to preservation. This success is relative: although emulation of a game system that is designed entirely by a single manufacturer might be possible, emulation of current mainstream 'office systems' is quite a different story. The latter category exists of systems that incorporate hardware designed by a multitude of manufacturers in many different configurations.

The first to propose emulation as a preservation strategy was Jeff Rothenberg in 1999 [10]. The widely held discussion on the choice between emulation and migration following his landslide 'Quicksand' article has for a large part set the scene for the problem area of preservation. This discussion, also known as 'Rothenberg vs. Bearman' as David Bearman replied to the 'Quicksand' article with a now equally famous critique [11] seems to have ended in a tie: most researchers seem to feel that neither one approach is feasible to solve all problems. From a certain point of view, the difference between the two boils down to the difference in costs between computer power and storage capacity [9].
Migration and emulation can be seen as two dimensions of one plane. Every solution (a point in the plane) can be regarded as a combination of the two extremes of complete migration and complete emulation. If objects are migrated (converted) at regular intervals to keep up with technology, emulation is not necessary. On the other hand when a 'complete' emulator is build to provide an environment for the original viewer, migration is out of the picture. As migration has high variable costs (it has to be done for each object at regular intervals) and emulation is extremely costly in development and maintenance due to its complexity and has to be repeated for each legacy platform to be 'projected onto' each future platform, optimization by combination seems to be the best way to go.

Such a combination is proposed by Raymond Lorie [12]. His Universal Virtual Computer is for a large part based on emulation, and the entire approach ends in a migration step.
The idea is to design a small and very easy to implement computer. This computer is implemented on each future platform (at relatively low cost, due to its simple design). In this way, a rather inexpensive 'emulator' is provided to run UVC programs. By standardizing the UVC design, it is guaranteed (or expected) that UVC programs do not have to be changed (or recompiled as the case may be) in the future. The second step is to build a UVC program for each format to be supported in the archive. This program 'decodes' a format into a logical representation that can be understood by future users - a migration step. In the future viewers can be built to render this representation.
The UVC is currently being developed and will become operational in the electronic deposit as it is in development at the Royal Library of the Netherlands [13]. It is included in this project as a last resort: once document viewers can no longer provide access to legacy formats, the UVC approach will be used to provide long term access to images of document pages.

Source code

When discussing program preservation, two types of objects can be considered as input of the preservation process: compiled executables and source code. As it is (very) likely that only compiled programs are available to the repository, the most probable option to program preservation is emulation. If the source code is still available, one could argue that the expense of designing a verifiably correct emulator could be saved by re-engineering the program to run on a future computing platform. In simple terms: 'just' re-compile using a more current compiler for a more current platform. Attractive as this may sound, there are still a few complicating issues to deal with.
To start with, code is written in a specific programming language. Even though such languages tend to be standardized (the computer language C is the most obvious example: it is ISO standard 9899:1999), there are few guarantees that a program written for a specific runtime environment can be compiled without problems for another. Programming libraries providing access to platform specific features may differ significantly. Functionality on the level of the operating system may not be available in the same way if available at all. Imagine a program designed to run on a Windows environment that has to be compiled for a future UNIX-like environment. These systems differ significantly. Reconstructing (in software engineering called 'porting') the program is not a trivial task.
To allow for programs to be ported, the source code needs to be well documented and written in a language for which compilers will still be available in the future. If this is not the case, code may still be portable if the programming paradigm does not differ between the language the program was written in and the language to which it is to be ported.
Between language classes of the same paradigm code can be 'translated'. It requires a skilled programmer with expertise in both languages to assert the validity of the translation. The effort of translating code to another language class (for example from a logical language like Prolog to a functional language such as Miranda or to an object oriented language like C++) equals or exceeds that of redesigning the complete program.

These drawbacks illustrate the complexity of reconstructing software, but in some cases this approach may be preferable to emulation. The execution speed and the possible integration of the reconstructed program in existing systems are the most obvious. The end-user will be provided with a program suitable to execute on a current platform and will require no or little additional tools to do so. Problems regarding peripheral devices and user interfaces are dealt with adequately: instead of having to work with ancient text-based interfaces, the user is provided with the modern graphical interface he/she is more used to. Even though the effort required might be comparable to emulation or re-engineering, reconstruction of software might be the preferable preservation technique in situations where a large user community is planning on using the program frequently for years to come.
As this technique requires specific (possibly legacy) programming expertise, this is not a task suitable to be accomplished by repositories. It might even be argued that it is not a preservation technique at all, as the information object (the program) is altered drastically. Yet it is a way to provide access to information structures (such as databases) on abandoned platforms that might otherwise be lost forever. ThereforAn example of a restoration of a program was the restoration of E-plot [3]. The restoration of this program was necessary as its results are used for a widely used reference model. The program was originally written in Fortran and C (to run on an IBM-RT using AIX as operating system) and was dependent on specific source code libraries in use at the time of development. In the article 'programs for ever' the authors describe in detail the complexity of reviving software no longer maintained and stress the importance of preservation of scientific software to allow for preservation of scientific data sets. It proves to be possible to reconstruct old

software to execute on a more modern platform. Again, the use of OAIS AIP's proved to be applicable. The result is in a way medium independent and platform independent.

Generalization

Programs are designed to be executed in a specific runtime environment. Unlike 'static' documents that are nothing more than chunks of data independent of computing platform (as they do not contain machine specific instructions), the functionality of programs is dependent on machine specific parameters. Technology independence is hard to achieve when objects are designed to be technology dependent. Standardization is no longer the remedy of choice. For existing platforms, combinations of hardware and software, these runtime environments cannot be standardized as this would result in 'freezing' technology and disallowing innovation. For abstract platforms standardization is possible. This is the approach used by virtual machines such as the UVC: technology independence by introducing a standardized abstract machine that is to be emulated on existing platforms.
As there are several ways in which digital objects can be used, different preservation strategies are applicable to different types of objects. Even though emulation and migration can be applied to every object type, feasibility and costs are the determining factors in choosing strategies. It is possible to migrate an executable to another platform (by 'translating' the instruction stream), but the costs may be higher than building a general emulator. Emulating a platform to run a viewer for an old format version of software still in use is more costly than allowing for the current software to convert old formats.
Program preservation is a problem that can only be tackled by emulation or reconstruction, due to the nature of programs as instruction streams. The complexity of the emulation solution can be reduced by using virtual machines: this solution is however only feasible for relatively simple programs (of the 'viewer' type) that have to be compiled especially for the virtual machine at hand. To allow access to existing legacy software of a more demanding nature (games), or for which the reconstruction for a modern platform or redesigning/recompiling for a virtual machine is not feasible (i.e. cheaper than building an emulator), 'pure' emulation of legacy platforms is the only possible way to (re)gain access in the future.

4. Standards

Reuse of information objects demands agreement on all aspects of the information objects themselves as well as anticipation on the possible uses of the information objects. These agreements have partly been put down in standards. Partly, because standards have advantages (enhancement of the usage of common tools, enabling the reuse of experts experience) but also disadvantages (they deprive a user of some freedom to optimize a solution to his/her preference, and they take time of qualified staff). In order to discuss design desiderata of a durable repository of information objects, an inventory of standards is presented. Standards have been designed mostly for reuse of information objects independent of distances. Everyone should (under conditions) be able to reuse them.

XML and relations

The information objects are often structured according to the Extensible Markup Language, XML and it relations. Occasionally, domain specific derivatives are found, like MathML, WAP (wireless), XLS (location-based services). Data type specific derivatives include SVG (Vector Graphics) and SMIL for streaming media. Relations are xmlns (namespaces), the Resource Description Framework RDF for content specification. Moreover, XML is the basis for the lay-out structure by the Extensible Stylesheet Language XSL (more precise: XSL Transformations XSLT and the navigation mechanism XPath); other members of this family need not to be mentioned here. The popularity of XML with its derivatives is very impressive.
The great news of XML is that it is self-descriptive, a valuable property for preservation. If in the future a part of an electronic object is found without head or tail, and it contains structures like <Tag>Value</Tag> (to be recognized at byte level), then it is XML or at least HTML. From the name of the tag (when standardized or chosen carefully) the meaning of the tag content can be deduced and the value can be interpreted correctly. This way, a structure and a part of the semantics present themselves. Structures with attributes like <Tag Attribute="AttrValue">Value</Tag> can be interpreted in the same way.
The bad news about XML is that its longevity is not ensured. XML itself is the successor to SGML, ISO standard 8879:1986, its relation XSL is derived from DSSSL, ISO standard 10179:1996, the companion to SGML and XML is a successor to ODA, ISO standard 8613:1986. SGML and XML are not fully compatible. The future of SGML looked bright once, just like that of XML does now. XML is known to have drawbacks. An example: XML files are big and clumsy for location based services. Will there be a successor to XML,

named Enhanced XML - Improved Technology! (EXIT!); and if so, what shall be the future of XML files?

Presentation

For presentation PDF is often used. PDF is not an open standard; it is owned by Adobe. That makes this standard vulnerable for economic incidents. An initiative has been reported by Boudrez et al. in which it is tried to realize a PDF subset for archiving: PDF/A. In PDF/A the targets are as autonomously as possible. External dependencies as encryption, compression methods (that could be proprietary), copyrighted character sets, references to external files, encapsulation of executables etc. are being avoided. The alternative to PDF is the XML partner XSL; occasionally HTML and CSS (Cascading Stylesheets, a companion to (X)HTML) are mentioned. Both XML and PDF are often mentioned as acceptable formats to deliver information objects to the end-user: the output of the preservation process.

OAIS, Open Archives Information System, ISO standard 14721:2003

The OAIS model is a reference model for a system for archiving information, both digital and physical, with an organizational scheme composed of people with the responsibility to preserve information and make it available to a designated community. Firstly, it describes at a high level the processing of information objects. The acceptance procedure, called ingest, describes the processing of Submission Information Packages (SIPs). Also it enables the process of keeping and preserving Archival Information Packages (AIPs), and the delivery to the end-user of Dissemination Information Packages (DIPs). The OAIS model enables to define task structures for the electronic archive in the form of workflow processes. Secondly, the OAIS model contains an anticipation to the future users of the information objects. It is being presented under the term of 'designated communities'. A description of the designated communities enables to state what information objects will have to be kept, and what quality conditions apply.

US DoD 5015-2, MoReq and ReMaNo

They are meant for software specifications for record management applications. The US DoD (Department of Defense) 5015-2 Standard is a set of requirements. It is well known and proves to be in accordance to electronic records management. MoReq, MOdel REQuirements for the management of electronic records, is its up-to-date EC equivalent; ReMaNo (Softwarespecificaties voor Records Management Applicatie voor de Nederlandse Overheid) aims at the same goal but is based upon the Dutch law of Archives. These standards define aspects like control and security, acceptance, folder structure, retrieval, appraisal, selection, retain time, transport, destruction, access and presentation, administrative functions and performance requirements.

Records management, ISO standard 15489:2001

The ISO standard on Records Management is the successor to the Australian AS 4390 standard. In a way, it is a well established standard: many people have expressed ideas about records management, and applied the idea that if the costs to keep records exceed the damage if the records have been disposed of, is a basis principle for records management. The standard addresses policy and responsibilities defined and assigned throughout the organization as well as the records management requirements authenticity, reliability, integrity and usability.

Retrieval languages: OAI-PMH and ANSI Z39.50

In distributed systems, in order to find preserved objects, all kinds of query systems can be used. When several collections are coupled or when multiple copies of objects are stored at different locations (the LOCKSS principle - Lots Of Copies Keep Stuff Save), a mechanism is needed to retrieve information about collection contents in order to search for objects. In the Internet, a well known technique is harvesting: retrieving information by having an automated process retrieve information from data publishers at regular intervals. A result of the Open Archives Initiative (OAI) was the building of the Protocol for Metadata Harvesting (PMH). An archive willing to disseminate their content through the web can open up its electronic archive for harvesters. A harvester of a service provider contacts the archive and retrieves records containing metadata about the objects archived. The service provider offers indexes and retrieval facilities based on these records to end-users. The OAI-PMH does not demand much expertise, less than the older well-known and more powerful ANSI Z39.50 protocol (and the corresponding ISO standard 23950:1998) that has been in use for over a decade.

Data carriers

Information objects have to be saved on 'data carriers' that can be read on all kinds of equipment. Quite a few standards have been established. As an example, the ISO working party of optical disk cartridges gives a list of 32 standards [14]. That, at least, is a witness to the aim for interoperability.

The article 'Overview of technological approaches to digital preservation and challenges in coming years' by Thibodeau [15] is an excellent overview on digital preservation.
However, his article seems to treat ICT standards as fixed entities, as boundary conditions. ICT and its consequences are rather more a variable than a fixed entity. The design of any system means balancing between needs and wants of users, technical possibilities, changes, disadvantages and risks. Also, forecasts on technical possibilities are often inaccurate. The expectations on Information retrieval of the general public and even some experts in the 1980's and 1990's serve as an example. Computers would make it possible to store all documents. It was expected that, once all documents would be stored electronically, full text retrieval would make it possible to find all known information. A complete mistake: the Stairs experiment [20] was the first to shed doubt on the expectation; in 1998 came Schwartz's sigh [17]: improvement on general-domain web search engines may no longer be possible or worth the effort!
IT aspects influence the design desiderata so pervasively that it cannot be 'sorted out' (Thibodeau) and must remain at the heart of the design desiderata.

Generalization

Standards enhance reuse of information objects independent of design environment. But reuse independent of time leads to a different view.
The nice thing of standards is, there are so many ones to choose from (a quote generally ascribed to Tanenbaum). However, from a longevity point of view there is not much choice. The long use of XML is disputable, as it may not live very long. In fact, most standards are blind to the teeth of time. The OAIS model generally adhered to is an exception. It demands to think of future users, although its guidelines are superficial. The standards on software specifications reflect the legal differences between nations on laws on archives. The standard on records management may be the best thing that ever happened to archives, but not all record creators use it well. Still that is essential for a costly repository. Many creators do not know the standard, let alone its consequences. The state of retrieval languages shows one more reinvention of the wheel: although OAI-PMH may be made compatible with Z39.50, it was not created as such. Chances are that enormous investments of libraries and archives and other memory institutions in Z39.50 may eventually be discarded. In the list of standards on data carriers at least relations between types of standards have been inserted, but it looks like the tower of Babel.
For longevity purposes, standards should be built and maintained as long-lived artifacts. One could draw design desiderata for long-lived standards, like: standards should not be too complex, too large and too 'fat'. For standards small is not only beautiful but probably also lasting: the motto 'less is more' certainly applies to standards. This kind of desideratum needs further research.

5. Authenticity

Authenticity of digital objects is probably the most debated preservation requirement. Obviously every object that 'comes out of storage' should preferably be authentic, 'real' and 'trustworthy'. As every computing application imposes different requirements on the objects it requires, authenticity in its broadest sense could be defined for each and every application differently. A digital repository designed to preserve objects of any kind requires a general notion of authenticity or at least an objective means to measure the result of the preservation efforts against the applicability or usability of objects once they are delivered after years of storage.

This research borrows two fundamental concepts from other disciplines. Firstly, the context-dependent interpretation of the 'copy' concept put forward by Paskin in relation to digital rights management [18]. He suggests that two digital objects are only to be considered identical within the same context (i.e. when used for the same purpose). The context of use is the determining factor in establishing the correctness of the copy, the 'sameness'. Properties of the object not of relevance for the purpose to be served are not necessarily copied. This interpretation of the 'copy' concept matches with its use in everyday life: an encoding of digital audio (in MP3 for example) is clearly a 'copy' of a copyrighted work used to serve the purpose of playing music at a reasonable level of audio quality. It is not a copy in the context of CD manufacturing, as in that

context the lost property of binary integrity is relevant. The concepts 'copy' and 'original' only have meaning in a particular context of use: in that context the original is obviously the input of the transformation (copy) process and the copy is the output. This is intuitive: a digital object cannot be a context-independent, 'absolute' original. The original information (the first manifestation of the information) is always lost: whether it is the performance of which the CD is the recording or the document typed in a word processor of which a copy was saved from memory to disk. Only information relevant for the object use is recorded or saved: not the expression on the artists face or the typing rate of the author. Note that a clear definition of the context replaces any physical or logical requirement to be imposed on the copy to assess its quality.
Secondly, from cryptography, it is recognized that messages sent between parties are considered 'secure' if their integrity, authenticity and confidentiality can be established and the procedures used are tamper-free (the requirement of 'non-repudiation'). In this application, authenticity means the requirement that the origin of messages can uniquely be established. This requirement of identification in this context suffices to establish authenticity.

These two building blocks provide all the concepts needed to build a conceptual framework to deal with authenticity.
The terminology used in the literature suggests which properties are relevant: what constitutes authenticity. Dollar for example states "authentic records are records that retain their reliability over time" [19]. The term 'reliability' refers to the authority and trustworthiness of records: they "stand for the facts they are about". Bearman and Trant suggest that authenticity consists of three 'provable' claims: the object is unaltered, it is what it purports to be and its representation is transparent [20].
Using the concepts borrowed from cryptography, relevant requirements are object integrity and identification. The requirement of non-repudiation is implied: Dollar's 'authority' and the 'transparent representation' mentioned by Bearman and Trant indicate the requirement of verifiably tamper-free preservation procedures. The fourth element in cryptography does not seem to be applicable: confidentiality of information conflicts with the purpose of preserving information for the public.
The requirements of 'trustworthiness' and 'authority' can be considered to be combinations of integrity and identification. If any of these two fails, an object is clearly not 'trustworthy'. An additional requirement is needed to assert whether an object can actually replace the original object in the process in which the original was used. This is the intrinsic value of the object: it always serves some purpose and if it no longer can do so it loses its value (and thus the reason to be preserved).
This requirement is taken to be 'authenticity': for a specific (identified) purpose, an authentic object achieves this purpose at least equally well as did the original object. More formally: within a certain context, an authentic object is a verifiably correct implementation of the functional requirements relevant in that context imposed on the original object. This context is the designated community from the OAIS model.

Complex issues regarding authenticity can now be answered. The answers might be surprising at first glance, but are logical expansions of the intuitive notion of authenticity. Two examples are given.
A legacy program, accompanied by a database, is preserved by an repository. The program contains the 'millennium bug' causing it to yield incorrect answers to queries. The repository has preserved the program bit stream flawlessly and is even able to provide a verifiably correct platform emulator (an achievement only possible in theory). Executing the program in 2003 correctly yields the incorrect results. The question arises which object would be the authentic one: the preserved bit stream or a debugged and thus altered copy (with the purpose of execution under emulation)? What purpose does the program bit stream serve if its execution is not without failure? If some researcher wishes to examine the program as it was run decades ago, this bit stream is the authentic one. In the more likely situation that the object is to be executed in order to obtain the correct answers to queries, the altered object is the authentic one.
The Night Watch by Rembrandt, one of the most famous paintings in the Dutch cultural heritage, is in its current form not even close to authentic. During its 360 years of existence, a part has been cut off, it has been 'knifed' by a museum visitor and it has been cleaned. It clearly fails to meet requirements of object integrity and the preservation process does not meet requirements of non-repudiation (as it allows the object to be damaged and altered). Yet thousands of museum visitors from all over the world flock to the Rijksmuseum to see 'the real thing'. For them, there is no question about its authenticity. For the purpose of looking at a painting by Rembrandt, the object stored serves this purpose at least equally well as the original object (in this case the same) did 360 years ago. For this purpose, authenticity is derived from identification: if it is the picture that Rembrandt painted, it is authentic. Any derivative (photo, sketch, drawing) is not. If Rembrandt had painted the picture twice, the second one would have been authentic for the purpose of attracting museum visitors, but not for the purpose of studying the cloth used in the first version.

These examples illustrate that preserving original bit streams and building computing museums do not

provide solutions to all problems regarding object authenticity. Authenticity is not the same as integrity, identification or originality. Terms as 'trustworthiness' and 'reliability' (a term broader than reliability in computing architectures) are too subjective to allow for practical assessments. The reason why in cryptography the terms 'authenticity' and 'identification' are interchangeable is that in those systems the purpose of the messages sent is achieved 'just' by identification of their origin.

Preserving digital objects to keep them 'available', i.e. to allow for future use of the object, imposes functional requirements on the repository. As authenticity is context dependent, the context in which the object is to be used in the future needs to be described in as much detail as possible. This context allows for the identification of what features of the object, which functionalities, need to be preserved. If only the text of newspaper articles need to be preserved (future users will only need to read the information contained and search for strings), it suffices to store text files in Unicode, which is cheaper and less difficult than storing the articles as PDF (for example). If the requirement allowing for textual search is dropped but graphical lay-out is to be provided, optical scans stored in BMP could be stored. To allow for both, both can be stored.
Reducing authenticity to a set of functional requirements seems to be an obvious, somewhat belittling approach as one is tempted to store the object as it is today and engage in all kinds of difficult technical approaches to keep it accessible, convinced that the original object will always be the authentic one. As stated earlier, no object can be authentic for each and every unforeseeable future purpose.

The most important consequence of the concept of authenticity as a context-dependent aspect of objects in storage is that it can (and should) be made explicit as a set of functional requirements that are negotiated upfront, prior to storage. A result of this negotiation would be a service level agreement (SLA) of sorts: a document serving as a contract, exactly describing what preservation efforts are to be expected from the repository and, partly as a result of these, what functionality can be expected of stored objects once they are delivered in the future. Such a 'preservation effort agreement' (PEA) can be the basis of quality assessment after delivery and, in its quality as a contract, a basis to solve disputes once objects do not meet requirements. Such negotiation upfront solves a lot of issues regarding vague and subjective (and therefore unquantifiable) requirements of 'authenticity'. A list of functionalities and quality indicators, the PEA is unambiguous. Furthermore, it connects well to the SLA which has been part of system development and maintenance for years. As digital preservation may itself be part of a larger information system, the PEA could prove to be a valuable quality indicator as part of a larger SLA.

Another problem with storing originals is that no file format lives forever. Preservation techniques might change the object to keep the information it contains available (migration) or provide access in a possibly reduced form by providing a virtual computing environment (emulation). Neither technique can provide warranties that an object stored today can function in the exact same way in the future: probably something, some functionality, will be lost. It seems to be logical to assure oneself that the functionalities crucial to the object's use within a certain context are not among the functionalities in danger of getting lost: hence the formal specification of functional requirements upfront. If these requirements are not made explicit before collections of objects are ingested, design choices in preservation techniques or restrictions on migration possibilities might cause irreparable restrictions for future use. They might even render objects entirely useless for their designated communities.

An example taken from a current preservation project for e-mail proves this point. The strategy adopted was to convert e-mail messages in textual form to XML. In the specific case of 'raw' textual messages this can be done rather easily as the fields used in the SMTP protocol are fixed in amount and the structure of an SMTP message is very suitable to be captured in XML. As it turned out, the conversion process did not allow for so-called HTML-mail: messages with an HTML document as body. Style elements were lost as the body was reduced to its textual content. This is a restriction of functionalities that might be relevant for the future user. Implied by design choices made for preservation strategies, in a worst-case scenario these invisible restrictions would only noticed after years of preservation, when it is too late for repair.

The weak spot in reducing preservation efforts to a set of functional requirements is the necessity to identify the 'designated community' and, more importantly, identify its needs. It is impossible to know upfront what future users will expect from archived objects and how they will use the objects. This is a problem that obviously cannot be solved before the invention of time travelling. As one cannot give more than one has, regarding object quality one can only store objects at the quality they are now. If that quality is reasonable for us, it will (have to) be enough for any future user. Guarantees on authenticity and quality of preservation can only be given by explicitly formulating what constitutes that authenticity and quality for a particular object in a particular context at the time of ingest.

Generalization

Authenticity is clearly a central issue in preservation. On the one hand it defines the quality of the preservation efforts of the repository and on the other hand it defines the objects usability or applicability for the user. As it is the user who will assess both, it is imperative to include the intentions of that user in the authenticity requirement. Practically speaking, the authenticity requirement needs to be regarded in the context of the objects purpose and use. Catched in a catchphrase: 'authenticity is nothing without purpose'.
The design criterium that results from the presented view on authenticity is that of goal dependence. Where preservation, as stated, should be independent of technology, it should be dependent of the designated community, in OAIS terms. This means that the intended future object use should be considered when designing a repository. Illustrated in the previous section, this requirement cannot simply be ignored. If the designated community is not taken into account, stored objects have to be authentic for everyone and every purpose. As shown, this is an unrealistic requirement.

7. Conclusions

Digital information objects in digital repositories should last until Doomsday or until they are no longer useful - whichever comes first. This means that preservation efforts have to be technology independent in order to survive technology obsolescence. This technology independence can partly be realized by standardization: adhering to the OAIS model and choosing XML as intermediate file format are design choices common to most preservation projects in current development.
For dynamic objects such as programs or documents containing 'active content', standardization is only partly applicable. As these objects contain instructions to be executed within a particular runtime environment, this environment needs to be preserved or recreated in order to preserve the object. Technology independence is hard to achieve here, and can only be realized when using virtual machines to provide the runtime environment. This approach is only feasible to preserve dynamic objects that are logically independent of specific hardware (such as viewers). For other dynamic objects (such as games) or legacy software for which reconstruction or recompilation is too costly or impossible, emulation is the only possible solution: temporary technology independence by projecting one computing platform onto another.
Whether emulation or migration will prove to be the most successful remains to be seen. Most likely, every preservation problem for every digital repository will have the choice on the degree to which they will be combined.
Using standardization to achieve technology independence does not result in time independence. Unfortunately, ICT standards do not seem to last and chances are that the standard of choice today will be abandoned tomorrow. When designing preservation repositories using standards as a cornerstone, it is imperative to recognize this weak spot.
Authenticity of digital objects is determined by object purpose. Asserting the authenticity of stored objects requires taking the designated community, the future user, into account. As authenticity determines the value of objects stored and authenticity is dependent of the objects use and purpose, 'purpose dependence' should be taken into account when designing repositories. This dependence could be made explicit by using a Preservation Effort Agreement that serves as a contract containing functional requirements the stored objects have to meet after years of storage.
In order to build repositories that are and will remain useful, technology independence has to be achieved. To allow for 'purpose dependence', clear and well documented functional requirements have to be defined prior to long term storage.

References

All URL's are checked and valid in September 2003.

1. T.K. Bikson and E.J. Frinking: Preserving the present / het heden onthouden. SDU, The Hague, 1993.
2. H. Varian and P. Lyman: how-much-information?
3. http://www.sims.berkeley.edu/research/projects/how-much-info/ (September 2003)
4. E. Dürr and W. Lourens: Programs for ever. In: P. Isaías: Proceedings on NDDL 2002, Ciudad Real, 2002. pp. 63-79.
5. Digitaal Archief Vlaamse Instellingen en Diensten, DAVID.
6. http://www.dma.be/david/
7. ICTU: Bewaren van email. 2003

8. http://www.digitaleduurzaamheid.nl/bibliotheek/docs/bewaren_van_email.pdf
9. J. Steenbakkers: The Nedlib Guidelines. Nedlib report series, 5. Koninklijke Bibliotheek, Nedlib Consortium, 2000.
10. R.J. van Diessen en J.F. Steenbakkers: The long-term preservation study of the DNEP project. IBM/KB Long-term Preservation Study Report Series 1. IBM / Koninklijke Bibliotheek, 2002.
11. Cedars, Curl Exemplars in Digital Archives.
12. http://www.leeds.ac.uk/cedars/
13. R. Dekker, E.H. Dürr, M. Slabbertje and K. van der Meer: An electronic archive for academic communities. In: P. Isaías: Proceedings on NDDL 2002, Ciudad Real, 2002. pp. 1-12.
14. J. Rothenberg: Avoiding technological quicksand. CLIR report 77. 1999.
15. D. Bearman: Reality and chimeras in the preservation of electronic records. D-Lib magazine, April 1999.
16. R.A. Lorie: Long term preservation of digital information. ACM/IEEE Joint Conference on Digital Libraries, 2001.
17. http://www.informatik.uni-trier.de/%7Eley/db/conf/jcdl/jcdl2001.html
18. R. Lorie: The UVC: a method for preserving digital documents. IBM/KB Long-term Preservation Study Report Series 4. IBM / Koninklijke Bibliotheek, 2002.
19. ISO: Standards and guides on JTC 1 / SC 23:
20. http://www.iso.ch/iso/en/stdsdevelopment/tc/tclist/TechnicalCommitteeStandardsListPage.TechnicalCo
21. K. Thibodeau: Overview of technological approaches to digital preservation and challenges in coming years.
22. http://www.clir.org/pubs/reports/pub107/thibodeau.html
23. D.C. Blair en M.E. Maron: An evaluation of retrieval effectiveness for a full-text document-retrieval system. Commun. of the ACM 28 (1985), 289-299; D.C. Blair: Full-text retrieval: evaluation and implication. Int. Class. 13 (1986), 18-23; D.C. Blair en M.E. Maron: Full-text information retrieval: further analysis and clarification. Info. Proc. Mgmt. 26, (1990), 437-447.
24. C. Schwartz: Web search engines. J. Am. Soc. Info. Sci. 49 (11), (1998), 973-982.
25. N. Paskin: On making and identifying a "copy". D-Lib magazine, January 2003.
26. C.M. Dollar: Authentic electronic records. Cohasset Associates, Chicago. 2002.
27. D. Bearman and J. Trant: Authenticity of digital resources. D-Lib magazine, June 1998.

A case for incorporating vague concepts in formal information modeling

Sander Bosman, Theo van der Weide
Computing Science Institute, University of Nijmegen, The Netherlands
{sanderb,tvdw}@cs.kun.nl

Abstract:

This paper gives a fundamental overview of the information modeling process in the context of requirements engineering. From this we propose an extension to conventional modeling techniques by introducing so-called vague concepts.

1. Information modeling

Information modeling is one of the main tasks during requirements engineering. Its result is a concise overview of concepts and their relations as they occur in the application domain under consideration (Universe of Discourse, UoD). This overview is called the information structure, and can be seen as a model of the UoD. During information modeling, two roles can be distinguished, referred to as domain expert and system analyst. The specification of the information structure forms both the basis for and the subject of understanding and communication between domain expert and system analyst.

In this paper we focus on natural language based modeling techniques. Such techniques aim at modeling how the UoD is communicated about; the resulting information structure is called the information grammar. Examples are NIAM [8] and PSM [6]. From the information grammar the concrete information structures are readily derived. Note that other information modeling techniques (such as UML [1] and ER [2]) focus on this concrete information structure, thereby abstracting from the linguistic packing of the information. However, conserving the linguistic packing (information grammar) allows a discussion in terms of the natural concepts in the UoD.

Our aim is to propose the introduction of vague concepts: concepts which are recognized as important in the modeling process, but yet have no complete and formal specification of meaning. The intention of the modeling process is to construct a precise (as opposed to vague) specification. During modeling, vague concepts will be subject to further refinement.

Before introducing these vague concepts, we first take a fundamental look at the modeling process as consisting of two main activities: 1) providing domain knowledge, and, 2) processing (modeling) the provided knowledge. For convenience, as mentioned above, we will assume these actions will be performed by two separate persons: domain expert and system analyst. The aim is to work towards a method that lets a person create a formal model by starting with vague informal descriptions, and incrementally making these more formal until a clear, consistent and precise model results. Finally, we draw some conclusions.

2. The modeling process

In this paper, we are interested in the modeling process as an interaction between domain expert and system analyst. In this section, we will first focus on general communication between human beings. Then, we will discuss how information modeling can be seen as finding the information grammar of the communication between domain expert and system analyst.

2.1 Human communication of conceptions

Consider the situation where the domain expert watches the UoD and wants to communicate information about this UoD. As this involves processes that occur inside a human's mind/brain (and are therefore largely unknown), we adopt a rather abstract cognitive model of how this works (see figure 1), based on [3].

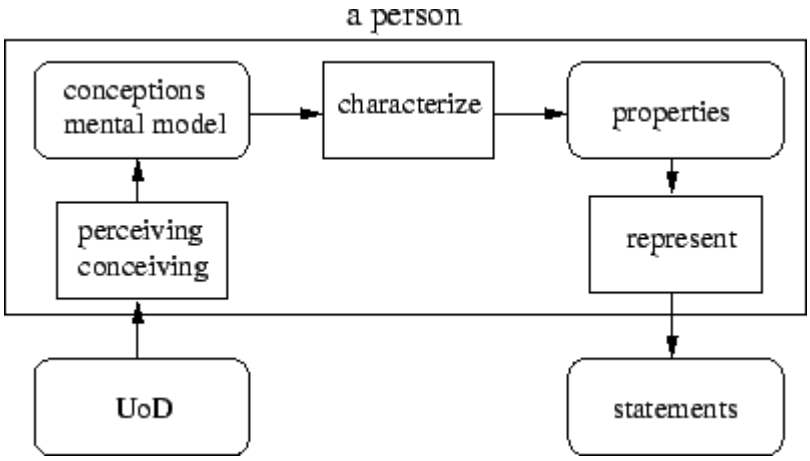


Figure 1. Communicating conceptions.

2.1.1 Perceiving and conceiving

We assume the Universe of Discourse to be perceived by the domain expert. The resulting perceptions, which initially can be regarded as raw data, are to be interpreted and elaborated on, resulting in conceptions as they are stored in the brain. Or, as taken from [3]:

Assumption 1 *Human beings are able to form conceptions in their minds, as a result of current or past perception, by means of various cognitive or intellectual processes, such as recognition, characterization, abstraction, derivation, and/or inner reflection. The collection of (relatively) stable and (sufficiently) consistent conceptions in a person's mind is called his or her knowledge.*

The collection of conceptions that describes (part of) the UoD is called the mental model of this UoD. Each conception can be said to model some aspect of the UoD, at some level of abstraction.

We do not assume that mental models are complete and consistent representations of the UoD. More typically, a mental model has a level of completeness and consistency good enough for its use.

2.1.2 Characterizing

To communicate about a mental model, we assume the domain expert derives relevant properties from conceptions in accordance with some goal (e.g., in response to a question of the system analyst).

Derivation of properties may also be influenced by the characteristics of the communication channel on which the properties will be represented. For example, in direct communication such as face-to-face speech it makes sense to communicate 'small' properties, allowing for interruption. When writing a document, a person will focus on properties on a higher level of abstraction.

2.1.3 Representing

Properties need to be represented in some language on a medium, in order to be communicated. A represented property is called a statement.

A common way for humans to construct statements is to formulate them in natural language. It has been postulated that verbal communication still dominates these other styles of communication (the 'telephone heuristic', [8]). However, true as this may be, verbal communication is not always the most effective communication modality. This is why we allow graphical elements to be part of the communication.

Since the creation of representation takes time, it is possible for the UoD and its mental model perceived by the domain expert to change while communicating statements. Suddenly a sequence of valid statements may become invalid. The communicating person then has to 'invalidate' the invalid statements.

2.1.4 Interpreting representations

On the other end of a communication channel, someone can perceive and conceive communicated statements. This on its turn creates conceptions, in the listener's mind, being the interpretation of the statements. We assume interpretation follows the opposite way of characterizing and and representing:

1. the person creates a sequence of properties in his mind, by perceiving and conceiving the statements.
2. the person tries to form a mental model of these properties, trying to 'make sense' of the properties. Thereby he will take the (probable) goals and context of the communicating party into account.

The two steps are not necessarily performed sequentially, as humans have a limited short term memory for storing statements.

2.2 The informal specification

In the remainder of this paper we will consider a domain expert communicating with a system analyst, who in turn creates a formal model of the UoD. This setting is shown in figure 2.

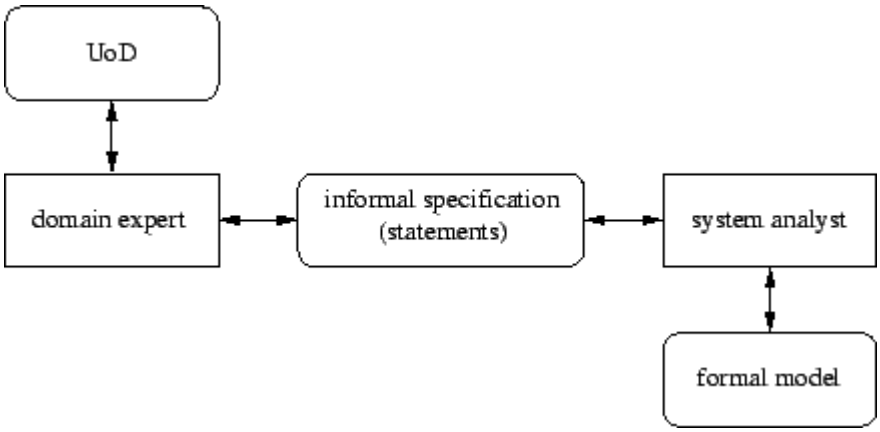


Figure 2. Setting

As discussed in the previous section, the domain expert perceives the UoD and has a mental model of it. The expertise of the domain expert consists of having thorough knowledge of the UoD, and being able to communicate this model. The domain expert does not need to have the skills of providing a well abstracted model [5].

The domain expert communicates statements about the UoD, to be interpreted by the system analyst. The system analyst, in turn, can respond with remarks (e.g., questions). This results in a sequence of statements, referred to as the informal specification IS:

$$IS = [s_1, r_1, s_2, r_2, \ldots]$$

where each s is a statement from the domain expert, and each r a response by the system analyst.

Goal of this communication is to create an informal specification that represents the mental model of the domain expert. We assume transitivity of the relation between UoD and mental model of the domain expert, and the relation between this mental model and the informal specification. This allows us to view the informal specification as a model of the UoD.

2.3 Modeling by communication

Modeling is the process of creating a model. There are many definitions of what a model is. For example, the definition used in [3] is:

> A model is a purposely abstracted, clear, precise and unambiguous conception.

We will use the term in the following way:

Definition 2.1 *(Model)*

1. *A model is a* description of the UoD *seen from some* perspective. *The perspective is defined by the model's goal: it defines the aspects of the UoD it should describe and the abstraction level.*
2. *A model can be abstract or physical. A mental model can be regarded abstract, while a description on paper is a physical model.*
3. *A model of some UoD can be used as* substitute *for that UoD, allowing to derive propositions valid in that UoD (perspective), represented as statements via the representation mechanism encapsulated in the model.*

Using a model as substitute for a UoD is the main reason for creating a model, as the model provides more insight in the UoD. It allows the creation of a 'shadow' UoD, which can be questioned and examined more efficiently than the 'real' UoD.

Example 1 Consider an organization that maintains a compact disc library (e.g., a hospital that has a music collection for their internal radio programs). To facilitate searching for CDs on keyword, the organization decides to construct a music database. For the purpose of searching CDs on keyword, the database is a model that acts as a substitute for the physical CD collection.

2.3.1 Information language

The goal of the system analyst is to create a complete and consistent formal model, based only on the communication between her and a domain expert. The system analyst is not assumed to have direct knowledge of the UoD, but her expertise is to make a well abstracted and complete formal model from the informal specification [5].

Let the information language be the set of possible statements the domain expert can give about the UoD, which are relevant for his perspective on the UoD. Now we can describe the modeling goal for the system analyst as follows:

> The creation of a formal model by the system analyst is equivalent to the finding of the information language. The information language has been found when the system analyst can produce all statements that the domain expert could have communicated.

To be able to talk about intermediate stages in the modeling process, we relax the necessary properties of a formal model. Modeling starts with an empty formal model. As statements are communicated, the intermediate formal model grows towards the final formal model.

The formal model is finished when it can produce the information language. Until then, some required statement may not be generated by the model. Alternatively, a generated statement may not be interpretable by the domain expert.

2.3.2 Information grammar

An extensional model is an explicit listing of all the statements in an information language. The informal specification may be seen as an incomplete extensional model.

An extensional model has some disadvantages:

- It takes a long (and maybe infinite) time to communicate all statements in the information language. In practice, listing is infeasible for non-trivial UoDs.

- When the UoD changes, or the conceptions of this UoD, changes in the information language must be communicated. It may be a difficult task to determine which old statements are to be replaced by new statements.

For these reasons, we limit formal models to be intensional models, containing the structure of the information language. This structure is called the information grammar. From the information grammar, all statements in the information language can be generated. The information grammar does not have the disadvantages of the extensional model: it is much more compact, and may even describe infinite information languages.

The task of the system analyst can now be described in further detail:

> The task of the system analyst is to create an intensional formal model from the informal specification. This involves finding structure in the informal specification, as well as obtaining or inducing additional information that is not part of the informal specification.

The need to obtain or induce additional information is a direct result of the assumption that the informal specification may be an incomplete extensional model. Although induction may be performed by the system analyst, we assume information about the structure of the information language can be obtained from the domain expert:

Assumption 2 A domain expert can provide statements about the structure of the information language he uses.

Note that this does not imply the domain expert can directly give the complete structure of the information language. Typically, the domain expert will give 'pieces of the puzzle' that are analyzed and combined by the system analyst.

We can now distinguish two types of statements communicated by the domain expert:

1. Example statements, elements of the information language.
2. Structural statements, which specify their structure.

Example 2 Consider the 'multiplication' domain. The following informal specification may be given by a domain expert:

```
s1: 5*3=15
s2: 8*11=88
s3: The grammar of these sentences reads:
    S := INT "*" INT "=" INT
    INT := [0-9]+
s4: the 3rd integer is the multiplication of the first two
s5: numbers range from [1..100]
```

The first two statements are example statements, while the others are structural statements. Both types of statements are accepted by the system analyst as input for creating a formal model. Note that the system analyst will have to obtain or induce more information than is available in the informal specification, in order to create a working calculator.

3. Vague concepts

The communication pattern between domain expert and analyst is the way in which the dialogue between them takes place. Typically, the domain expert provides details about the UoD, whereas the system analyst asks questions in order to trigger the domain expert to provide new or revised information.

The communicative behavior of the system analyst is determined by the task to construct a complete and consistent formal model [9]. The system analyst may exhibit the following two extreme types of behavior: awaiting and strict.

3.1 Awaiting behavior

An awaiting system analyst waits for the domain expert to produce an initial description of the UoD. The system analyst will interpret the description and create a formal model from it.

This behavior has several disadvantages:

- The domain expert typically is not able to provide a complete description of the UoD without feedback and elaboration.
- Questions by the system analyst can be posed only after the description is finished. When answering a question causes reconsidering the view on the UoD, the rest of the description may be useless.

In short, there is a lack of interaction and direct feedback.

3.2 Strict behavior

When displaying strict behavior, the system analyst wants to be able to interpret and understand a statement as complete as possible directly after it has been communicated. This implies:

- The domain expert has to explain the strict syntactical structure of the sentence.
- The domain expert must be able to explain how the concepts are related within the sentence, and how they are related to the current formal model maintained by the system analyst.
- The domain expert must not introduce inconsistencies, as this violates the consistency of the formal model.

A new sentence has to fit nicely in the current formal model, otherwise the system analyst will try to revise the formal model immediately, or has to refuse to incorporate the statement.

3.3 Towards allowing vague concepts

It is preferable in most cases to have a behavior that is somewhere in between the two given extremes. This is what people generally seem to do in practice: when the meaning of a sentence is not directly clear, and the sentence seems to be non crucial, we wait a little hoping that later statements will provide clues about how to interpret this unclear sentence. If this takes too long, or the misunderstanding becomes too crucial, we ask questions for further specification, hoping to get enough clues to proceed [7] (page 64).

We introduce vague concepts as a means for obtaining the behavior sketched above. Vague concepts are concepts or concept structures that are probably important for the final formal model, but for some reason do not fit into the current formal model. They have to be remembered, and opportunities have to be awaited or created which allow the concepts to become part of the final formal model.

Example 3 Suppose a domain expert gives the following statement:

```
s1: John lives in Madrid
```

Assuming the word structure is recognized, the statement is accepted as instance of a yet anonymous object type. The dialog continues:

```
r1: accept
s2: Mary lives in Madrid
```

Statement s_2 can not be generated from the existing information grammar. However, s_1 and s_2 may be generalized into

```
{John | Mary} lives in Madrid
```

suggesting the introduction of a special object type for John and Mary.

```
r2: accept
r3: a bicycle is a transportation means
```

Suppose our modeling technique required information structures to be completely connected, then there is no way to integrate this new statement. Displaying strict behavior, the system analyst has to reject the sentence, and the domain expert will have to offer the sentence s_3 at a later, more adequate moment.

```
r3: reject
```

If vague concepts would be allowed, the system analyst would not reject statement s_3. Instead, he would accept it, waiting for (or creating new) opportunities to incorporate the statement into the formal model.

4. Conclusion and further research

This paper discussed formal information modeling, based on analysis of the communication between domain expert and system analyst. The result of this information modeling is a formal model that can be said to be a model of the UoD with a certain level of completeness and consistency. We argued that in order to obtain desirable communication behavior of the system analyst, we need to deal with vague concepts.

In our future research we will try to further develop a theory concerning vague concepts in formal information modeling, as well as describe ways in which vague concepts can incrementally be made precise as part of the final formal model.

References

1. Booch, G., Rumbaugh, J., and Jacobson, I. (1999). The Unified Modelling Language Used Guide. Addison-Wesley, Reading, Massachusetts.
2. Chen, P. (1976). The entity-relationship model: Toward a unified view of data. ACM Transactions on Database Systems,1(1):9-36.
3. Falkenberg, E., Hesse, W., Lindgreen, P., Nilsson, B., Oei, J.,Rolland, C., Stamper, R., Van Assche, F., Verrijn-Stuart, A., and Vos,K., editors (1998).
4. A Framework of Information Systems Concepts. IFIP WG 8.1 Task Group FRISCO.
5. Frederiks, P. and van der Weide, T. (2003). Information modeling: the process and the required competencies of its participants. Technical report, Department of Information Systems, University of Nijmegen. submitted.
6. Hofstede, A. t., Lippe, E., and Weide, T. v. d. (1997). Applications of a Categorical Framework for Conceptual Data Modeling. Acta Informatica, 34(12):927-963.
7. Hoppenbrouwers, S. (2003). Freezing Language; Conceptualisation Processes across ICT-Supported Organisations. PhD thesis, University of Nijmegen.
8. Nijssen, G. and Halpin, T. (1989). Conceptual Schema and Relational Database Design: a fact oriented approach. Prentice-Hall, Sydney, Australia.
9. Veldhuijzen van Zanten, G., Hoppenbrouwers, S., and Proper, H. (2003). System development as a rational communicative process. In Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics, volume XVI, pages 126-130, Orlando, Florida.

Interorganizational Systems From Different Perspectives

M.K.M. Ibrahim
Department of Information Systems and Management
Tilburg University, Postbox 90153, 5000 LE Tilburg, The Netherlands
Tel. +31 13 466 8080
m.k.m.Ibrahim@uvt.nl

Abstract

The adoption and use of IOS's by organizations has proved to be difficult and complicated due to a number of reasons. Accordingly, IOS research has been distributed into multiple streams. This paper discusses four widely used theoretical perspectives and the IOS literature related to each perspective. The perspectives are transaction cost economics, incomplete contracts theory, adoption theory and resource dependence theory.

1. Introduction

Organizations are compelled to develop interorganizational relationships (IORs) to enable a range of activities such as supplying goods, research and development (R&D), and outsourcing. The use of information technology can facilitate a smooth flow of information from one organization to another using interorganizational systems (IOSs) - automated information systems shared by more than one organization and allowing information flow across organizational boundaries. IOSs can reduce the costs of communications and at the same time extend the possibilities of coordination. The academic literature that discusses IOSs is massive and applies many theoretical perspectives to view and analyze issues regarding the use of IOSs within interorganizational relationships. The objective of this paper is to provide a brief review of four widely used theoretical perspectives and the main IOS literature related to each perspective. The four perspectives are transaction cost economics, incomplete contracts theory, adoption theory and resource dependence theory.

The organization of the paper is as follows. Each of the sections two till five will discuss a theoretical perspective by first providing a concise review of the theory and consequently the IOS literature that uses that particular theory. Finally section six will provide a brief conclusion of the paper.

2. Transaction Cost Economics

The Theory
Transaction cost economics (TCE) concentrates on the make or buy decision. The theory argues that it is more efficient for an organization to buy a standard product externally from a special supplier who is an expert in producing that product than to produce the product internally. Nonetheless, buying products on the market can be less attractive when certain conditions apply such as for example when the organization needs a specific customized product. The organization is forced to internalize production under these conditions. TCE justifies why and predicts when an organization chooses to internalize the production process or conduct market exchange to acquire the product.

TCE identifies two types of costs that have to be considered to determine whether a transaction should take place externally on the market or internally within the firm: production costs and transaction costs. On the one hand, acquiring a product on the market is argued to lower production costs and to raise transaction costs. The production costs decline due to the economies of scale and specialization advantages the supplier benefit from. The transaction costs raises due the required negotiations and monitoring within the market. On the other hand, producing a product internally increases the production costs and lowers transaction costs. Hence, the organization will choose the most attractive alternative that minimizes the total costs.

The amount of transaction costs depends primarily on three factors [36]. First, the frequency of contracting; active first within a specific market usually have more knowledge regarding market conditions, traded

products and active suppliers within the market than markets that do not use the market as often. Second, the degree of uncertainty; uncertainty can arise from technological changes, unpredictable changes in consumer preferences or strategic behavior regarding nondisclosure and distortion of information. Third, the degree of asset specificity, which is the degree to which assets are specifically designed for a particular objective. TCE contends that transactions that are characterized by higher levels of asset specificity should be produced by organizations internally because such assets only can be redeployed at a great loss in values resulting in considerable quasi-rents.

IOS Literature Using The TCE Theory

IOS Research applying the TCE has tried to investigate the impact of IOS on the transaction structure. Malone [24] proposed the 'electronic markets hypothesis, which argues that information technology will reduce the information coordination costs and this will encourage the use of electronic markets. He contends that eventually electronic markets will obtain the preference above electronic hierarchies for coordinating economic activities. Clemons et al [7] disagreed and proposed the 'move to the middle hypothesis, where they argue that information technology in the form of IOS will reduce coordination costs, operation risks and opportunism risks. Because of these reductions, it will be more efficient to create long-term relationships with a smaller number of suppliers. Gurbaxani and Whang [14] focused on three types of costs: external and internal coordination costs and operating costs. They argue that information technology has shrunk external and internal coordination costs and improve the operational efficiency. Consequently, the use of both electronic markets and electronic hierarchies will be increased. Moreover, they content that the general impact of information technology will largely depend on the factors specific to the organization and the industry.

TCE has enabled scholars to justify the formation of many IORs and the use of IOSs within these relationships. The limited focus of TCE on short term cost minimization results in the ignorance to consider other important criteria such as social issues and learning within the relationship. These criteria can have a significant impact on the relationship.

3. Incomplete Contracts

The Theory

A complete contract is a contractual agreement between economic agents that specifies the responsibilities of each party in every possible situation or contingency. Williamson [36,37] and Maskin and Tirole [25] reason that contracts are almost never complete. As discussed earlier, Williamson [37] argues that the cost of contracting and subsequently enforcing these contracts depends on the chosen governance structure, i.e. market or hierarchical. Grossman and Hart [13] contend that each governance structure involve a different type of contractual rights: specific and residual rights. If it is too costly to stipulate all the specific rights in the contract, then all the rights will be transferred excluding few rights that are mentioned in the contracts. Therefore, ownership is important in the incomplete contract theory. Ownership dictates the destiny of an asset in contingencies not described in the contract, that is to say ownership is the purchase of the residual rights of control [3, 13]. Because contracts are almost never complete, owners have a relatively stronger position compared with non-owners because they mostly acquire the residual income streams due to their strong negotiating position.

There are several reasons that compel contracts to be incomplete. Hart and Moore [16] argue that some contractual terms are unverifiable because they are not commonly observable by all parties or more specifically the party that is responsible for enforcing the contract (e.g. the court). Second, Grossman and Hart [13] contend that is impossible to incorporate every potential contingency in the contract. Because the parties cannot identify ex ante all possible ex post contingencies, they are constrain to an incomplete contract they does not enclose all contingencies. Third, even though some contingencies can be predicted, discussing and writing them into a contract may not take place. Maskin and Tirole [25] point out that this can be due to the high transaction costs of describing the possible states of nature. As a result of these three reasons, there will be some possible contingency not included in the contract making the contract incomplete.

IOS Literature Using The Incomplete Contracts Theory

Incomplete contracts theory has considerable relevance to IOS theories. IOS contracts are inherently incomplete as all three earlier discussed reasons that cause contracts to be incomplete are present [15]. First, the use of an IOS necessitates asset-specific investments in IT assets such as hardware, software and also complementary investments in assets such as expertise and training. These investments can be hard to

observe by the other participants involved and perhaps impossible to verify by a third party. Second, the various applications of IOS for different activities make it difficult to foresee all the contingencies. Due to the high speed of environmental change, organizations need to react frequently and change their strategies to adapt to the environmental change. This can be even exacerbated when the value of the IOS increases with the number of organizations employing the IOS. This is the case for electronic market places. Finally, some future contingencies can be foreseen and nevertheless, organizations may choose not to put them in the contract. This is observed when organizations create partnerships. Organizations choose to enter long-term relationships without specifying all contingencies and instead relying on the interorganizational trust present within the relationship.

The failure to attain complete contracts underlines the importance of IOS ownership as portrayed by the case of the Airline Computer Reservation Systems (CRS's). The CRS's were traditionally owned by the individual airlines and American Airways and United Airlines were leading and affecting the market. Smaller airlines contended that American and United should divest their CRS's to create independent intermediaries. This ownership structure would serve competition better and encourage higher levels of investment, and eventually higher economic surplus.

The incomplete contract theory was used by Bakos and Brynjolfsson [2] to determine the optimal strategy for buying organizations that use IOS. They argue that buying organization can maximize their profits by reducing their bargaining power through limiting commitments to a small number of suppliers. Even though this is apparently inconsistent, the reduction in the number of suppliers is required to persuade suppliers to conduct noncontractible investments. This is explained by the rationale that when a suppliers perceive a particular buyer to be dependent and willing to enter a long term relationship, then the supplier will be more willing to conduct asset specific investments. Another IOS related application of incomplete contracts theory is regarding the ownership structures in electronic networks. Bakos and Nault [3] argue that if there are one or more essential assets for the functioning of the IOS, then all the assets of the IOS should be owned together. Hence, common ownership by all participants is optimal when an IOS requires essential assets, such as a common IT infrastructure. Furthermore, they argue that when essential assets and indispensable participants are absent, sole ownership will not be the optimal ownership structure. Therefore, if IOS partners want to prevent any single party from controlling the IOS, then they should make certain that the IOS doesn't need any essential assets and if there are such assets, then they should be owned by everyone.

Banker, Kalvenes and Patterson [4] argue, contrary to the mainstream, that IT increases contract completeness. They contend that the progress in communication technologies will reduce monitoring costs. It will be possible to increase monitoring and some parts of the contract will be converted from non-contractible to contractible. The buyer may then choose to enter more terms in the contract to decrease his risk and make the contract more complete. Due to these additional contractual terms, the cost of monitoring for that particular supplier will increase. Banker, Kalvenes and Patterson [4] contend that the decrease of transaction costs generated by IT will be cancelled and surpassed by the increase in contractual terms and monitoring costs per supplier, leading to a reduction in the optimal number of suppliers. This shows that the claims of Bakos and Brynjolfsson [2] and Clemons et al [7] hold under the more general conditions of Banker, Kalvenes and Patterson [4].

4. Adoption Theory

The Theory

Adoption generally refers to the decision of any individual or organization to make use of an innovation [12]. IOS adoption research has been influenced by the broad organizational adoption approach [32] significantly [6, 27]. This approach emphasizes that adoption can be based on the perceived characteristics of the innovation. Rogers [32] identified five characteristics that can either facilitate or impede the adoption of an innovation. First, relative advantage is the extent to which the innovation is perceived better than that it is replacing. Second, compatibility is the perceived consistency of the values, needs, and experiences of potential adopters with the innovation. Third, complexity is the extent to which an innovation is difficult to understand. Fourth, triability refers to the extent to which an innovation can be experimented on before a full commitment must be made. Finally, observability is the degree to which the benefits of the proposed innovation are visible. These characteristics are primarily based on individual-level adoption decisions.

Framback and Schillewaert [12] argued that features of the adopting organization can affect the adoption process and they pointed out to three main organizational features. First, the size of the organization is argued

to be positively or negatively related to innovation adoption. On the one hand, larger organizations are under higher pressure to adopt innovations to support and improve their performance [32]. On the other hand, smaller organizations are more flexible and have enhanced receptiveness towards new innovations. This apparent inconsistency can be accredited to the relationship of organization size with other organizational features, such as structure, strategy and culture. Organizational structure is the second feature argued to influence organizational adoption. Organizational structure is shaped by multiple attributes, which can have diverse impacts on adoption. High levels of centralization and formalization have a tendency to encourage the implementation of adoption decision, while low interconnectedness have a tendency to inhibit the information flow and consequently the implementation of the adoption. Finally, the degree of organizational innovativeness influences the adoption propensity. For example, Hurley and Hult [20] point out that organizational cultures that call attention to learning, development and participative decision-making produce higher levels of innovation.

IOS Literature Using The Adoption Theory

The IOS literature has identified three main groups of factors that influence the adoption of IOS: nature of the technology adopted, the adopting organization, and the interorganizational relationships or more generally the external environment [23].

The nature of the adopted technology may create difficulties for the adopting organizations. Important factors regarding the technology that effect IOS adoption comprise network security, system integration, data conversion and the compatibility of software and hardware [22]. The security is a key issue as IT do not always fulfill the transaction safety requirements of organizations [31]. Moreover, the adoption of an IOS may generate complex and expensive integration issues. The integration of the IOS with the internal IS can involve rigorous technical efforts involving activities such as the conversion of program codes, databases and the validation of data formats [35].

The second group of factors that influence IOS adoption consists of organizational factors. Organizations participate in IOSs or adopt new innovations in general only when they offer better benefits compared with the previous situation [32]. IOS benefits can range from modest gains such as reduced communication costs and improved customer service [14, 30] to transformative advantages that enhance competitive advantage [26], enable business process reengineering and support industry value chain integration [6]. Besides the benefits, the compatibility of the IOS with existing organizational policies, procedures, values, and systems and top management support are mostly perceived as relevant aspects of IOS innovation and adoption [6, 21, 27].

The third group of factors consists of the stimulators and barriers that other organizations set on the focal organization to enhance or inhibit the adoption of IOS. Competitive pressure and exercised power have been found to influence EDI adoption [30]. Hart and Saunders found both power and trust are important issue for adoption and use. Powerful organizations can manipulate their partners in two ways. The powerful organization can induce its partners to adopt the new technology by providing rewards and benefits or it can force them to adopt it with the threat of abandoning the partner if it rejects. Trust is also identified as an important factor as its presence can provide monitoring and transaction cost reductions and its abuse will initiate a vicious cycle and impede constructive cooperations [18].

5. Resource Dependence Theory

The Theory

The roots of resource dependence theory (RDT) can be found in an article by Emerson in 1962 where he illustrated the analogy between power and dependence across all forms of social relationships [11]. Emerson argued that the dependence of a party B on party A is a function of availability and motivational investment and is directly comparative to the power of A over B. In economic expressions, this is known as supply and demand. The theory of Emerson was later applied by Pfeffer and Salancik [28] to analyze the relationship between the organization and its external environment. They distinguished between general structural characteristics that describe the entire environment and particular relationships among identifiable social actors. The three most elemental structural characteristics of the environment are concentration, munificence, and interconnectedness. Concentration is the level of diffusion of power and authority within the environment, munificence is the level of availability or shortage of critical resources, and interconnectedness is the number and configuration of connections between organizations. These three structural characteristics shape the general relationships between social actors. On an individual level, the degree of dependence that

an individual organization faces is determined by the importance of the externally controlled resources to the success of the focal organization, the degree of discretion that the external environment has over the resource allocation of that resource and finally the number of alternatives to that particular resource.

The RDT acknowledges that a single organization can not produce or own all the required resources for its operations. The organization is forced to acquire these resources from several other actors and organizations in its environment. Therefore, a successful organization is an organization that is able to satisfy the demands of the various stakeholders such as employees, customers and shareholders. To realize this, the organization can choose between three alternative types of action to deal with the demands of the external environment: it can avoid them, comply with them or try to modify them to acquire a better set of demands, which can be fulfilled easier. The third alternative is the main focus of RDT. The theory contends that organizations conduct actions to reduce their dependence on other organizations and the risk that is emanating from these dependencies. The dependencies can be modified using two strategies. The first strategy is the ownership-alteration strategy, which implies that the needed external resource should be purchased. This can result in vertical integration, horizontal integration and diversification. The second strategy entails creating a quasi-hierarchical relationship to govern the uncertainty within the relationship. Examples of quasi-hierarchical relations are joint ventures, interlocking boards of directors, associations and cartels. The purpose of both strategies is to create stability by achieving better planning and more accurate forecasting. Basically, the organization will try to reduce its dependence on the environment by constantly balancing two contradictory forces: certainty and autonomy [9].

IOS Literature Using The Resource Dependence Theory

The interdependence between organizations is the focus of IOS literature that uses the resource dependence theory. Various authors found that more effective use of IOS can be related to the level of integration between the interorganizational IT infrastructure and the internal IT infrastructure of each organization [5, 8, 17]. This high integration results in higher interdependence between the organizations [10, 17]. Therefore, intensive use of interorganizational systems results in a shift in the relationship between organizations to a reciprocal interdependence, that is the outputs of each organization become inputs for one or more of the other organizations. Thompson argued that reciprocal interdependence has to be kept low in the organization structure [33]. Consequently a potential disadvantage of interorganizational systems is that they can make entire organizations reciprocally interdependent on each other. The impact of this interdependence under future unexpected results is unknown; this can decrease the flexibility of organizations and produce new uncertainties. Furthermore, it illustrates the argument of Pfeffer and Salancik [28] that organizations react to the uncertainty problems by intensifying their interconnectedness by coordinating their behaviors in ways predictable to each other. This will produce higher interorganizational interdependence and new uncertainties that where not present in the initial situation.

Furthermore, the use of IOS is argued to influence the power and control structures within interorganizational relationships [1, 17]. The propositions on how IOSs influence the power and control are divided along two directions. Some literature, mostly earlier published, argued that the use of IOS's is exclusive to selected organizations that fulfilled the demands and rigorous criteria to join. They mostly referred to EDI systems that needed high set up costs. The technology restricted the IOS to organizations that possessed the required resources. Recent literature contends that the use of modern IOS leads to more just relationsips between organizations [1, 34]. Angeles [1] argued that I-EDI modifies the power structure by transfer the power from large hub organizations to smaller and mid-sized organizations. The large organizations used previously their central position to dictate the terms of relationships and they exploited this by utilizing power to their favor. The progress of IT has and the emergence of standards, such as XML and ebXML, has enabled small and mid-sized organizations to have a broader choice of trading partners.

6. Conclusion

IOSs are used in various ways to facilitate interorganizational relationships. This paper has provided a concise review of four theoretical perspectives that are used within the IOS literature. TCE has received significant attention within IOS literature as it focuses on how organizations should organize their boundary-spanning activities so as to minimize the sum of its production and transaction costs. Information technology has major affects on interorganizational communications and coordination and consequently TCE has been used to study the impact of IOS on production and transaction costs. The second perspective discussed was incomplete contracts theory. This perspective is relevant to the study of IOSs as IOS contracts are inherently incomplete in the three perspectives; the IOS requires asset specific investments that are hard

to observe by other parties involved, it is difficult to foresee all contingencies related to IOSs as they can be involved in many complex activities and even some contingencies that are foreseen are not included in the contract. The third perspective discussed is adoption theory. Theories using this perspective have illustrated that the adoption and use of an IOS is dependent on three main groups of factors; the nature of IOS technology being adopted as some technologies can create difficulties that inhibit successful adoption, the adopting organization as it is mainly the organization that need to initiate and execute the adoption and the relationship with other organizations as the use of the IOS's can have a major impact on the IORs. Finally, the resource dependence theory was discussed and how it is used to analyze the impact of IOSs on the interdependence within IORs. IORs are found to influence the power structure within IORs as they can eliminate the power of big organizations that operated as hubs and forced small organizations to follow their regulations.

References

1. Angeles, R. (2000), "Revisiting the role of Internet-EDI in the current electronic scene," Logistics Information Management, 13 (1), 45-57.
2. Bakos, J.Y. and Erik Brynjolfsson (1993), "Infroamtion Technology, Incentives and the Optimal Number of Suppliers," Journal of Management Information Systems, 10 (2), 37.
3. Bakos, Y. and B. Nault (1997), "Ownership and Investment in Electronic Networks," Information Systems Research, 8 (4), 321-41.
4. Banker, R., J. Kalvenes, and R. Patterson (2000), "Information Technology, Contract Completeness and Buyer-Supplier RElationships," in The 21st Annual International Conference on Information Systems. Brisbane, Australia.
5. Chandrashekar, A. and P. Schary (1999), "Toward the Virtual Supply Chain: The Convergence of IT and Organization," The International Journal of Logistics Management, 10 (2), 27-39.
6. Chwelos, P., I. Benbasat, and A.S. Dexter (2001), "Research Report: Empirical Test of an EDI Adoption Model," Information Systems Research, 12, 304-21.
7. Clemons, Erik K., Sashidhar P. Reddi, and Micheal C. Row (1993), "The Impact of Information Technology on the Organization of Economic Activity: The "Move to the Middle" Hypothesis," Journal of Management Information Systems, 10 (2), 9-35.
8. D'Amours, S., B. Montreuil, P. Lefrancois, and F. Soumis (1999), "Networked manufacturing: The impact of information sharing," international journal of production economics, 58 (1), 63-79.
9. Davis, Gerald and Walter Powell (1992), "Organization-environment relations," in Handbook of Industrial and Organizational Psychology, M. Dunnette and L. Hough, Eds. Vol. 3. Palo Alto CA: Consulting Psychologists Press.
10. Ekering, Chad F. (2000), De Specificiteit van EDI: Dutch University Press.
11. Emerson, R.M. (1962), "Power-dependence relations," American Sociological Review, 27, 31-41.
12. Frambach, R.T. and N. Schillewaert (2002), "Organizational innovation adoption: a multi-level framwork of determinants and opportunities for future research," Journal of Business Research, 55 (2), 163-76.
13. Grossman, S.J. and O.D. Hart (1986), "The costs and benefits of ownership: A theory of vertical and lateral integaration," Journal of Political Economy, 94 (4), 691-719.
14. Gurbaxani, V. and S. Whang (1991), "The Impact Of Information Systems On Organizations and Markets," Communications of the ACM, 34 (1), 59-73.
15. Han, K., R.J. Kauffman, and B.R. Nault (2003), "Who Should Own 'IT'? Ownership and Incomplete Contracts in Interorganizational Systems," Working Paper.
16. Hart, O. and J. Moore (1988), "Incomplete Contracts and Renegotiation," Econometrica, 56, 755-85.
17. Hart, P. and D. Estrin (1991), "Inter-Organization Networks, Computer Integration, and Shifts in Interdependence: The Case of the Semiconductor Industry," ACM Transaction on Information Systems, 9 (4), 370-98.
18. Hart, P. and C. Saunders (1997), "Power and Trust: Critical Factors in the Adoption and Use of Electronic Data Interchange," Organization Science, 8 (1), 23-42.
19. Holland, C.P. and A.G. Lockett (1997), "Mixed Mode Network Structures: The Strategic Use of Electronic Communication by Organizations," Organization Science, 8 (5), 475-88.
20. Hurley, R.F. and G.T.M. Hult (1998), "Innovation, Market Orientationm and Organizational Learning: An Integration and Empirical Examination," Journal of Marketing, 62 (3), 42-54.
21. Iacovou, C.L., I. Benbasat, and A.S. Dexter (1995), "Electronic Data Interchange and Small Organizations: Adoption and Impact of Technology," MIS Quarterly, 19 (4), 465-85.

22. Jones, M.C. and R.C. Beatty (1998), "Towards the development of measures of perceived benefits and compatibility of EDI: a comparatice assessment of competing first order factor models," European Journal of information systems, 7 (3), 210-20.
23. Kurnia, S. and R.B. Johnston (2000), "The need for a processual view of inter-organizational systems adoption," The Journal of strategic information systems, 9 (4), 295-319.
24. Malone, T.W., J. Yates, and R.I. Benjamin (1987), "Electronic Markets and Electronic Hierarchies: Effects of Information Technologies on Market Structure and Corporate Strategies," Communications of the ACM, 30 (6), 484-97.
25. Maskin, Eric and Jean Tirole (1999), "Unforeseen Contingencies and Incomplete Contracts," Review of Economic Studies, 66 (1), 83-114.
26. Mukhopadhay, T., S. Kekre, and S. Kalathur (1995), "Business Value of Information Technology: A Study of Electronic Data Interchange," MIS Quarterly, 19 (2), 137-56.
27. O'Callaghan, R., P.J. Kauffman, and B. Konsynski (1992), "Adoption Correlates and Share Effects of Electronic Data Enterchange Systens in Marketing Channels," Journal of Marketing, 56 (2), 45-56.
28. Pfeffer, Jeffrey and Gerald R. Salancik (1978), The External Control of organizations: A resource dependence perspective. New York: Harper & Row.
29. Premkumar, G. and K. Ramamurthy (1995), "The Role of Interorganizational and Organizational Factors on the Decision Mode for Adoption of Interorganizational Systems," Decision Sciences, 26 (3), 303-36.
30. Premkumar, G., K. Ramamurthy, and S. Nilakanta (1994), "Implementation of electronic data interchange: an innovation diffusion perpective," Journal of Management Information Systems, 11 (2), 157-86.
31. Ratnasingam, Pauline Puvanasvari (2001), Interorganizational Trust in Business to Business E-Commerce. Rotterdam: Erasmus Research Institute of Management (ERIM).
32. Rogers, Everett M. (1995), Diffusion of Innovations (Fourth ed.). New York: The Free Press.
33. Thompson, James D. (1967), Organizations in action. New York: McGraw-Hill.
34. Threlkel, M.S. and B. Kavan (1999), "From traditional EDI to Internet-based EDI: managerial considerations," Journal of Information Management, 14, 347-60.
35. Truman, G.E. (2000), "Integration in Electronic Exchange Environments," Journal of Management Information Systems, 17 (1), 209-45.
36. Williamson, O.E. (1985), The Economic Institutions of Capitalism. New York: Free Press.
37. ---- (1975), Markets and Hierarchies. New York: Free Press.

Het selecteren van een geschikte methode voor het formuleren van indicatoren

What you measure is what you get

Bernd Wondergem
LogicaCMG
Meander 901, Postbus 7015, Arnhem
bernd.wondergem@logicacmg.com

Abstract

Performance management (PM) is een vorm van 'management by fact': de doelen van de organisatie worden eerst expliciet gemaakt en vervolgens wordt met feitelijke informatie in kaart gebracht in hoeverre deze doelen bereikt worden. Hierbij worden de doelen meetbaar gemaakt door ze te vertalen in zogenaamde indicatoren. De indicatoren bepalen waarop gemeten zal gaan worden. Het is dus zaak de juiste indicatoren te formuleren.

Voor het formuleren van indicatoren bestaan verschillende methoden. Aan de hand van de voor- en nadelen kan een organisatie de best passende methode kiezen. In dit artikel worden verschillende methoden benoemd. Daarnaast worden verschillende criteria besproken voor de selectie van een geschikte methode. Ook laten we in dit artikel zien hoe de technieken van performance management toegepast kunnen worden in de informatiewetenschap.

1. Inleiding

Performance management (PM) is een vorm van 'management by fact': de doelen van de organisatie worden eerst expliciet gemaakt en vervolgens wordt met feitelijke informatie in kaart gebracht in hoeverre deze doelen bereikt worden. Hierbij worden de doelen meetbaar gemaakt door ze te vertalen in zogenaamde stuurvariabelen of indicatoren (zie bijvoorbeeld [15] en [12]). Als instrument worden bij PM vaak scorecards gebruikt. Deze geven per organisatieonderdeel een overzicht van de indicatoren waarmee gemeten wordt. De indicatoren geven kwantitatief weer in hoeverre de organisatie haar doelen realiseert.

Een belangrijke stap in PM is het formuleren van indicatoren. Dat bepaalt immers waarop gemeten zal gaan worden. Hierop is het adagium 'what you measure is what you get' van toepassing. Mensen zijn geneigd aandacht en energie te schenken aan die zaken waarop ze beoordeeld worden. De onderwerpen van prestatiemeting worden daarmee als vanzelf aandachtspunten voor medewerkers. Dit maakt ook duidelijk dat het belangrijk is om de goede zaken te meten. Ofwel, om de juiste indicatoren te formuleren.

In de literatuur en de praktijk wordt een groot aantal methoden voor het formuleren van indicatoren beschreven en gebruikt (zie ook [14]). Ook worden verschillende criteria genoemd waarmee een geschikte methode gekozen kan worden. Welke methode geschikt is hangt namelijk af van verschillende voor de organisatie specifieke zaken.

In dit artikel geven we een overzicht van criteria en methoden. Ook gebruiken we de criteria om de methoden te classificeren. We willen hiermee het inzicht in de methoden vergroten en organisaties een handvat bieden om de voor hen meest geschikte methode te kiezen. Uiteindelijk kan dit bijdragen aan effectiever performance management. Ook laten we in dit artikel zien hoe de technieken van performance management toegepast kunnen worden in de informatiewetenschap.

Dit artikel is als volgt opgebouwd. In sectie 2 wordt een aantal methoden voor het formuleren van indicatoren summier beschreven. Sectie 3 behandelt de criteria waarmee een geschikte methode geselecteerd kan worden. In sectie 4 worden de methoden op basis van deze criteria geclassificeerd. Sectie 5 behandelt het selecteren

van een geschikte methode. In sectie 6 wordt beschreven hoe de methoden van performance management toegepast kunnen worden in de informatiewetenschap. In sectie 7 worden conclusies getrokken en aanbevelingen gedaan.

2. Methoden voor het formuleren van indicatoren

In deze sectie worden verschillende methoden voor het formuleren van indicatoren bondig beschreven.

Een bekende en veel gebruikte manier om indicatoren te formuleren is de Balanced Scorecard (BSC) [5, 6]. In de BSC methode worden de indicatoren verdeeld over vier perspectieven: financieel, klant, proces en innovatie. In de perspectieven wordt aangegeven welke onderwerpen voor de organisatie van prominent belang zijn. Deze zogenaamde kritieke succesfactoren worden vervolgens vertaald in (geoperationaliseerd met) indicatoren.

Het INK-managementmodel [3] wordt gepositioneerd als kwaliteitsmodel met negen velden (perspectieven). Binnen deze velden kunnen, net als bij de BSC methode, indicatoren geformuleerd worden. Het INK model bevat daarnaast ook kwalitatieve suggesties over de inrichting van organisaties en kent een groeimodel met vijf fasen. Deze aspecten zijn voor het formuleren van indicatoren echter minder van belang.

De methode van de Customer Safisfaction Cockpit (CSC) [13] bevat indicatoren voor het besturen van klanttevredenheid. De CSC maakt een systematische uitsplitsing van dit onderwerp en identificeert de factoren die klanttevredenheid bepalen. Als uitbreiding op de indicatoren is in de CSC een integraal besturingsmodel voor contactcenters opgenomen. Het model met deze uitbreiding zullen we met CSC aanduiden; de term CSC- gebruiken we voor het model van indicatoren voor klanttevredenheid zonder het besturingsmodel.

Activity-based costing (ABC) [4] is een manier om de kostprijs van producten en diensten te berekenen. Daartoe wordt in een vastomlijnd model beschreven welke mensen en middelen ‘verbruikt’ worden per activiteit. De kosten per product worden bepaald door de mate waarin activiteiten mensen en middelen verbruiken. Deze verbanden worden nauwkeurig in kaart gebracht, zodat er wiskundige berekeningen mee uitgevoerd kunnen worden.

Six Sigma [2] heeft tot doel om productiefouten te minimaliseren. Dit wordt gedaan door de variëteit in productieprocessen te verminderen. Via statistische analyses wordt de ‘business performance’ in kaart gebracht. Er bestaan varianten voor het verbeteren van bestaande processen en voor het inrichten van nieuwe processen.

Quality Function Deployment (QFD) [Mazur, 1993] is een methode om nieuwe producten en diensten te ontwikkelen, waarbij de klantwaarde maximaal is. Vanuit gebruikerswensen wordt outside-in een vertaling gemaakt naar producteisen en procesinrichting.

Value-based management (VBM) (zie [11]) richt de organisatie op het maximaliseren van aandeelhouderswaarde. Dit is een sterk economisch getinte methode, waarbij boekhoudkundige indicatoren de basis vormen.

Het procesmodel [7] start met een beschrijving van de processen van een organisatie. Voor de producten uit de processen worden vervolgens indicatoren geformuleerd. Het systeemmodel heeft een vergelijkbare werkwijze, maar onderkend ook verbanden tussen processen.

3. Criteria voor het selecteren van de juiste methode

In de vorige sectie zijn verschillende methoden voor het formuleren van indicatoren beschreven. Welke methode voor een organisatie (het meest) geschikt is, hangt van verschillende zaken af. In deze sectie beschrijven we een aantal criteria, waarmee organisaties een keus kunnen maken voor een methode. Het maken van de keus wordt beschreven in de volgende sectie.

In [7] worden twee criteria benoemd: de ‘stijl van leidinggeven’ en de ‘heersende problematiek’. Bij stijl van leidinggeven wordt onderscheid gemaakt tussen stimuleren en beheersen. Bij stimuleren past een bottom-up aanpak waarbij decentraal door de medewerkers zelf doelen en indicatoren geformuleerd worden. Het centrale management heeft daarbij een ondersteunende taak. Bij beheersen als dominante leiderschapsstijl

horen de tegenovergestelde termen als top-down, centraal en sturend.

Bij de heersende problematiek wordt onderscheid gemaakt tussen consolideren versus innoveren. Consolideren heeft tot doel de organisatie te stabiliseren: het in stand houden van bestaande activiteiten. Innoveren mikt op verandering in de organisatie. Bij innoveren gaat de organisatie bijvoorbeeld op zoek naar nieuwe producten of werkwijzen.

In de praktijk merken we dat de twee genoemde criteria niet altijd bruikbaar zijn. Soms spelen er in organisaties namelijk andere thema's die belangrijker zijn. Ook komt het voor dat managers en medewerkers onvoldoende beeld hebben bij de genoemde criteria; ze kunnen dan niet goed inschatten welke keuze gemaakt moet worden. Twee andere criteria kunnen dan uitkomst bieden.

Met het onderwerp van sturing geeft de organisatie aan welk onderwerp er met indicatoren bestuurd moet worden. Hierbij kan onderscheid gemaakt worden in proces versus strategie. Bij proces staat de procesgang centraal, inclusief de daarin gemaakte producten, de actoren in het proces en de gebruikte stuurinformatie en kwaliteitseisen. Processturing is de laatste jaren hoger op de agenda komen te staan door de opkomst van het INK-managementmodel. De procesgerichte organisatie is daarin één van de vijf fasen uit het groeimodel. Bij strategie gaat het om de succesfactoren en doelen die de organisatie wil realiseren.

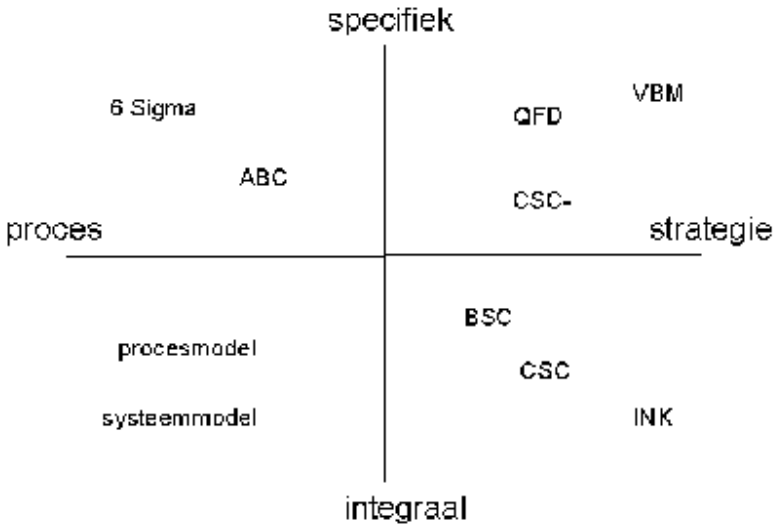
De mate van focus geeft aan of een organisatie specifieke onderwerpen wil besturen of dat de organisatie integraal bestuurd moet worden. Bij sommige methoden staan bijvoorbeeld specifieke financiële maatstaven centraal. Andere richten zich specifiek op klanttevredenheid. Integrale sturing, waarbij aan alle aspecten van de bedrijfsvoering aandacht geschonken wordt, is de laatste jaren in opkomst geraakt.

Naast de reeds genoemde criteria, worden in de literatuur ook andere criteria gevonden. Zo wordt in [1] gesproken over verticale en horizontale benaderingen. Verticale benaderingen worden centraal bestuurd, mikken op controle en standaardisatie en richten actie op het verbeteren van de slechts presterenden. Horizontale benaderingen benadrukken zaken die decentraal van belang zijn, hebben oog voor kwalitatieve resultaten, werken met informele systemen en streven naar continue verbetering ongeacht het startpunt. Het onderscheid in horizontale en verticale benaderingen heeft overlap met de tweedeling in stijl van leiderschap. Beide indelingen hebben als nadeel dat ze in de praktijk slechts beperkt onderscheidend zijn. De methoden zelf kunnen namelijk op verschillende manieren toegepast worden: in verschillende werkvormen met meer of minder decentrale of lokale inbreng. De indelingen hebben daarmee niet zozeer betrekking op de methoden zelf, maar meer op de manier waarop deze toegepast worden.

Een laatste criterium is de sturingsvorm, waarbij onderscheid gemaakt kan worden in actiematige sturing of sturing op resultaat [9]. Actiematige sturing beschrijft hoe de stappen in het voortbrengingsproces uitgevoerd moeten worden, het schrijft werkwijzen voor. Resultaatgerichte sturing legt het beoogde resultaat of effect vast en stelt niet vast hoe dat resultaat bereikt moet worden. Bij beide vormen kunnen passende indicatoren geformuleerd worden. Deze indeling lijkt op die van het onderwerp van sturing, met proces en strategie als tweedeling. Echter, ook als het proces het onderwerp van sturing is, kan nadruk gelegd worden op de resultaten (producten) van het proces.

4. Indelingen van de methoden

De methoden voor het formuleren van indicatoren kunnen ingedeeld worden met de in de vorige sectie beschreven criteria. Allereerst delen we de methoden in volgens de criteria leiderschapsstijl en heersende problematiek (zie Figuur 1). Deze indeling verfijnt het schema zoals beschreven in [7] met concrete methoden. Leiderschapsstijl vormt de verticale as, de heersende problematiek de horizontale. Deze indeling genereert vier kwadranten, die hieronder afzonderlijk besproken worden.



Figuur 1. Indeling naar leiderschapsstijl en heersende problematiek

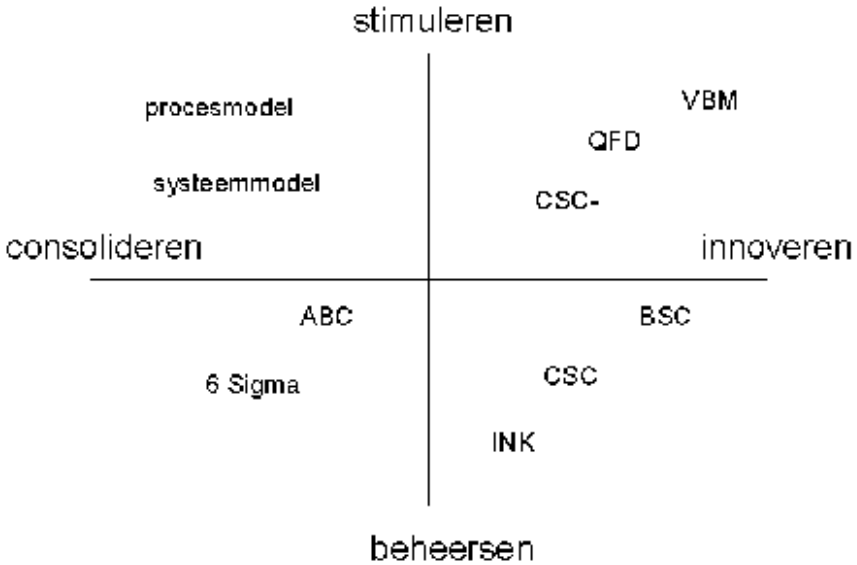
In het kwadrant linksboven in Figuur 1 staan de methoden die goed passen bij een stimulerende leiderschapsstijl en tot doel hebben de organisatie te consolideren. Proces- en systeemmodellen bieden, binnen de grenzen van de bestaande processen, ruimte aan medewerkers om zelf indicatoren te formuleren.

Het kwadrant rechtsboven bevat methoden die ook passen bij een stimulerende leiderschapsstijl, maar waarbij innovatie het doel is. De methoden schrijven niet voor hóe de organisatie ingericht moet worden, alleen wat het resultaat moet zijn. Er wordt daarom veel vrijheid geboden aan medewerkers, zowel bij het formuleren van indicatoren als bij het inrichten van de organisatie.

In het kwadrant rechtsonder wordt gemikt op beheerste innovatie. Er worden daarom duidelijke kaders geboden waarbinnen innovatie plaats moet vinden. Dit gebeurt door het aanbieden van een besturingsmodel. Daardoor worden spelregels en een referentiekader vastgesteld voor de besturing.

Het kwadrant linksonder is gericht op beheerste consolidatie. Er wordt strikt vastgelegd hoe er gewerkt wordt en aan die werkwijze wordt een vaste manier methode gekoppeld voor het formuleren van indicatoren. Dat betekent dat de uitgangspunten van de methode duidelijk beschreven zijn. Bij ABC, bijvoorbeeld, staat het model voor het formuleren van indicatoren op hoofdlijnen vast. Bij Six Sigma staat vast welk doel de methode dient en waar de aandacht op gericht zal zijn: productiefouten minimaliseren.

Een andere indeling is die naar de criteria mate van focus en onderwerp van sturing. Het onderscheid tussen specifieke en integrale sturing is verticaal uitgezet (zie Figuur 2). Het onderwerp van sturing, ofwel de tweedeling tussen proces en strategie, is horizontaal uitgezet.



Figuur 2. Indeling naar mate van focus en onderwerp van sturing

De methoden die een specifiek doel hebben, vinden we terug in de bovenste helft van Figuur 2. Aan de linkerzijde zien we methoden die het proces centraal stellen. Dat zijn Six Sigma, die mikt op proceskwaliteit, en ABC, waarbij vanuit de processen kostprijzen berekend worden. Aan de rechterzijde staan de methoden die uitgaan van strategische doelen. Daar vinden we de CSC- en QFD, die klanttevredenheid centraal stellen, en VBM, waarbij de aandeelhouderswaarde gemaximaliseerd moet worden. Al deze methoden missen een integrale benadering.

De integrale methoden staan in de onderste helft van de figuur. Het proces- en systeemmodel redeneren daarbij vanuit de bestaande processen. Zij bezien echter de stappen van de werkwijzen in onderlinge samenhang. Bij het systeemmodel worden ook afhankelijkheden tussen processen meegenomen. Aan de rechterkant staan integrale benaderingen die vertrekken vanuit de strategie van de organisatie. Zij bieden alle een integraal besturingsmodel. Dat van de BSC is het simpelst. De CSC en het INK-managementmodel gaan een aantal stappen verder en beschrijven bijvoorbeeld ook kwalitatieve aspecten van het inrichten van de organisatie.

5. Selecteren van een geschikte methode

Het selecteren van een geschikte methode kan nu in een aantal stappen uitgevoerd worden. Eerst wordt bekeken welke doelen de organisatie nastreeft met het invoeren van performance management. Daaruit worden de belangrijkste criteria gedestilleerd. Aan de hand van de criteria wordt bekeken welke methoden in aanmerking komen. Hieruit wordt een keuze gemaakt.

Dit selectieproces is uiteraard wat te kort door de bocht. Vaak voldoet namelijk geen enkel ‘standaardmodel’ precies aan de eisen van de organisatie. Dan kan door combinatie uit een aantal modellen een nieuw model gemaakt worden. Zulke hybride vormen bieden vaak uitkomst. Zo wordt bijvoorbeeld vaak in het procesmodel toch enige vorm van strategische sturing gebracht.

Het combineren van modellen is echter niet altijd zinvol. Door twee modellen te combineren kan de kracht van de individuele modellen verloren gaan: de som der delen is dan minder dan het geheel. Het is dus raadzaam om eventuele combinaties zorgvuldig uit te voeren.

6. Toepassing van performance management in informatiewetenschap

Performance management kan gebruikt worden voor de besturing van organisaties in de informatiewetenschap, zoals universiteiten, bibliotheken en uitgeverijen. Dit geldt zowel voor integrale besturing van deze organisaties als voor besturing van deelgebieden. Deze onderwerpen krijgen de laatste jaren nadrukkelijk aandacht. De Universiteit Utrecht bijvoorbeeld, benoemt in haar ‘Ontwikkelingsplan 2001-2005’ expliciet het meetbaar maken van doelstellingen.

In de rest van dit hoofdstuk wordt aangegeven hoe het publiceren van artikelen (deels) bestuurd kan worden met indicatoren. Dat wordt gedaan vanuit het perspectief van een onderzoeksgroep op een universiteit. De verschillende kwadranten van Figuur 1 corresponderen met verschillende besturingsstrategieën. Deze hebben ook verschillende uitkomsten tot gevolg. Het is dus ook voor het publiceren van artikelen van belang goed na te denken over de besturingsvorm.

De verschillen tussen de kwadranten worden hieronder met voorbeelden uitgewerkt. In het kwadrant linksonder, horend bij een beheersende stijl voor consolidatie, worden bestaande kaders nadrukkelijk bevestigd. De hoogleraar van een onderzoeksgroep stelt daarbij zelf vast hoe hij publicaties zal waarderen. Hij gaat daarbij uit van de huidige situatie, waarin de ranking van tijdschriften bevestigd wordt. Als gevolg daarvan krijgen de gevestigde tijdschriften de beste publicaties en maken nieuwe tijdschriften geen kans. Daarnaast is de inbreng van de leden van de onderzoeksgroep gering; de hoogleraar regeert.

Het kwadrant linksboven gebruikt een procesbeschrijving als basis voor sturing van het publicatieproces. Als procesmodel kan bijvoorbeeld een waardeketen voor wetenschappelijke informatie (zie [10]) gebruikt worden. In de keten staat de samenwerking tussen de partijen centraal. Dit komt in de besturing tot uiting door het formuleren van indicatoren op de informatieproducten die tussen de partijen doorgegeven worden. Deze documenten vormen de interfaces tussen de partijen in het proces. De beschreven procesgang zal daarmee gemeengoed worden en het proces zal steeds beter gaan verlopen. Dit stimuleert de samenwerking tussen universiteit en uitgeverij. Er is echter buiten de gebruikte waardeketen weinig oog voor vernieuwing.

Vernieuwing wordt wel bereikt in het kwadrant rechtsboven. Daar start de besturing met de vraag welke belanghebbenden er zijn voor de publicaties. Vervolgens wordt onderzocht wat deze (verschillende) belanghebbenden belangrijk vinden aan de publicaties. De inrichting van de organisatie, inclusief de besturing, wordt hierop aangepast. Stel dat bedrijven nadrukkelijk als belanghebbenden gezien worden. Dan is de praktische toepassing van onderzoeksresultaten belangrijk. De nadruk komt dan te liggen op publiceren bij conferenties en op innovatieve vormen van publiceren. Bij dit laatste kan bijvoorbeeld gedacht worden aan communities op het internet waarbij publicaties gekoppeld worden aan discussiesites. De interactiviteit bevordert daarbij een praktische discussie over het onderzoek. Het kan zelfs nieuw onderzoek initiëren of een deel van de uitvoering van het onderzoek gaan vormen.

Het kwadrant rechtsonder streeft ook naar vernieuwing, maar op een beheerste manier. De randvoorwaarden en belangrijke uitgangspunten worden dan in een te gebruiken format geplaatst, zodat daar niet van afgeweken kan worden. De hoogleraar van een onderzoeksgroep kan bijvoorbeeld stellen dat 50% van de publicaties moet gaan over de wiskundige benadering van information filtering. De leden van de onderzoeksgroep kunnen – binnen gestelde grenzen – zelf het onderwerp van de overige publicaties bepalen.

7. Conclusies en aanbevelingen

Indicatoren vormen een belangrijk ingrediënt voor performance management. Voor het formuleren van indicatoren bestaan veel verschillende methoden. Het kiezen voor een bepaalde methode beïnvloed het type indicatoren dat uiteindelijk geformuleerd zal worden. We weten dat effectieve sturing alleen bereikt kan worden als de indicatoren zijn afgestemd op het doel van de organisatie. Het is dus zaak de juiste methode te kiezen voor het formuleren van indicatoren.

Vaak wordt onvoldoende nagedacht over de manier van formuleren van indicatoren. Veel organisaties kiezen tegenwoordig ‘klakkeloos’ voor de BSC methode. Ook worden nog vaak indicatoren ‘uit de losse pols’ geformuleerd. Dit komt de kwaliteit van sturing vaak niet ten goede.

In dit artikel hebben we ons sterk gemaakt voor een stapsgewijze aanpak in het formuleren van indicatoren. Als belangrijke eerste stap geldt het selecteren van een passende methode voor het formuleren van indicatoren. Daarna volgt pas het toepassen van de methode. Dit resulteert in een op de organisatie toegesneden methode. Ook de kwaliteit en bruikbaarheid van de indicatoren wordt hiermee verhoogd. We hebben aangegeven dat het vaak soelaas biedt om uit enkele basismethoden een gecombineerde variant te maken.

Als nuancering bij ons artikel stellen we dat het gebruik verschillende selectiecriteria kan helpen, maar niet zaligmakend hoeft te zijn. Uiteindelijk gaat het er ook om dat de organisatie vertrouwen in en een goed gevoel bij de methode heeft. Dat is niet altijd in criteria uit te drukken.

Literatuur

1. M. Goddard en R. Mannion, Horizontal and Vertical Approaches to Performance Measurement. In, Performance Measurement and Management: Research and Action, Juli 2002, Boston, USA.
2. M.J. Harry. The vision of Six Sigma, 8 volumes. Phoenix Arizona: Tri Star Publishing. 1998.
3. INK. Manual for assessing the position of businesses. Zaltbommel, The Netherlands: INK, 2001.
4. R.S. Kaplan en R. Cooper. Cost & Effect: Using Integrated Cost Systems to Drive Profitability and Performance. Harvard Business School Press. November 1997.
5. R.S. Kaplan en D. Norton. The Balanced Scorecard – Measures that Drive Performance. Harvard Business Review, January/February 1992.
6. R.S. Kaplan en D. Norton. The Strategy Focused Organization. Harvard Business School Press. 2000.
7. L.A.F.M. Kerklaan, J. Kingma en F.P.J. van Kleef. “De Cockpit van de organisatie”. Kluwer Bedrijfswetenschappen, 1994.
8. Mazur, G.H.. QFD for Service Industries. Proceedings of The Fifth Symposium on Quality Function Deployment, Novi, Michigan, 1993.
9. H. Mintzberg. Organisatiestructuren. Academic Service, Schoonhoven, Nederland. 1992.
10. H.E. Roosendaal, T.W.C. Huibers, P.A.Th.M. Geurts en P.E. van der Vet. Changes in the value chain of scientific information: economic consequences for academic institutions, Online Information Review, Vol. 27 (2), 2003.

11. G. Scheipers, A. Ameels en W. Bruggeman. Value-based management: an integrated approach to value creation. A literature review. In Peeters, L., Matthyssens, P. & Vereeck, L. (Eds.), Stakeholder Synergie: Referatenboek 25e Vlaams Wetenschappelijk Economisch Congres, Hasselt, 77-128, 2002. Leuven-Apeldoorn: Garant.
12. B.C.M. Wondergem en J. Eskens. Bestuurlijke begrenzingen van de Balanced Scorecard. Management & Informatie, augustus, 2003.
13. B.C.M. Wondergem en S. Verzijl. Grootmoeder is aan het stuur in contactcenters, Callcenter Magazine, september 2003.
14. B.C.M. Wondergem en N. Vincent. Transformations in Performance Management. Geaccepteerd voor publicatie in "Transformation of Knowledge, Information and Data: Theory and Applications", te verschijnen in 2004.
15. B.C.M. Wondergem en H. Wulferink. Prestatie-indicatoren – weten hoe je moet meten. Informatie, oktober 2002.

Metadata in Science Publishing

Anita de Waard (Advanced Technology Group, Elsevier)
Molenwerf 1, 1014 AG Amsterdam

Joost Kircz (KRA-Publishing Research) *
Prins Hendrikkade 141, 1011 AS Amsterdam
kircz@kra.nl

Abstract

In the design of authoring systems in electronic publishing a great challenge is to what extent the author is able, can be enabled and is willing to structure the contribution her/himself. After all, all information that is denotated properly in the writing stage enhances the retrievability later on. Metadata are the crucial ingredients. Hence, prior to design and experiment is the need for a full-fledged understanding of metadata. In this contribution we discuss an attempt to classify metadata according to type and use and elaborate on the many complicated and unsolved issues. The message of all this is that metadata should be treated on equal footing as the objects they describe, in other words metadata are information objects in themselves. We show that all issues that pertain to information objects also pertain to metadata.

1. Introduction

With the impressive growth of hyper-linked information objects on the World Wide Web, the best possible way of finding gems in the desert is to create a system of filters - sieves, that enable a large throughput of information in the hope that the residue is of relevance to the working scientist. Two methodological directions can be taken to find relevant information. One approach starts from the assumption that information growth cannot be tamed. Purely statistical information retrieval techniques are a prime example of such an approach, which can be void from any semantic knowledge about the content at stake. In these IR techniques, context is inferred from patterns that contain the query words. In the extreme case, not even words are used as in the powerful n-grams technique [1,2].

The other approach is based on denotating information. Every relevant piece of information is augmented with data describing the information object, so-called: metadata. Metadata can be seen as filters as they distribute information according to classes, such as a name, an address, a keyword, etc. Looking for the name of the person Watt, we only have to look in the class of authors, whilst looking for the notion Watt (as a measure for electric power) we only have to look in the class of keywords belonging to the field of electric engineering. Due to the ambiguity of words, normally metadata are added by hand or based on the structure of the information object, e.g., a document. In a standardised environment we can infer with 100% certainty what the name of the author is, which is impossible if we deal with a document with an arbitrary structure in a language we don't master.

It goes without saying that both approaches, purely statistical and pre-coordination are needed in a real life environment. Statistical approaches have a number of obvious problems (lack of semantic knowledge, inability to interpret irony or casual references), while full pre-coding by the author might on the one hand be impossible to achieve, and on the other hand prevent the browsing reader to stumble on unexpected relationships or cross-disciplinary similarities. The challenge is how we can prepare information in order to enable quick and relevant retrieval, while not overburdening the author or indexer.

In adding metadata to documents, more and more computer assisted techniques are used. Some types of metadata are more or less obvious, e.g., bibliographic information, while others demand a deep knowledge of the content at issue. At the content level we deal with authors who are the only ones who can tell us what they want to convey and professional indexers who try, with the help of systematic keyword systems, to contextualise the document into a specific domain. In particular the last craft is creating essential added value by securing idiosyncratic individual documents into a domain context, by using well designed metadata systems in the form of thesauri and other controlled keyword systems.

We are currently working on the design of a system which enables the author to add as much relevant information as possible to her/his work in order to enhance retrievability. As writing cultures do change as a

result of the technology used, we propose to fully exploit the electronic capabilities to change the culture of authoring information. In such an approach, it is the author who contextualises the information in such a way that most ambiguities are pre-empted before release of the work. Such an environment is much more demanding for the author and editor, but ensures that the context of the work is well-grounded.

To build a useful development environment, in this contribution we define different categories of metadata, that are created, validated and used in different stages of the publishing process. Given the importance of metadata, we believe it should be treated with the reverence usually reserved for regular data, in other words, we need to worry about its creation, standardisation, validation and property rights. In this contribution, we want to explore how metadata is used, and consider the issues of versioning, standardisation and property rights. We then come up with a proposed, and very preliminary, classification of metadata items, and discuss some issues concerning the items mentioned. As we believe that metadata should be treated on equal footing as the objects they describe, in other words metadata are information objects in themselves, we show that all issues that pertain to information objects also pertain to metadata.

This contribution is meant to support our own work in building an authoring environment, and therefore does not present any conclusions yet- but we invite responses to this proposed classification and the issues at hand (versioning, validation, standardisation and property rights of metadata). Preferably, based on comparison of documents of different scientific domains, as it turns out that different domains can have substantial differences in structure and style. As is clear from the above and in particular from the table, many issues are still uncertain and in full development. For the design of an easy to use and versatile author environment, where the author can quickly denote her/his own writing and create and name the links to connotate the work, an analytically sound scaffolding is needed before such a system can be built.

Below we discuss a classification of metadata leading to an overview presented in a table. Items in the table refer to further elaboration via hyperlinks. As this presentation also has to be printed, in this version the elaborations and digressions are located linearly as sections after the table.

2. Classification of metadata

2.1 Different types of Metadata

In first approximation we make a distinction into three broad categories of metadata, which are accompanied by three uses of information:

- Type: Descriptive of content. Here we deal with typifying information that pertains intellectual knowledge needed for understanding and placing the work in context. Typical items are: the author's name, keywords & classification codes, an abstract, captions to various enhancements such as figures. etc. It can be argued that author's names, abstracts, captions and references are not metadata, but simply content. However, data about data are not necessarily atomic. An abstract denotes a story, hence, we have included it in the table.
  Use: Interpret and validate. As the reader normally is disjoint in time and place from the originator the interpretation of a work depends on its context. Note that this context is not only a matter of proper semantic indexing, but also defined by the technology used. If the original is handwritten on parchment or typed with 8 bit WordStar, the reader can make interferences about the cultural/technological state-of -the-art at the time of creation.
- Type: Descriptive of location. In this category we deal with traditional bibliographic references and their modern extensions such as the Digital Object Identifier (DOI) as well as status information such as draft (normally on a home page), preprint (on a home page and on a preprint server), revised version, final version (in a certified journal from a publishing institution), etc. In web-based publishing many versions of the same article abound, and knowledge of an object's location has to be augmented with knowing its status. Location thus means physical location as well as location in the added value chain from draft to certified document.
  Use: 1-Locate and connect. Here we deal with the traffic to and from data such as links to a work, to an author/subject index, or between works.
  2- Interpret and validate. If it is located on a preprint server, it can receive a different scientific status than if it is located on an online version of a high-impact journal.
- Type: Descriptive of format. In an electronic environment we are blessed with a plethora of technical rendering possibilities. Hence, every information object needs a complete description of its technical format, so in this category we deal with issues such as: presentation versions (txt, pdf, wrd, wpd, html,

etc., etc.) and in structured environments with descriptors such a Document Type Definition (DTD) and XML data standards (SVG, MathML, etc.).
Use: Manipulate. This can involve e.g. rendering certain data types or running programs. We have to know how to represent the information or how to use the metadata for statistical approaches or datamining.

2.2 Creation

Metadata can be created by different parties - authors, editors, indexers and publishers, to name a few. It is important to realise that at some times, the creating party is not the validator; also, if the creating party is not part of the versioning cycle, the party creating the latest version can be not aware of necessary updates in the metadata. Therefore, only the creator can add and validate such items as her/his own name or references to other works. Additional metadata can be generated by machine intervention, such as automatic file-type and size identification, whilst professional indexers, be it by hand or computer assisted, will add domain dependent context to a work.

2.3 Validation

Very often, metadata is not validated per se. For convenience's sake, it is often assumed that links, figure captions, titles, references and keywords are correct. An extra challenge in electronic publishing is the validation of non-text items - for one thing, most reviewers and editors still work form paper, thereby missing are hypertextual and/or interactive aspects of a paper (hyperlinks that no longer work are an obvious example of this problem).

2.4 Rights

The role of Intellectual Property Rights (IPR) and Copyright in particular, is a hot issue in the discussions on so-called self-publishing. A great deal of difficulty is in the differences between the various IPR systems, in particular between (continental) Europe and the US.
However, besides this issue, electronic publishing generates a series of even more complicated questions that have to be addressed. As metadata allow the retrieval of information, they become "objects of trade" by themselves. Below we only indicate some issues pertaining to our discussion. A more detailed overview on the complicted legal aspects in ICT based research is given in Kampermann et. al ([3] and references therein). This short list below, shows that the heated debate on the so-called copyright transfer (or really: reproduction rights) from the author to a publishers is only a small part of the issue. Metadata as information objects face at least the same right problems as the document per se.

- What is a work? The role of databases
  An electronic document is a well-defined set of different kinds of elements: texts, images, tables, sets of hyperlinks, & (meta)data. The E-document is an envelope of independent objects. The E-document can then be considered as a new object with various levels of granularity that can be accessed separately. By nature an E-publication is part of a virtual world-wide database. As soon as works are loaded on a (publisher's) web-site and value is added by, e.g.: the maintenance of links, the conversion to a standardized storage scheme etc., we can speak of a database that can claim the database protections given in the EC council Database Directive [4]. Here, in Article 1, a database is defined as a collection of independent works, data or other materials arranged in a systematic or methodical way and individually accessible by electronic or other means. A publisher's database becomes an integrated whole, an object by itself and can claim rights! Hence, the fact that the database is ruled by metadata has a crucial impact on the IPR's of authors.
- Metadata structures as object
  In order to fully exploit the electronic capabilities an author has to create his/her work according to well-defined rules that enable storage in a multi-media format. The author creates a work including a metadata structure that guides the reader. The presentation and the content in the electronic version are converging to one representation. In a controlled publishing environment a publishing organisation (commercial or not) creates and maintains the metadata structure. This means that at least the intellectual ownership is with that organisation and the added value to the "database" in which this structure is implemented is a genuine new creation. In case of the Open Archive self-publishing initiative it is the author who adds a limited set of meta-data his/herself.
- Controlled keyword systems and thesauri.

The design and maintenance of a thesaurus or ontology is intellectual labour and hence can be considered a work with its own IPRs. Different parties in the value chain can generate profit from this. In a world where documents (in one version or another) swarm around in cyberspace, the keys to disclosure become an obvious commodity.

- Form versus content (content driven publications)
  The whole new industry of SGML/XML declarative languages is geared towards presentation independent storage and the capability to "render" the "content" on different "platforms". So called: single source, multiple delivery publishing. The present IPRs deal only with the "content".
- Real multimedia "documents" (layout driven publications)
  A scientific publication is a mixture of text and non-text elements and in some case even non-text elements only. This original version of a scholarly publication will be a multi-media "document". The paper instantiation becomes a spin-off, needed for those who want to carefully read and annotate the work. But this version is not necessarily (and in the near future even pertinently not) the e-version minus the "unprintable" objects. A text for reading demands another grammar than a multi-media document. Hence, a scientific publication will consist of two or more presentation forms, that all need certification, authentication and validation, just like the old paper-only publication. Each form presents an independent entity, and deserve independent IPRs.
  The publication environment becomes intrinsically a collection with added value, due to the structuring and interlinking of the elements. New extra value can always be added by keyword and classification indexes as well as link taxonomies (different kind of hyperlinks, each with a meaning of why the linking is added). Those extra metadata systems are creative products with their own IPR.
  There is a difference between a real multimedia document, where the various expressions (text and non-text) are a united whole and the present-day patch works of various types of media (in fact multiple-media documents). Real hypermedia documents (an integration of hypertext and multi-media in which time-lines and spatial lay-out are well-defined) will directly be specified in terms of the final presentation (lay-out driven), the segregation between structure and presentation disappears. In such a case Database directive art.5.b, that allows database owners to carry out.....translation, adaptation, arrangement and any other alteration, becomes under heavy pressure, as form and content together establish a creative whole.
- Software
  Apart from these issues, we also have to consider that e-publishing requires a series of software licences from the Operating System to the Video Manipulation Package. All those rights become an intrinsic element of the publication and readers need to know which licences they need, even for simply reading a text. Hence, an extra system of metadata describing the required software and its parameters (version, single or multi-user, etc.) is appearing.

2.5 Metadata classification

Using the categories defined above, we can come to a first list of metadata items, that include comments on their usage, creation/validation and rights, and define a number of issues, that are described in the paragraphs below.

What is it	Category	Who creates	Who validates	Who has rights	Issues
Author name	content	Author	Author	Author	Unique author ID (see below 3.1)
Author affiliation	content	Author's Institute	Editor? Publisher?	Author?	Corresponding author address only? / Present address vs. at the time of writing. In other words is the article coupled to the author and her institution during creation, or does an article follows an author in time.
Author index	content	Publisher	Publisher	Publisher/Library	Author name issues (Y. Li issue, see below 3.1)

Keywords	content	Author, editor, publisher, A&I service, library, on-the-fly	Editor, publisher, A&I, library	See section 2.4	Multi-thesaurus indexing (see below 3.2) Ontologies
Abstract	content	Author, A&I service	Editor, A&I editor	Author/A&I service	Types of abstracts? Usage of abstracts? (see below 3.3)
References	location	Author	Editor, Publisher	None for individual reference; document collection - yes	DOI, http as reference; link reference to referring part of document; versioning! See also Links (below 3.4)
Title, Section division, headers	content	Author/publisher	Publisher	Publisher?	Presently based on essayistic narratives produced for paper
Bibliographic info (publisher's data)	location	Publisher	Publisher	Publisher (TM)™	DOI refers to a document, but is intrinsically able to refer to a sub-document unit. No pagination in an electronic file, referencing is now point-to-point instead of from page-to-page.
Bibliographic info (Other data)	locate	Library	Library	Library	Multiple copies in a library system, signature, etc. Does this all evaporate with the new license agreements, where the document is hosted at the Publisher's database?
Clinical data	content	Author	Editorial	Doctor/patient?	Privacy; standardisation; usage?
Link (object to dataset, object to object)	location/ content	Author Publisher	Publisher	Author? Publisher?	Are information objects (see below 3.4)
Multimedia objects Visuals, Audio, Video, Simul-(Anim)ations	content/ format	Author, Publisher	Editor? Publisher?	Rights to format (cf. ISO and JPEG) vs. rights to content	Who owns SwissProt nr? Genbank ® nr? Chemical structure format? JPEG org? Issue of layout-driven vs. content-driven data Who validates the scientific value of such an object? We don't have yet referee standards as we have for text.
Document status, version	content	Editor, publisher, (author for preprint/OAI)	Publisher	Publisher	Version issue (see below 3.6) Updated version in preprint/Open Archive Initiative (OAi)I - which is the original? Multiple copy problem; virtual journals
Peer review data	content	Reviewer	Editor	Reviewer?	How to ensure connection to article? Privacy? vs Versions of articles? Open or closed refereeing procedures
Document	content/ location/ format	Author, Publisher Reviewer	Editor, Publisher	Author ("creator")	Integrity of components that make up document; Versioning. Intellectual ownership vs reproduction rights (see also 2.4)

DTD	content/ format	Publisher	Publisher	Open source, copyleft?	Versioning? Standard-DTD (see below 3.7) (Dublin Core)? Ownership
Exchange protocols e.g. OAI	locate/ format	Library, Publisher, archive	"Creator"	?!	Rights! Open standards Original copy issue standardization (ZING);
Document collection - Journal (e.g. NTvG)	content/ location/ format	Editor/Publisher	Editor /Publisher	Publisher	Integrity of collection - multiple collections E-version versus P-version
Document collection - Database (e.g. SwissProt)	content/ location/ format	Publisher - Editor?	Publisher	Organization?	Validation? Rights?
Data sets collaboratories - Earth System Grid	content/ location/ format	Federated partners	Nobody!	Creator?	Validation? Usage?

3. Some issues

The following elaborates on some of the issues raised in the table in the previous paragraph (connected by hyperlinks in the online version). This elaboration is needed as only after a full understanding of the qualities and values of the various types of metadata and their mutual relationships we can start with the system requirements of new types of authors' environments to be designed in close connection with the storage and retrieval environment of genuine electronic -multimedia- documents.

3.1 Unique Author ID

The demand for an unique author id is as simple as reasonable. However, in the real world we encounter the following caveats:

- How do we know that it is the same author? Many people have the same name such as: Y. Li, T. Sasaki, Kim Park, or Jan Visser.
- Many transliterations from non- European languages into the Latin alphabet are different. Happily most academic library systems now do have concordance systems in place. But still, many uncertainties remain in cases such as Wei Wang (or Wang Wei).
- Authors change address, institutes change name (and address), and this amplifies the problem.
- Authors sometimes change their name after marriage, immigration or change of sex. This might be minor problem to the above mentioned, but is a persistent and frequently occurring problem .

So, do we want to use a social security (or in The Netherlands SOFI) number or picture of an iris scan? Or even introduce a Personal Publishing Identification Number (PPIN)?
A lot of practical and legal issues still stand in the way of true unique identification, but first steps are being set on this path by publishers, agents and online parties to come to a common unique ID - the INTERPARTY initiative being one of them.

3.2 Controlled Keyword Systems

Indexing systems are as old as science. The ultimate goal is to assign an unambiguous term to a complex phenomena or reasoning. As soon a something has a name, we can manipulate, use and re-use the term without long descriptions. In principle, a numerical approach would be easiest, because we can assign an infinite number of ids to an infinite number of objects. In reality, as nobody things in numerical strings, simples names are used. However, as soon as we use names we introduce ambiguities as a name normally has multiple meanings
A known problem is that author added keywords normally are inferior to keywords added by trained publishing staff, as professional indexers add wider context where individual authors target mainly on terms

that are fashionable in the discussion at the time of writing, as the experience in the journal making industry learns. Adding uncontrolled index terms to information objects therefore rarely adds true descriptive value to an article, a prime reason to use well-grounded thesauri and ontologies.

A so-called Ontology is meant to be a structured keyword system with inference rules and mutual relationships beyond "broader/narrower" terms. At present we are still dealing with an mixed approach of numerical systems such as: Classification codes, e.g. in chemistry or pharmacology, and domain specific thesauri or structured keyword system such as EMTREE and MeSH terms in the biomedical field. Therefore, most ontologies still rely on existing indices, and ontology mapping is still a matter of much debate and research. Currently, multifarious index systems are still needed, based on the notion that readers can come from different angles and not necessarily via the front door of the well established Journal Title. Index systems must overlap fan-wise and links have to indicate what kind of relationship they encode. The important issue of rules and particular the argumentational structure of these roles is part of our research programme and discussed elsewhere [5, 9].

3.3 Abstracts

The history of abstracts follows the history of the scientific paper. No abstracts were needed when the number of articles in a field was fairly small. Only after the explosion of scientific information after WWII we see the emergence of abstracts as a regular component of a publication. Abstracting services came into existence and in most cases specialists wrote abstracts for specialised abstracting journals (like the Excerpta Medica series). Only after the emergence of bibliographic databases the abstract became compulsory as it was not yet possible to deliver the full text. After a keyword search, the next step towards assessing the value of retrieved document identifiers was by reading the on-line abstract. In an electronic environment (where the full article is as quickly on the screen as the abstract) the role of the abstract as an information object is under scrutiny, since for many readers, it often replaces the full text of the article. As already said in section 2.1, abstracts are identifiers for a larger information object: the document. In that sense an abstract is a metadata element.

In a study at the University of Amsterdam [6] to assess the roles of the abstract in an electronic environment, the following distinctions are made :

Functions:

1. Selection. You cannot read all articles published. Facilitates choice.
2. Substitution. All relevant information is in the abstract, e.g. essential experimental results.
3. Retrieval. "In fact, the ideal abstract from an indexer's point of view is a string of keywords linked into an easily read sentence" .
4. Orientation. In supporting those who read (parts of) the source text. In an electronic environment it can be the linchpin of all components.

Type of abstracts:

1. Characterizing. A brief indication of what is it all about. Often a clarification of the title. Often author created.
2. Slanted. Oriented to a well-defined audience. E.g. abstracts of biological articles for chemists. Often made by A&I service.
3. Extensive. Useful if the source text is not easily available. Often made by editor/ domain expert.
4. Balanced. Reflects all phases of the reasoning. Imported if the abstract fulfills and orientation function.

This analysis shows that the database field "abstract" now has to be endowed with extra specifying denotation. As our research is on design models for e-publishing environments, we have to realise that at the authoring stage of an abstract a clear statement about function and role is needed, as more abstracts -of a different type- might be needed to cater for different reader communities.

3.4 Hyperlinks

As already discussed above, analysing components of a creative work into coherent information objects, means that we also have to define how we synthesize the elements again into a well behaving (new) piece of

work. The glue for this puzzle are the Hyperlinks. An important aspect in our reserach programme is to combine denotative systems with named link-structures that add connotation to the object descriptors. By integrating a proper linking system with a clear domain-dependent keyword system, a proper context can be generated.

If we analyse hyperlinks we have to accept that they are much richer objects than just a connection sign, as:

- Somebody made a conscious decision to make that link and so a link has formally an originators or Author.
- A link has been made during a particular process where the relevance of making the link became clear. E.g., in a research process it becomes clear that there is a relationship with other research projects. Hence, a link has a creation date. In an even more fundamental approach one can say that the creative moment of linking one piece of information to another is a discovery and is linked to a creator like any other invention or original idea.
- Hyperlinks belong to a certain scientific domain. A reference in geology will normally not point to string theory. Hence, the point where the link starts is an indication for the information connected. It goes without saying that this is never completely the case as a geologist might metaphorically link to a publication on The beginning of Time according to mathematical physics.
- Links can carry information on the reason of linking (see below).
- Most important, links carry knowledge! They tell us something about relationships in scientific discourse.

All in all, hyperlinks are information objects with creation date, authorship, etc. and hence, can be treated like any other information object. This means that we have to extent our discussion of metadata as data describing information to hyperlinks.

Apart from the obvious attributes such as author, date, etc. we can think about an ontology for links. This ontology will be on a more abstract level than an ontology of objects in a particular scientific field as we here we deal with relationships that are to a large extent domain independent.
A first approach towards such system might go as follows:

A) Organisational

- Vertical. This type of links follow the analytical path of reasoning the relations are e.g., hierarchical, part-of, is a, etc.
- Horizontal. This type of link points to sameness and look alikes, such as: see also, siblings, synonyms, etc.

B) Representational

- The same knowledge can be presented in different representations depending on the reading device (PDA, CRT, Print on paper) or by the fact that some information can be better explained in an image, a table or a spread- sheet. Therefore we have a family of Links that relate underlying information (or data-sets) to a graph, a 3D model, an animation or simply a different style-sheet. As these links will also related different presentations of the same information, if available, many of these links might be generated automatically.

C) Discourse
The great challenge in designing a link ontology, and metadata system is in developing a concise but coherent set of coordinates. As discussed in more detail elsewhere [7, 8].
We suggest the following main categories:

- Clarification (link to educational text)
- Proof (link to mathematical digression elsewhere, link to law article, etc.)
- Argument e.g., for/ against different author

In conclusion: as links are information objects we have to be aware of validation and versioning the SAME way as textual or visual objects and data-sets!

3.5 Standardization

In an electronic environment where documents (or parts thereof) are interlinked, no stand-alone (piece of) work is created/edited/published anymore. All creative actions are part of a network. So, all parties need to discuss and use standards: (partly) across fields, (certainly) across value chains. However "The great thing about standards is that there are so many to choose from..." and that they evolve all the time.

In library systems, we rely on a more or less certified system of index terms such as Machine-Readable Cataloging (MARC) records, where a distinction is made between: Bibliographic, Authority, Holdings, Classification and Community information. In a more general perspective we see all kind of standardisation attempts to ensure interchange of information in such a way that the meaning of the information object remains intelligible in the numerous exchanges over Internet. {See e.g.. The National Information Standards Organization (NISO) in the USA for the Information Interchange Format and The Dublin Core Metadata Element Set}.

An immediate concern is the level of penetration of a standard in the field and its, public or commercial, ownership. Who has the right to change a standard, who has the duty to maintain a standard, how is the standardisation work financed and who is able to make financial gains out of a standard? For that reason the discussion of standardisation and Open Standards in particular are crucial in this period of time.

3.6 Versioning and modularity

Today's authoring tools allow the easy production of many different versions of a publication prior to certification. Often drafts are sent around to colleagues for comments. It is not unusual that drafts are hosted on a computer system that allows others to approach the directory where the draft is located. Comments are often written into the original work and returned with a new file name. That way, many different versions of a document float around without any control and without any guarantee that after the drafting process is closed an a final work is born, all older versions are discarded. The same problems occurs again in a refereeing process if the paper resides on a pre-print server. All this forces the installment of a clear versioning policy. In practice this means that the metadata describing a work (or parts thereof) must have unambiguous data and versioning fields, indication not only the "age" of the information but also its status.

An interesting new phenomenon appears here. As is well known, in may fields so-called salami publishing is popular. Firstly a paper is presented as short contribution on a conference, than a larger version is presented on another conference and after some iterations, a publication is published in a journal. It is also common practice that people publish partial results in different presentations and then review them again in a more comprehensive publication. This practice can be overcome if we realise that an electronic environment is essentially defined as an environment of multiple and re-use of information. The answer to the great variety of versions and sub-optimal publications might lie in a break up of the linear document into a series of inter-connected well defined modules. In a modular environment the dynamic patchwork of modules allows for a creative re-use of information in such away that the integrity of the composing modules remain secured and a better understanding of what is old and what is new can be reached. Such an development is only possible if the description of the various information objects (or modules) is unique and standardised [7, 8, 9,10].

3.7 DTD

As said in section 2.1, the description of the format of the information is an essential feature for rendering, manipulating and datamining the information. This means that we need a full set of technical describers identifying the technical formats as well as identifiers that describe to structure of the shape of the document. Opposite to the simple technical metadata, e.g., are we dealing with ASCII or Unicode, the metadata that describe the various linguistic components and the structure of a document are interconnected one way or the other. This means that we need a description of this interconnection, hence metadata on a higher level.
A Document Type Definition (or its cousin, a Schema) defines the interrelationship between the various components of a document. It provides rules that enable checking (parsing) of files. For that reason a DTD, like an abstract belongs to the metadata of a document.

- E.g. A name field MUST have a Family name, must have at least a first initial, may have a second name/initial, may have pre- and post particles.

Based on such a skeleton DTDs and Style Sheets can be designed that keep the integrity of the information

(up to a -to be defined- level) tailored to various output/ presentation devices (CRT, handheld, paper, etc.).

- E.g. If a full first (or subsequent) name(s) is available, then on paper it is spelled out in its entirety, but on a handheld we only see the first initial.
- E.g. If colour is essential but the output device does not support colour, a message is added to the presentation.

Within this problem area it is important to mention the difference between content driven publications, i.e. publications that allow different presentations of the same information content and can be well catered for by a DTD and lay-out driven publications, which are publications where e.g., the time correlation between the various elements is essential for the presentation. See e.g. the work done at the CWI [11].

*) Also at : Van der Waals-Zeeman Institute, University of Amsterdam and the Research in Semantic Scholarly Publishing project of the University Library, Erasmus University, Rotterdam

4. References

1. Marc Damashek. Gauging Similarity with n-Grams: Language-Independent categorization of text. Science. vol 267. 10 February 1995. pp 843-848.
2. Alexander M. Robertson and Peter Willett. Applications of n-grams in textual information systems. Jnl of Documentation vol 54. no.1 January 1998, pp 48-69.
3. European Research Area Expert Group Report on: Strategic Use and Adaptation of Intellectual Property Rights Systems in Information and Communications Technologies-based Research Prepared by the Rapporteur Anselm Kamperman Sanders in conjunction with the chairman Ove Granstrand and John Adams, Knut Blind, Jos Dumortier, Rishab Ghosh, Bastiaan De Laat, Joost Kircz, Varpu Lindroos, Anne De Moor. EUR 20734 — Luxembourg: Office for Official Publications of the European Communities. March 2003 — x, 78 pp. — 21,0 x 29,7 cm. ISBN 92-894-6001-6.
4. Directive 96/6/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases. Official Journal L 077, 27//03/1996 p.0020-028.
5. Joost G. Kircz. Rhetorical structure of scientific articles: the case for argumentational analysis in information retrieval. Jnl. of Documentation, vol.47, no.4, December 1991, pp. 354-372.
6. Maarten van der Tol. Abstracts as orientation tools in a modular electronic environment. Document Design, vol. 2:1, pp.76-88, 2001.
7. J.G. Kircz and F.A.P. Harmsze. Modular scenarios in the electronic age. Conferentie informatiewetenschap 2000. Doelen, Rotterdam 5 april 2000. In: P. van der Vet en P. de Bra (eds.) CS-Report 00-20. Proceedings Conferentie Informatiewetenschap 2000. De Doelen Utrecht (sic), 5 april 2000. pp. 31-43. and references therein.
8. Joost G. Kircz. New practices for electronic publishing 1: Will the scientific paper keep its form. Learned Publishing. Volume 14. Number 4, October 2001. pp. 265-272.
   Joost G. Kircz. New practices for electronic publishing 2: New forms of the scientific paper. Learned Publishing. Volume 15. Number 1, January 2002. pp. 27-32. See: www.learned-publishing.org
9. F.A.P. Harmsze, M..C. van der Tol and J.G. Kircz. A modular structure for electronic scientific articles. Conferentie Informatiewetenschap 1999. Centrum voor Wiskunde en Informatica, Amsterdam, 12 november 1999. In: P. de Bra and L. Hardman (eds). Computing Science Reports. Dept. of Mathematics and Computing Science. Technische Universiteit Eindhoven. Report 99-20. pp. 2-9.
10. Frédérique Harmsze. PhD Thesis, Amsterdam, February 9, 2000. A modular structure for scientific articles in an electronic environment.
11. Jacco van Ossenbruggen. PhD Thesis, Amsterdam, April 10, 2001 . Processing Structured Hypermedia- A matter of style.

URL's mentioned

DOI: http://www.doi.org
Elsevier: http://www.elsevier.com
Dublin Core: http://dublincore.org/
Earth Systems Grid: http://www.earthsystemgrid.org/
Genbank: http://www.ncbi.nlm.nih.gov/Genbank/index.html
Interparty: http://www.interparty.org/
JPEG-Org: http://www.jpeg.org/

JPEG: http://www.theregus.com/content/4/25711.html
KRA: http://www.kra.nl
Marc: http://www.loc.gov/marc/
NISO: http://www.niso.org/standards/
NTvG: http://www.ntvg.nl/
OAi: http://www.openarchives.org/
Ontologies: http://protege.stanford.edu/ontologies/ontologies.html
Research in Semantic Scholarly Publishing project: http://rssp.org/
Swissprot: http://www.ebi.ac.uk/swissprot/index.html
Trec: http://trec.nist.gov/
ZING: http://www.loc.gov/z3950/agency/zing/zing-home.html

Federating Resources of Information Systems: Browsing Interface

Andrei V. Malchanau, Paul E. van der Vet & Hans E. Roosendaal
Department of Computer Science, University of Twente
P.O.Box 217 7500AE Enschede, The Netherlands
{a.v.malchanau, p.e.vandervet, h.e.roosendaal}@utwente.nl

Abstract

Designing the user interface of a federated system (what we call a browsing interface) must consider the knowledge gap that exists between desires of the users and the needs the systems are built to support.

The concept of Habitable Interfaces aims to bridge the knowledge gap by providing kinds of representations and the interaction with these representations that are based on domain knowledge. Habitable Interfaces will allow the organising of currently disparate archives into cohesive domain specific federations of information resources.

To approach designing Habitable Interfaces we propose a model of communication and a criterion.

1. Introduction

There are varieties of information resources that are available for scientists through the Internet. These resources are heterogeneous such as databases and archives of documents and multimedia. More, there are resources that run algorithms rather than retrieving data. In many cases, these resources are built around a particular need of a local group of scientists that are collecting data (or writing algorithms) for a particular reason. Clearly, maintenance of this variety of information resources cannot be centralised. On the other hand, the scientists need an access to these resources regardless from the underlying technological differences. Federated systems aim at providing an access to and combining information from disparate and heterogeneous information resources.

There are several ongoing efforts worldwide aimed at designing federated systems as well as data warehouses.Some examples are [1, 4, 5, 6, 14]. Most of this work is being done on combining data and solving technological issues of creating federated systems. Although having resources readily accessible is a necessary condition, the user interface makes a difference between a collection of independent information resources and a federated system.

Van der Vet (2000) proposed a research environment to alleviate some of the issues of accessing web-based information resources.

As we noted above, there are many organisations maintaining information systems, and their number grows by the day (see, for example, an overview of information resources in molecular biology in [12]). Individual research groups generally will want to leave maintenance of these resources to the groups who created them. The organisation of the access to existing resources should better be based on federating these resources rather than on integrating them into monolithic systems [8].

When federating information resources a number of high level issues should be addressed:

- Interaction of the user with the system: retrieving data, interacting with these data and receiving user feedback
- Representing information about the content of a federated system and retrieved data to the user
- Combination of the data (a common schema and algorithms for combining data from different resources)
- Accepting, planning and optimising user's queries
- Communication with the resources

Building federated systems requires a design of the user interfaces that will allow users to utilise available information effectively and efficiently. Existing approaches to the design may not fit the scale of the federated systems.

Perhaps, the most important issue in creating a federated system, is the gap between a variety of possible views and classifications of the same facts and rules that constitute knowledge, on the one hand and the limited representation that a designer can show to the user on the other hand. In the following part, we consider some issues of this gap in more detail.

Interaction of the user with the system and representing information about the content of a federated system are the most relevant topics for this paper. This brings us to the concept of Habitable Interface.

2. The knowledge gap

Federating information resources brings up the issue of the gap between varieties of views on information stored in an archive and the necessarily limited design of the archive. Further, we argue that the gap is inherent in the communication process and calls for new approaches.

Scientists are engaged in a knowledge discovery process. Knowledge is accumulated by collecting data. Collecting data requires a model that serves the purpose of practical guidance. Knowledge discovery is a collective effort, and a collective effort needs communication. In communication, researchers have generally different roles of authors and readers [13]. Given the variety of purposes that knowledge can be applied to, and the variety of data models, it is next to inevitable that there is a mismatch between the reader's and the author's data model. The situation worsens, when there are many readers and many authors who are trying to communicate on similar issues.

The archive can be perceived as an intermediary between authors and readers. Building an archive requires yet another data model (Figure 1). Differences between an archive's data model and authors' data models are not an obstacle for communication as it is only a question of converting known data using known data models.

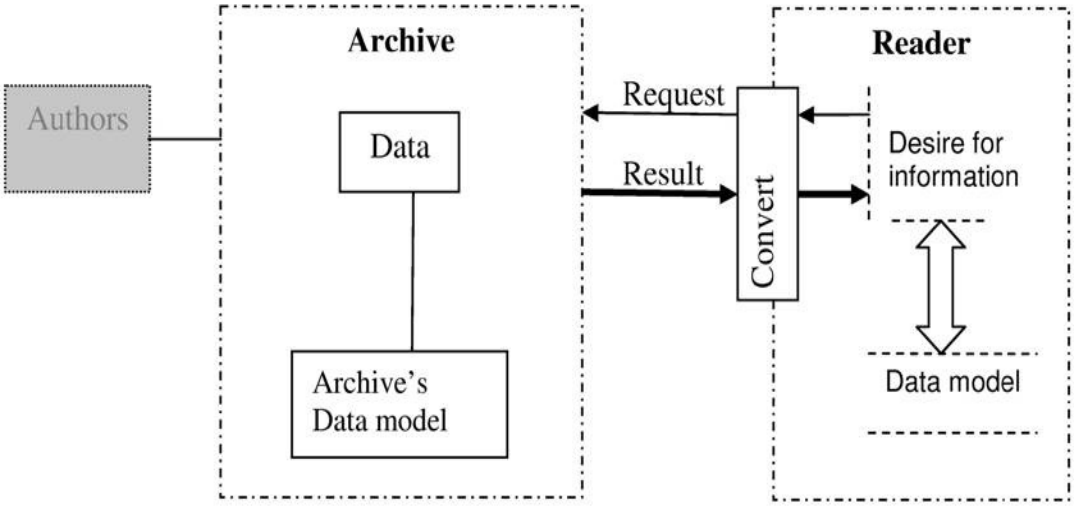


Figure 1. Communication between archive and reader

But for readers the situation is different. Readers do not need to know the archive data model and they do not want to know the archive data model, as it does not fit their mental frame. As a consequence, there is a gap between what we call desire (expressing what information the reader wants to know), on the one hand, and need (referring to the information in the archive's terms), on the other hand.

To fill the gap, an archive could convert data into a form required by readers. Multiplicity and dynamics of readers' interests present too great a challenge for designers of archives and in principle, even the best study of requirements would not provide a uniform representation of the readers' interests. Indeed, there is no average reader and there are many different archives.

3. Habitable Interfaces

Habitable Interfaces can help users to convert their desires for information into information needs that are then being communicated to the existing information resources. To arrive at an approach to designing Habitable Interfaces we start from a high-level model of communication between the reader and the archive. This model is rooted in other models proposed in the literature on Information Seeking and Information Retrieval: there are several overviews of the models and the concept of information in general (see for example [2, 10]). Here we would like to briefly consider the model proposed by R.S. Taylor [15] who describes the process of asking questions as starting from the 'visceral need':

- Visceral need – a vague dissatisfaction with the current knowledge about some topic;
- Conscious need – conscious understanding of what kind of information (knowledge) is missing;
- Formalised need – the formal statement of the need;
- Compromised need – the query that is presented to the archive

The model suggests that users formulate a query in several steps. In later experiments [3] the last three steps were reportedly observed. But Taylor's model does not explain how the visceral need is being converted into a compromised need. We believe that this conversion depends on the design of the retrieval system. An experimental investigation on such models has to generalize beyond the design of the system used in the investigation. In other words if the system design implies certain behaviour of the user it is likely to induce such a behaviour. For example with some interfaces, the readers have to explore the archive, with others they have to know the terms used in the archive before they can search for the desired information.

The significance of the Taylor's model for Habitable Interfaces is that it postulates that a request to the archive is a result of converting a particular 'inadequacy' in a reader's knowledge about some topic.

We add to this model of communication a model of the system. This allows stating a hypothesis about the system design that can be validated using empirical data.

To show how our model can be constructed we consider the communication process that takes place between the reader and an archive.

First, before starting the communication process, the reader has a certain desire for information. The word 'desire' implies a strong intention or aim. It is in contrast with the 'need' that is in general defined as a lack of something requisite or useful. Figure 2 shows this distinction from a number of viewpoints.

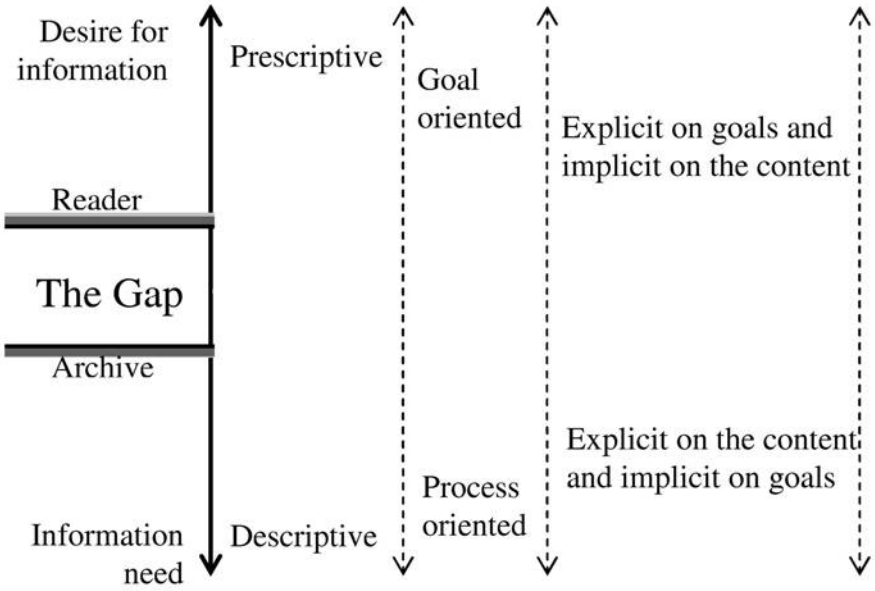


Figure 2. Distinction between desires and needs. There is a gap if the reader and the archive are at the different levels.

Based on the above we arrive at the following characteristics of the model of communication between reader and federated archive:

- The readers have knowledge that can be divided into classes serving the purpose of building a model.
    - Domain knowledge is a set of facts and rules that are known within a certain research domain. In a federated archive, this knowledge can form a basis to organise disparate resources.
    - We refer to the knowledge about the current situation as the situation perceived by the reader. It may be not an adequate understanding of the situation by the reader, but for our purposes, this is not relevant. Situations, in which the reader can be, may be rather divers. If we add to this an individual interpretation of the situation, it is clear that this knowledge will be specific for any individual reader. The knowledge about the current situation will set the context, within the domain, for the information being communicated.
    - The knowledge about the system or the language of the system is needed for converting the desire for information into a query that is comprehensible for the system and for converting results returned by the system into a form that the reader understands. The knowledge about the system, as any knowledge on communication, can be considered at four levels: lexical, syntactic, semantic and pragmatic. The fourth pragmatic level in this classification is the level at which the desire for information is communicated.
- The desire for information stems from the current situation and it is based on domain knowledge. The desire also indicates some lack of domain knowledge;
- Readers do not need to know the language of an archive and many readers will lack knowledge about the archive and its language;
- The gap between the desires for information of the readers and the information needs supported by archives can be viewed as the combined effect of lack of domain knowledge and lack of knowledge about archives;
- Converting this lack of knowledge into communication requests generates different sorts of behaviour, that, in general, depend also on the design of the archive;
- The communication process stops when the reader is satisfied.

This model is depicted in Figure 3. Figure 3 also shows the way the information resources can be organised into a federated archive. An important question to be answered based on this model is the design of the "Unravel" and "Combine" functions. "Unravel" function presents to the reader what is available in the federated archive and allows building a comprehensive set of queries. In the federated archive, the query has to be "Translated" into requests to individual resources, since the internal representation of the federated archive differs from that of an individual resource. The results returned by the resources might need translation, too. Furthermore, these results must be combined into a single representation for the reader. However, the particular implementation would require answers to questions such as:

- What are the principles to organise the representations of the information sent to and retrieved from the different resources?
- What sort of interaction between the representations in the "Unravel" and the "Combine" functions is useful and adequate for the reader?

Our model suggests that the representations should be based on the domain knowledge and the interaction with these representations should be designed so that it requires minimal knowledge of the system.

The design of these functions serves to reduce the requirements on the reader's knowledge about the system, and to improve the efficacy and efficiency of the communication.

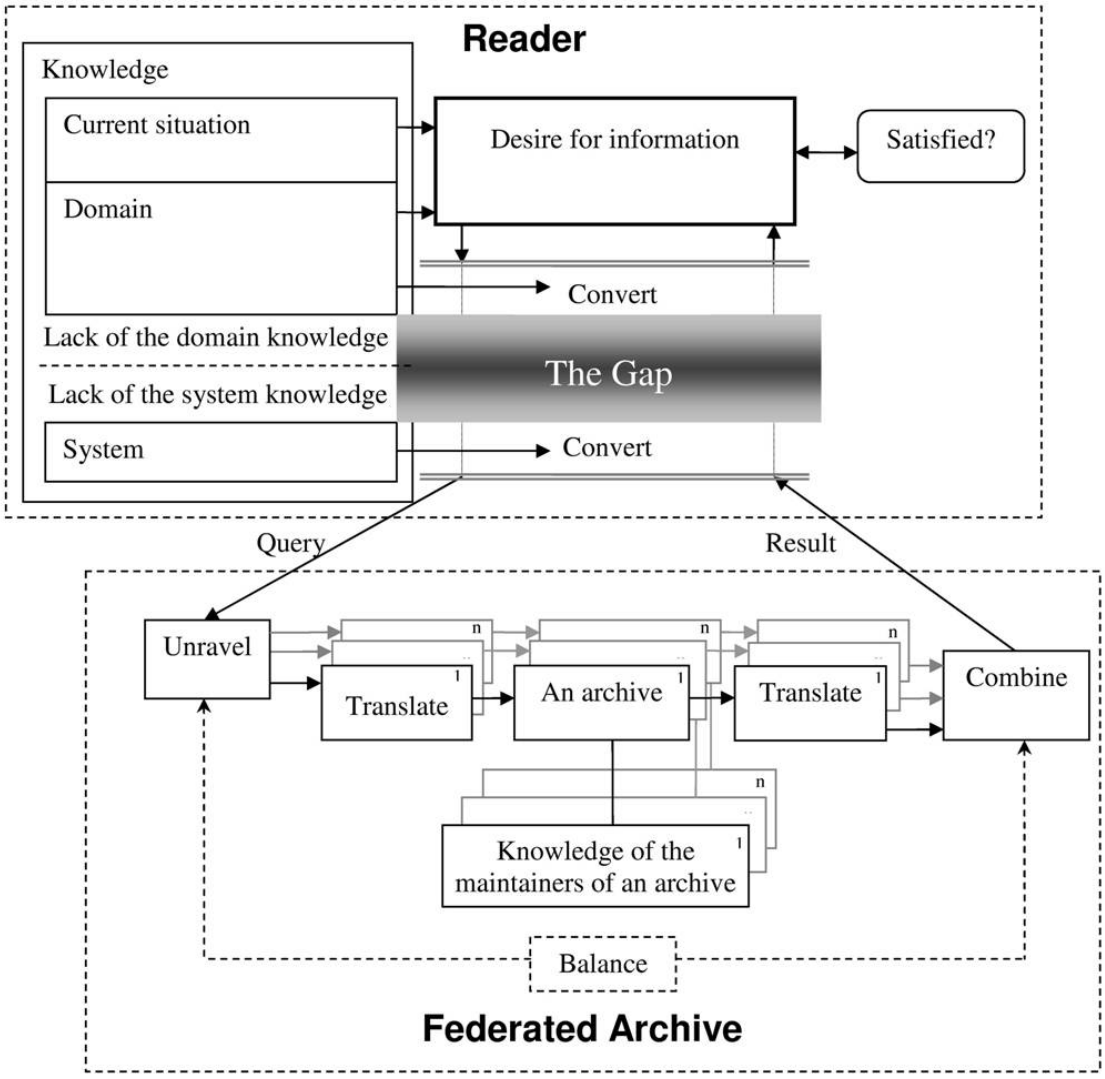


Figure 3. A model of Habitable Interfaces

3. Evaluation

For the high-level model to be applicable in designing a federated archive, it needs to be empirically validated. Such an evaluation can only be based on a priori agreed criteria and a method for evaluation. We argue that trust is a good indicator of the quality of the scientific communication that takes place and can be assessed using a mix of quantitative and qualitative methods.

In their communication, scientists are sharing with and delegating to the archive some of their tasks. In this perspective the reader, the trustor, should be able to trust the archive, the trustee, in this process of communication. This level of trust is posed to be a good indicator of the quality of information exchange. More on the relation of trust to scientific communication can be found in [9] and to information science and technology in [11].

4. Conclusions

Federating information resources requires new approaches towards designing user interfaces. The main issue is to deal with the degree of complexity and the scale of the integrated system.

On top of the complexity of the system there is a gap between the desires the users have and the needs that are supported by the systems.

The concept of Habitable Interfaces aims at helping the user to bridge this gap by means of incorporating the domain knowledge into representation and interaction.

At present, we work on the empirical validation of the model of Habitable Interfaces. The criteria of the validation will be based on the level of the user's trust in a federated system.

In addition, we would like to explore how intelligent agents may support the user in carrying out routine but specific tasks.

References

1. Blanco, J.M., Illarramendi, A., Goni, A. (1994) Building a Federated Relational Database System: An Approach Using a Knowledge-Based System, Int. Journal of Intelligent and Cooperative Information Systems, Vol.3, No.4 pp. 415-455
2. Capurro, R. & Hjørland, B. (2003) The concept of Information. In B. Cronin (Ed.), Annual review of Information Science and Technology, Vol. 37 pp. 343-411. Medford, New Jersey: Information Today
3. Chen, H. & Dhar, V. (1990) Online Query Refinement on Information Retrieval Systems: A Process Model of Searcher/System Interactions. In Proceedings of the 13th International Conference on Research and Development in Information Retrieval pp. 115-133. Brussels, Belgium, 5-7 September 1990
4. Chen, J., DeWitt, D., Tian, F. & Wang, Y. (2000) NiagaraCQ: A scalable continuous query system for internet databases. In Proc. of the ACM SIGMOD Conf. on Management of Data pp. 379-390
5. Davidson, S., Crabtree, J., Brunk, B., Schug, J., Tannen, V., Overton, C. & Stoeckert, C. (2001) K2/kleisli and GUS: Experiments in integrated access to genomic data sources. IBM Systems Journal, 40(2) pp. 512-531
6. Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajamaran, A., Sagiv, Y., Ullman, J., Vassalos V. & Widom, J. (1997) The TSIMMIS Approach to Mediation: Data Models and Languages, Journal of Intelligent Information Systems, 8 pp. 117-132
7. Gonçalves, M. A., France, R. K. & Fox E. A. (2001) MARIAN: Flexible Interoperability for Federated Digital Libraries. Research and Advanced Technology for Digital Libraries: Proceedings of the 5th European Conference, pp. 173-186. ECDL-01 (Darmstadt, Germany: 4-9 Sept.) Springer
8. Gray, P.M.D., Kemp, G.J.L. (2000) Federated database technology for data integration: lessons from bioinformatics. In Koslow, S.H., Huerta, M.F. Electronic collaboration in science pp. 45-72. Mahwah NJ: Lawrence Erlbaum
9. Hummels, H. & Roosendaal, H. E. (2001) Trust in Scientific Publishing, Journal of Business Ethics 34 pp. 87-100
10. Ingwersen, P. (1992) Information Retrieval Interaction. London: Taylor-Graham. X, 246 p.
11. Marsh, S., Dibben, M.R. (2003) The role of trust in Information Science and Technology. In B. Cronin (Ed.), Annual review of Information Science and Technology, Vol. 37 pp. 465-498. Medford, New Jersey: Information Today
12. Reed, J. (2000) Trends in Commercial Bioinformatics. Oscar Gruss
13. Roosendaal, H. E. & Geurts, P. A. Th. M. (1997) Forces and Functions in Scientific Communication: an Analysis of their Interplay, Proceedings of the Conference on Co-operative Research in Information Systems in Physics, University of Oldenburg, Germany, September 1-3
14. Stevens, R., Baker, P.G., Bechhofer, S., Ng, G., Jacoby, A., Paton, N.W., Goble, C.A. & Brass, A. (2000) TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. Bioinformatics 16(2) pp. 184-186
15. Taylor, R.S. (1968) Question-negotiation and information seeking in libraries. College and Research Libraries, 29 pp. 178-194
16. Vet, van der, P.E. (2000) Building web resources for natural scientists, in: Interactive distributed multimedia systems and telecommunication services (Proceedings IDMS2000), H. Scholten and M.J. van Sinderen (eds.), Berlin: Springer, (LNCS 1905), pp. 205-210

Profile-based retrieval on the World Wide Web

B. van Gils, H.A. Proper, P. van Bommel E.D. Schabell
basvg@acm.org, e.proper@acm.org, pvb@cs.kun.nl, erics@cs.kun.nl

Abstract

In this article we present a novel architecture for Information Retrieval on the Web called Vimes. This architecture is based on a broader definition of *relevance*. This broader definition lies in the fact that there is more then just topical relevance. Documents (or: resources) must also confirm to other constraints with regard to form, format and things like price and quality.

1. Introduction

In today's ``information society'' information plays an increasingly important role. The trick is to get the right information at the right time and in an appropriate format for a given goal. Finding the right information has been researched extensively in the IR-field. As recently as the 1970's people tried to devise computer programs to assist them in their search for information. These computerized searches started with searching in homogeneous document collections such as in the STAIRS-project [21]. The search process became more elaborate with the growing use of the Web. It led to the introduction of search engines such as GOOGLE which not only indexes (hyper)text, but also images, PDF-documents and interactive databases such as Citeseer [8]. In other words, search engines attempt to retrieve relevant *resources*, rather than documents alone.

The importance of the timing aspect is particularly obvious when investment decisions are involved, such as on the stock market. Getting some information late could have huge (financial) consequences. Implementing a strategy for getting information in time often depends on many things such as choosing the right partner/supplier: some news sites are `faster' then others in picking up news.

The third aspect mentioned deals with formats in the broad sense. It refers to ``file format'' (e.g. PDF, or HTML) as well as ``structural format'' (e.g. ``abstract'', or ``photograph''). The file format issue has been around since the early days of computing. Since people use different tools for jobs such as text processing a need for conversion tools between the file formats arose. Many of these conversions are available today. This is not (yet) the case for the latter issue, even though attempts have been made. A good example of this type of software is a computer program that generates abstracts for expository text (see e.g. [2]).

It is apparent that these factors vary for different users of IR-systems. For some users it is ok if certain financial records arrive slightly late, whereas for others it might have unpleasant consequences, some people would prefer an abstract of a (large) report over its full text, etc. In other words, each of these factors can be seen as a *characteristic* of a searcher. Loosely defined, a *profile* is the collection of all characteristics of a searcher that are relevant for the retrieval process.

The goal of this paper is to present a broader definition of what *relevance* is and to show how this can be used in IR. This broader notion of relevance is based on the mentioned issues and will be presented in Section 4. To this end, we briefly present a model for information supply in Section 3. This model (based on [14, 15]) allows us to introduce transformatons which are essential to the introduction of our prototype retrieval architecture in Section 5. Section 2 introduces the profiles which are used in our architecture (called Vimes).

2. Profiles

Already in [19] it was determined that information retrieval systems can be personalized for users by means of profiles. For years a lot of research has been invested in the area of user profiles. Often, these profiles are used to enhance the query by capturing the user's notions of query terms (see e.g. [7, 19, 20]). However, profiles can be used more extensively. For example, in [16] profiles are used for access control. We define that:

Profile
: A (user) profile consists of a set of preferences with regard to behavior of a search engine as well constraints on the results it presents to the user.

To illustrate this definition, the following list are the items that make up a particular user-profile:

preferences
: I prefer a maximum of 25 results per page, and by selecting a relevant resource (clicking on the link) will open a new window.

constraints
: I prefer HTML and PDF formats and refuse the Microsoft DOC-format. Furthermore, the size of the resource should not exceed 25Mb.

Using this definition, there are two areas in the retrieval process where profiles can be used. Firstly, they can be used for *post-processing* the results of the ranking process. For example, an resource that was found to be topically relevant can be converted to the proper format (See Section 4). Furthermore, profiles can be used to make sure that the retrieval engine operates according to the user's wishes.

In the previous section we explained what profiles are and what they can be used for. In this section we present a *possible* format for storing these profiles, whereas in the next section we explain how/where they are stored exactly.

Since we want the profiles to be re-used across (Web) search engines, the format should be an *open standard.* More specifically, we want our format to be machine understandable and interoperable. The eXtensible Markup Language (XML, see e.g. [5]) is particularly well suited for this task (see e.g. [24]). The following XML-fragment is an example of what a profile could look like:

```
<? xml version="1.0" ?>
<!-------------------------------------------------------------- -->
<!-- A profile has an owner, identified by his/herEmail-address. -->
<!-- Furthermore, a check-sum is included for security purposes. -->
<!-- This profile stores 3 characteristics.                      -->
<!-------------------------------------------------------------- -->

<!-- define the owner of the profile -->
<profile owner="Basvan Gils" email="bas.vangils@cs.kun.nl"cs="2768A493">

    <!-- 1stcharacteristic: how many results per page? -->
    <characteristic type="results">
        <page> 25 </page>
    </characteristic>

    <!-- 2nd charactersitic: the max. size in Mb -->
    <characteristic type="max_size">
        <mb> 5 </mb>
    </characteristic>

    <!-- 3rd characteristic: preferred file-types -->
    <characteristic type="file_type">
        <type nr="1"> HTML </type>
        <type nr="2"> PDF </type>
        <type nr="3"> PS </type>
    </characteristic>
</profile>
```

Note that this excerpt is intended to illustrate our ideas. Defining a formal DTD for profiles is part of future research.

3. The model

Our model of information supply is based on the distinction between data and information. The entities found on the Web, which can be identified by means of a URI [3], are *data resources*. These data resources *can* be

information, if and only if they are relevant with regard to a given information need. Also, we presume that many data resources can, at least partially, convey the same information. Hence, we define *information resources* to be the abstract entities that make up information supply. Each information resource has at least one data resource associated to it. Consider for example the situation in which we have two data resources: the painting Mona Lisa, and a very detailed description of this painting. Both adhere to the same information resource in the sense that a person seeking for information on `the Mona Lisa' will consider both to be relevant.

In a way, the data resources *implement* the information resources; a notion similar to that in [12] where `facts' in the document subspace are considered to be `proof' for hypotheses in the knowledge subspace. Note that each data resource may inplement the information resource in a different way. We define a *representation type* to indicate exactly how a data resource implements the information resource it is asociated to. Examples of representations are full-content, abstract, keyword-list, extract, audio-only, etc.

Many different types of data resources can be distinguised on the Web today, such as documents in different formats (HTML, PDF, etc), databases, interactive Web-services, etc. Hence, each data resource has a *data resource type* Furthermore, data resources may have several attributes such as a price or a measurement for its quality. Such attributes can be defined in terms of an *attribute type* and the actual value that a data resource has for this given attribute type.

Also, *values* can be attributed to data resources. For example, the value ``640x480'' can be used to denote the resolution of an image, or €20 the price of a data resource. We model this by defining that combinations of data resources and values associated to these data resources have an attribute type.

Last but not least, data resources can be interrelated. The most prominent example of this interrelatedness on the Web is the notion of *hyperlinks* [6, 9], but other types of relations between data resources exist as well. Examples are: an image may be part of a webpage, a scientific article may refer to other articles, etc.

Figure 1 shows the General Model for Information Supply, which is based on the following verbalisation:

- Information Resources have at least one Data Resource associated to them;
- A Representation denotes the unique combination of an Information Resource and a Data Resource
- Representations have at least one Representation Type
- Data Resources have at least one Data Resource Type
- Data Resources are related via Relations with a source and a destination.
- Relations have at least one Relation Type.
- Data Resources may have attributed values
- Attributes have at least one Attribute Type

The fact we have several *types* in our model indicated heterogeneity. The fact that many different data resources types exist refers to the file-format discussion from Section 1, where as the heterogeneity refers refers to the structural format. The following section explains how these affect the definition of relevance.

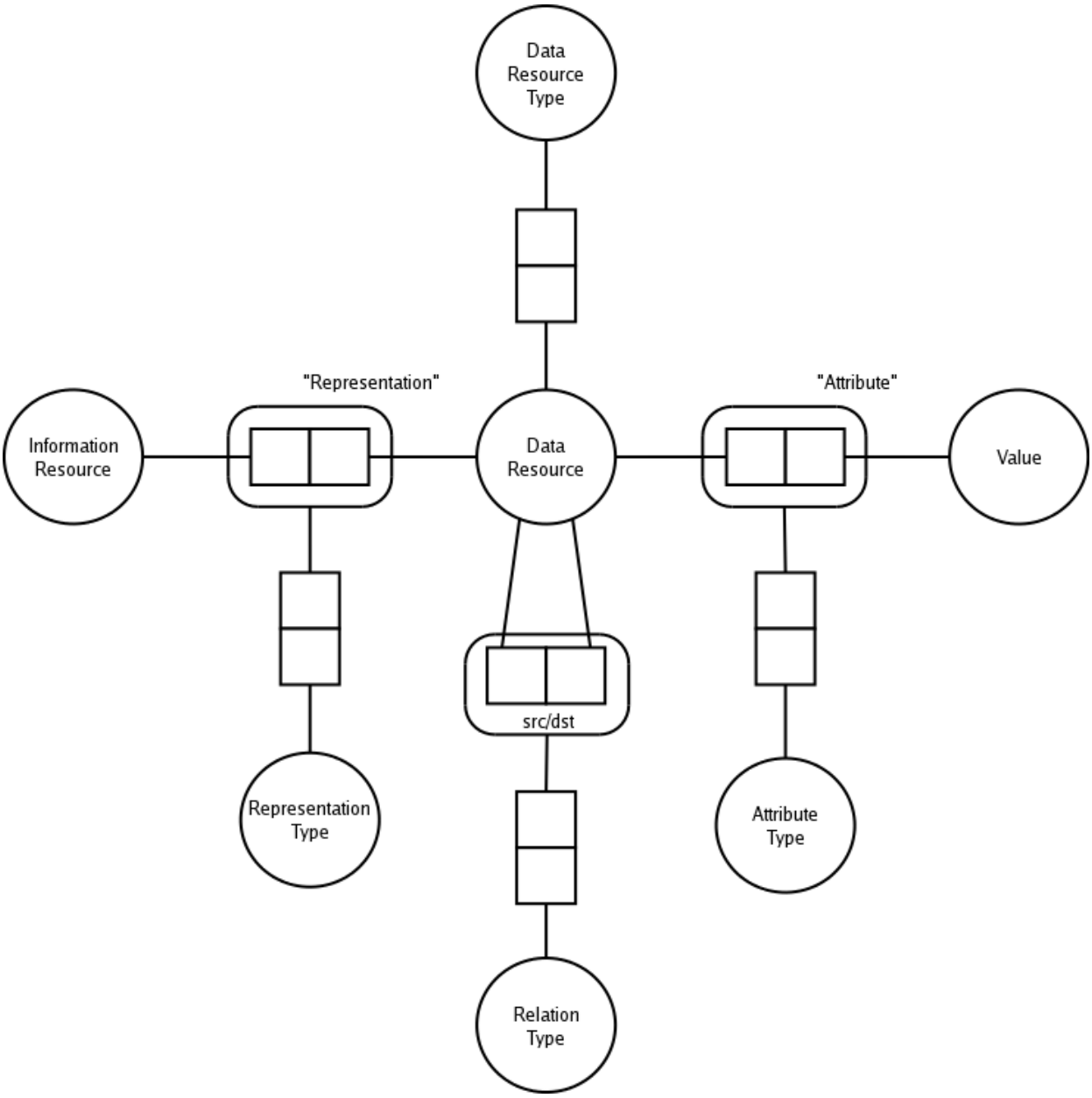


Figure 1. A general model for information supply

4. Relevance

One of the basic functions of any information retrieval (IR) system is *relevance ranking*: the (characterizations of) resources are ranked such that the resources that are ``most relevant'' are listed first, and the ones that are least relevant are listed last. In [11] an overview is given of metrics that are used to determine the relevancy of a Web-document with regard to a query. Furthermore, it is pointed out that relevancy involves more than *topical relevance*; other attributes of resources (such as its quality and price) are important as well.

Apart from topical relevance, which is the `traditional' way of measuring relevance, we define that other constraints must be met as well. Examples of such constraints are its format (as explained in the previous section), but also price, quality, etc. It may very well be that a searcher is willing to pay a certain amount of money in order to get his hands on a high-quality resource! Hence, we define relevance as follows:

Relevance

Resources are relevant with regard to a query if and only if this resource meets all the criteria that a searcher poses on it. These criteria can be formulated in either the query, or the user-profile.

This definition resembles the notion of functional versus non-functional requirements in Software Engineering [23]. It is now well accepted that non functional requirements and functional requirements are equally important to any software engineering project (see e.g. [1, 10] for a discussion on the importance of non functional requirements).

This modified view of relevance has an impact on *precision* and *recall*, for it is `less easy' for a document to be relevant with regard to a query. For example, it may be that a resource must be converted to another format before it is really relevant. In Section 5 we explain how a retrieval system can exploit this new notion of relevance in order to achieve `better retrieval'.

5. Architecture

In the previous sections we explained our notion of formats, profiles and relevance. These notions are essential for the architecture of Vimes, which we will introduce here. The architecture uses many elements that stem from previous research such as brokers, agents, semantic web components and web services.

5.1 Components

The first component is the *profile repository*. This repository stores the characteristics of users so that they are available at all times. This implies that they can/should be used for all the queries, regardless of the search engine that is used. In order to achieve this we make use of an open standard (XML) as outlined briefly in the previous section. An additional advantage of using such an open standard lies in the fact that it will make life easier for developers in the sense that they can more easilly integrate repositories. In the end, users should benefit from this all: they only need to specify their preferences and constraints once in a single profile (over which they have full control) and all search engines that are able to make use of profiles can re-use that single profile[1].

The second component in the Vimes architecture is the *transformation broker*. The basic functionality of this broker is simple: try to transform a given resource into a different form or format. To this end, the broker must have access to a number of transformations, and must be smart enough to be able to "ccompose" other transformations from them. For example, if there is no 1-step fransformation available from the LaTeX-format to the DOC-format, it may be possible to *compose* this transformation from two other transformations such as latex2rtf and rtf2doc.

With this transformation broker we hope to cater for a broader notion of relevance, as explained in the previous section. The broker will be a networked service, encapsulating functionality of all available transformation tools on the network and provide for multiple methods of transport. For example, a request to the transformation broker includes the form desired is PDF and the resource document is a postscript document. This particular conversion can be achieved by a transformation broker on the network that provides the tool *ps2pdf*. For similar reasons as before, we choose open standards for transport such as ftp and http. An additional benefit is that other parties can more easily participate and contribute by submitting transformation routines to the broker.

The broker component will be the main interface for users seeking information. It will interact with the user-profile repositories and search engines on the Web. Essential to our architecture is the broker's ability to interact with not only the well known web search engines (Yahoo, Google, AltaVista, Excite, etc.), but also with such enabling technologies as static agents, mobile agents, web services and services using the Semantic Web or Resource Description Framework (see e.g. [4, 13, 17, 18]). Our broker component will also provide interaction with different forms of user-profile repositories, both local and remote. This will allow interaction with other profile systems on the Web (see e.g. [20] for an agent-based approach along these lines). This leads to the following architectural diagram, with the components in the shaded area making up the system.

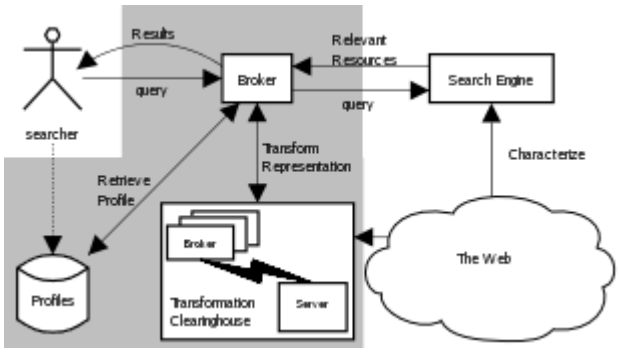


Figure 2. A general model for information supply

Please note that the components will be loosely coupled so that they (especially the profile repository and the transformation broker) can also be accessed by other systems via the Web.

To summarize: the Vimes architecture provides a single interface for users who want to search the Web. It makes use of loosely-coupled components which are available as services over the Web. The main innovation of Vimes is that it uses a broader notion of relevance in the sense that it is capable of doing "more" than just topical relevance.

5.2 Example session

In this section we describe what the retrieval process could look like, based on the architecture as defined in the previous section. The first thing to be done is that the user creates a profile, preferably via an intuitive Web-interface, after which it can securely be stored in the repository. The second step is to browse to the broker, which functions as the main interface for the rest of the process. The user identifies himself (either automatically via e.g. a cookie, or more explicitly via a login-screen) after which the relevant profile is retrieved from the repository.

When the profile is retrieved, the user can enter his query into the system. Two things can happen at this point: either the broker decides to reformulate the query based on the user-profile, or it leaves the query untouched. Subsequently, the query is submitted to one of the search engines. This can be one of the well known web search engines, but others are possible such as an agent, a web service or other external services as described above. Based on the user's profile, the broker may decide to post-process discovered resources. The returned list of discovered resources would then be transformed using the transformation broker. If this is indeed the case, the resources are processed and ranked again before they are presented to the user.

6. Conclusion

In this position paper we introduced a novel retrieval architecture called which is based on a broad notion of relevance and profiles as a means to store user preferences that are (semi) constant.

This broader notion of relevance is derived from a model for information supply (Section 3). The foundation for both this model lies in the *heterogeneity* of information supply: there are many different kinds of resources, several resources may (partially) convey the same information, may have attributes such as price, quality, etc.

The traditional notion of relevance, which ``only" considers topical relevance, can then be extended such that a resource is relevant with regard to a query if and only if all constraints that were posed on it by a searcher are met. These constraints may include things like price and quality, but also structural form and format.

Vimes is intended to be a broker that assists users in querying the Web. There are three important components in this architecture. The *profile repository* stores the profiles of all users in an open format such as XML. There are still some open issues in this area, such as specifying a language for storing the profiles, enforcing that they are stored securely, etc.

The second component is the *transformation broker*, which enables us to perform transformations on resources found on the Web. With these transformations we hope to be able to transform resources into formats that are both convenient and desired by individual users. For example, we can transform a HTML

document into PDF, or generate an abstract of a report that is too long according to a user's profile. We are currently working on a system that performs these transformations by setting up a Conversion Clearinghouse that is web accessible, allowing users to search through our available conversions.

Last but not least, the *broker* in the architecture is the user-interface. It interacts with the two other components, as well as with search engines on the Web. Much work remains to be done in this area. For example, we need to figure out what the interface will look like, which message-standards are going to be used to interface with the other components, etc.

This article intends to provide insight into our novel way of thinking about retrieval. It outlines our proposed architecture without giving a full specification. Finally, we have presented a road-map for our research.

References

1. Barrett, M. L. (2002). Putting non-functional requirements to good use. *The Journal of Computing in Small Colleges*, 18(2):271-277.
2. Barzilay, R. and Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97)*, Madrid, Spain.
3. Berners-Lee, T. (1994). Universal Resource Identifiers in WWW. Technical Report RFC1630, IETF Network Working Group,http://www.ietf.org/rfc/rfc1630.txt. last checked: 13-aug-2002.
4. Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web, a new form of web content that is meaningful to computers will unleash a revolution of new possibilities.*Scientific American*. 284(5):34-43
5. Bray, T., Paoli, J., Sperberg-McQueen, C. M., and Maler, E. (2000). Extensible markup language (XML) 1.0 (second edition). Technical report, World Wide Web Consortium,http://www.w3.org/TR/REC-xml. last checked: 19-may-2003.
6. Bush, V. (1945). As We May Think. *The Atlantic Monthly, 176(1):101-108.*
7. Chen, P.-M. and Kuo, F.-C. (2000). An information retrieval system based on a user-profile. *The Journal of Systems and Software*, 54(1):3-8.
8. Citeseer (1997). *NEC ResearchIndex Citeseer*. http://citeseer.nj.nec.com. Last checked: 19-may-2003.
9. Conklin, J. (1987). Hypertext: An Introduction and Survey. *IEEE Computer*, 20(9):17-41.
10. Cysneiros, L. M. and do Prado Leite, J. C. S. (2002). Non-functional requirements: from elicitation to modelling languages. In *Proceedings of the 24th international conference on Software engineering*, pages 699-700, Orlando, Florida. ACM Press. ISBN: 1-58113-472-X.
11. Dhyani, D., Ng, W. K., and Bhowmick, S. S. (2002). A survey of web metrics. *ACM Computing Surveys (CSUR)*, 34(4):469-503. ISSN:0460-0300.
12. Feng, Hoppenbrouwers, J. (2001). Towards knowledge-based digital libraries. *SIGMOD Record 30*, 1:41-46.
13. Fünfrocken, S. and Mattern, F. (1999). Mobile agents as an architectural concept for internet-based distributed applications - the wasp project approach. In Steinmetz, editor, *Proceedings of the KiVS'99 ("Kommunikation in Verteilten Systemen")*, pages 32-43. Springer-Verlag.
14. Gils, B. v., Proper, H. A., and Bommel, P. v. (2003a). A conceptual model of information supply. *(submitted to) International Journal: Universal Access in the Information Society.*
15. Gils, B. v., Proper, H. A., and Bommel, P. v. (2003b). Towards a general theory for information supply. In Stephanidis, C., editor, *Proceedings of the 10th International Conference on Human-Compu ter Interaction*, pages 720-724, Crete, Greece.
16. Gligor, V. (1996). Characteristics of role-based access control. In *Proceedings of the first ACM Workshop on Role-based access control*, Gaithersburg, Maryland, United States. ACM Press. ISBN: 0-89791-759-6.
17. Google (2003). *Google Web API's*. Google, http://www.google.com/apis. last checked: 16-May-2003.
18. Miller, E., Swick, R., and Brickley, D. (2003). *Resource Description Framework (RDF)*. World Wide Web Consortium, http://www.w3.org/rdf. last checked: 16-May-2003.
19. Myaeng, S. H. and Korfhage, R. R. (1986). Towards an intelligent and personalized retrieval system. In *Proceedings of the ACM SIGART international symposium on Methodologies for intelligent systems*, pages 121-129, Knoxville, Tennessee, United States. ACM Press. ISBN:0-89791-206-3.
20. Pierra, S., Kacan, C., and Probst, W. (2000). An agent-based approach for integrating user profiles into a knowledge management process. *Knowledge-Based Systems*, 13(5):307 - 314.

21. Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill New York, NY.
22. Schabell, E. D. (2002). Resource access in generic information retrieval systems. Master's thesis, Vrije Universiteit, Amsterdam, Netherlands.
23. Sommerville, I. (1989). *Software Engineering*. Addison-Wesley, Reading, Massachusetts.
24. Suryanarayana, L. and Hjelm, J. (2002). Profiles for the situated web. In *Proceedings of the eleventh international conference on the World Wide Web*, pages 200-209, New York, NY, USA. ACM Press. ISBN:1-58113-449-5.

Footnotes

... profile[1]

Open issues that we need to work out still are the management of these profiles: where to store them? how to achieve an acceptable level of security?

... broker[2]

This transformation broker flowed out of the earlier work done on resource access for generic information retrieval [22].

Managing a portal of digital web resources by content syndication

Paul van der Vet (a), Martin Hofmann (b), Theo Huibers (c,a) and Hans Roosendaal (d,a)

(a): Dept. of Computer Science, University of Twente, Enschede
(b): Department of Bioinformatics, Fraunhofer Institute, Algorithms and Scientific Computing Group, Schloss Birlinghoven
(c): KPMG Business Advisory Services, Amstelveen
(d): School of Business, Public Administration and Technology, University of Twente, Enschede

Abstract

As users become more accustomed to continuous Internet access, they will have less patience with the offering of disparate resources. A new generation of portals is being designed that aids users in navigating resource space and in processing the data they retrieved. Such portals offer added value by means of content syndication: the effort to have multiple, federated resources co-operate in order to profit optimally from their synergy. A portal that offers these advantages, however, can only be of lasting value if it is sustainable. We sketch a way to set up and run an organisation that can manage a content syndication portal in a sustainable way.

1. Introduction

The advent of motorways has created a market for one-stop shopping centres. As continuous Internet access becomes more widespread, the distinction in availability between in-house and remote resources loses its significance. The availability of resources thus grows virtually unchecked. However, Internet users can tap the ever-growing plethora of data and knowledge resources available over the web only in principle. Navigation is usually unaided and each resource comes with its own idiosyncratic operating instructions. This situation inhibits the growth of a market for one-stop information services. One-stop information services or portals, as they are often called, aim to provide their users access to information resources in a narrowly defined domain, such as GPCRDB, the portal for information about G-coupled protein receptors. The driving motivation for portals is content syndication: an effort to combine content to provide added value to patrons in making the back office more efficient.

The key success factor for a portal is sustainability. Whatever the portal offers, it should do so with a clear mission, with a clearly defined profile, and with a secured continuity of retrieving it. The current *modus operandi* of many web-based resources and portals is that of self-organisation. It is questionable whether this way sustainability can be assured. In this paper, we want to explore the alternative of an organisation modelled on that of a commercial enterprise for operating a portal in a sustainable way. We present an inventory rather than a complete model and will briefly touch upon a variety of topics to provide a background. The focus is on management and organisation. We will also be dealing exclusively with information produced by the so-called hard sciences like biology and physics.

For the design of one-stop scientific information services, two models stemming from the pre-Web era present themselves: the *repository model* and the *journal model*. They are end points of a continuum rather than models on their own. The repository model is the least ambitious of the two. It views the portal as the WWW analogue of a repository or archive. In this model, the focus is on availability, which in a web environment means navigation in resource space. Like the repository model, the journal model focuses on availability but in addition aims to set a quality standard. Like its source of inspiration, the scientific journal, a journal model portal generally offers less navigation than the repository model and it may cover a narrower field. Because navigation is mandatory when resource space expands, portals that follow the journal model will increasingly add navigation aids, as, indeed, publishers of scientific journals are now providing. The difference between the two models then becomes that of quality assessment. This difference affects the operation of a portal and the possibilities it can offer to its users.

Starting point is that there will be a growing market demand for integration options. Current portals offer access but it is up to the user to further process the information gathered through the portal by means of his

own desktop programs, quite a laborious enterprise. Companies have stepped into this market by offering pipelining systems that enable the user to set up a dataflow between applications with minimal effort. Examples of such tools are the Kensington discovery Environment, TurboWorx, and Pipeline Pilot. As such and similar tools become widespread, data taken from resources are increasingly input in complex calculations, so that it is difficult to assess how errors in the data will affect the result of the calculations. Errors in data are unavoidable, however, even when we disregard data entry errors. The data we are considering stem from experimental science that progresses both by new findings and by corrections of old findings that after a while proved to be erroneous. There are large quality differences between resources. Integration thus depends crucially on resources each having at least a predefined minimum quality. In this sense the repository model does not support integration while the journal model does.

In this paper we further explore the issue of portals that follow the journal model by presenting a design for the organisation that sets up and maintains such a portal, in particular for scientific information. Our more specific example will be a fictitious portal for molecular biology. We think that the design can be ported to other scientific domains like materials science, crystallography, or organic chemistry. It seems plausible that the design can also be ported to non-scientific domains, but we have not considered this issue.

The portal has to fulfil a number of technical and organisational desiderata. Among the technical desiderata are:

- A single entry point giving access to a critical number of resources in a homogeneous way.
- Ability to handle resources of different kinds: databases, knowledge bases, and programs, in a predictable way. Because most resources came into being as the result of a relatively isolated effort, operating instructions were and still are re-invented every time and therefore display a bewildering variety. The portal should hide the variety from the view of the user. [1], [2]
- Intuitive navigation in resource space. One way to ensure this is to present the user with an environment familiar to practitioners of the domain, in which resources are accessed by clicking.
- Ability to handle access fees or subscriptions in a transparent manner. Some resources may be free of charge for everyone, others may be fully commercial, and still others may be free of charge for some user groups but commercial for other groups. Pricing schemes, where applicable, may vary. Users should not be bothered with these details but pay the access fee required to a single party.

Organisational desiderata are:

- The portal must be sustainable. This means that there must be an organisation to secure sufficiently stable sources of income at a level allowing its sustainable operation.
- There must be an organised and accounted form of quality control. The quality of resources varies enormously from being indispensable to being a heuristic aid at best. Primacy must be given to active researchers in the field when matters of content are involved, such as quality criteria.
- The portal must also function as a platform for announcing the availability of new resources so that the user is not obliged to rely on information that has to be gained haphazardly.

We believe that by setting up a portal in this way, resources are used more economically and practising scientists can do their work more efficiently. In the following part of this article, we will further explore the organisation that supports this kind of portal by reviewing the four corners of Leavitt's square, beloved in management science circles: content, process, management, infrastructure. [3]

We will focus on the process and management corners.

2. Content

A portal is of value because it provides access to content that is of interest to a critical number of users. The content fits a profile that can be articulated to such a degree that the portal's existence and mission can be made known to the relevant communities. The content is typically tied to a particular community. In the scientific disciplines we are considering in the present paper, the content is both produced and used by the same community. Of particular relevance to a portal that adheres to the journal model is the presence of shared quality assessment methods in the user community. By contrast, for a virtual theatre portal, the content producers and consumers constitute different communities. This portal gives access to information about shows, concerts, the main performers, while also being a booking office.[4]

A molecular biology portal will give access to gene databanks, protein and pathway databases, literature abstracts and full-text versions of primary journal articles, sequence alignment tools such as BLAST, and more. As tools become mature, access to programs that perform operations on the data such as pathway simulation software and knowledge bases will be added. The portal presents itself to the biologist as a desktop that enables and supports the complicated operations on data required for research in molecular biology. The portal hides from the user whether resources are in-house, maybe even on the same machine, or remote. Biologists will want to be able to store data they obtain in wet labs through the portal, too, so that seamless integration with other resources is ensured from the start.

An issue related to content is the nature of the quality assessment. The assessment typically relates to entire resources. Items kept by resources will generally have been assessed for quality by the content providers of these resources. As a result, the assessment carried out by the portal should be an assessment of the primary quality assessment process carried out by the content provider. Scientific communities are quite familiar with quality assessments and the conclusions that can be drawn from them. The situation is different, however, in cases where the public is given access to resources. Consider, for example, a hospital that wants to provide access to selected resources for patients and their families. The hospital will obviously not want to warrant the correctness of all items to which it gives access this way. What kind of warrant is implied by the quality assessment procedure of the hospital constitutes a subject for legal and, one may add, moral concern.

3. Process

3.1 Introduction

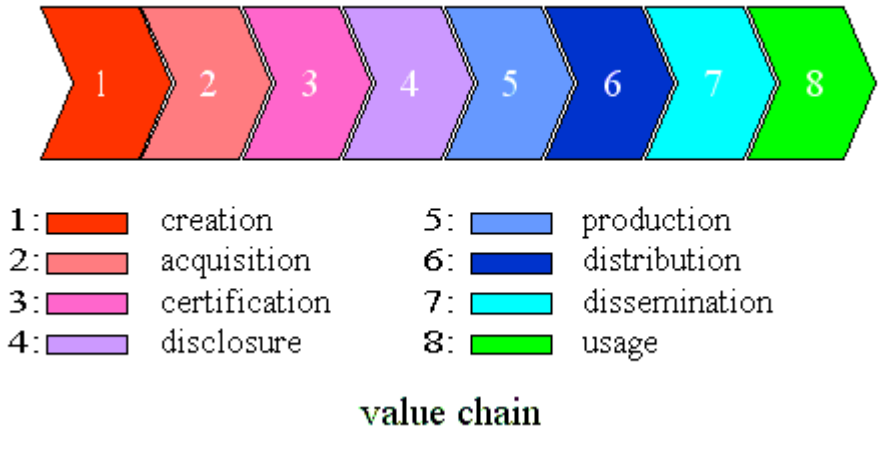
The description of the portal organisation is based on the value chain of scientific information.[5] The value chain consists of steps such that each step adds value to the output of the former step. Each step can be associated with one or more tasks to actually add the value, but the order in which these tasks are performed is only approximately determined by their sequence in the value chain. For example, two values may be added in what is a single process to an institute; or values are added in an iterative process. The entire chain spans a communication process from source to sink.

We use the value chain to define tasks and to allocate them to the various actors that play a role. There are different value chains for different levels of communication; communication may even, at each level, proceed in a different way.

The basic level in the biology example is that of the laboratory, where experiments lead to data that generally are published in peer-reviewed literature. It is possible to discuss the value chain between experiment and refereed paper, but this process is less relevant in the present context and we will regard it as a black box. Increasingly, journals require article authors to deposit their data in a publicly accessible data resource as part of the publishing process. The value chain of a data resource is highly relevant here and we will discuss it below in some detail.

3.2 Value chain of a data resource

The value chain of a data resource can be schematised as in the picture below:



value chain

We will structure the discussion by means of an example that features a fictitious database called E-Base of enzyme properties like chemical structure, 3D shape, genetic origins, and the like. The source of the communication channel is called creation. In the example, it is a black-boxed summary of the laboratory-level processes that lead to publication of enzyme properties in the literature. The acquisition step collects this information from the literature. The certification step subsequently assesses the quality of the data thus gained. In the field of biological databases, this process is often called curation. Note that if E-Base would follow the repository rather than the journal model, this step would consist of a marginal check, for example to ferret out corrupted data. Adding for example metadata in the disclosure step enriches the data for later retrieval. The production step prepares the data for distribution by storing them in a predetermined way on a carrier. The distribution step comprises the digital distribution of the data, including pricing schemes. The dissemination step ensures that the data are disseminated among the appropriate user groups. The end-usage step, finally, constitutes the sink of this value chain.

The value chain is instrumental in organising the tasks that have to be done in order to bring the contents of E-base to its users because, with the obvious exclusion of the creation and end-usage values, the addition of all other values corresponds to identifiable tasks. The end-usage value constitutes the *raison d'être* of the organisation that maintains E-base. One of the discussion points is who should do what tasks. Currently, it is not uncommon to see that an organisation like the one that maintains E-Base performs every task in-house.

3.3 Value chain of a portal organisation

The key observation that underlies the design of our portal is that precisely the same value chain can be assumed at the meta-level of an entry point that provides access to resources. The units transmitted this time are not data but entire resources. Thus, the creation step refers to the process of creating and maintaining resources available over the Web. From the point of view of the portal, this is a black box. In an acquisition step, the organisation responsible for the portal selects candidate resources for addition to the resources it makes accessible. This involves acquisition of a URL and negotiations about conditions of use such as price. The portal wants to be able to give its users an indication of the quality of the resource or, more generally, their 'value for money', whatever the currency may turn out to be. Quality judgements are produced in a certification step. Obviously, in this case the tasks that correspond to the acquisition and certification steps are closely connected because the quality of a resource is an important factor when the portal organisation determines whether it wants to add access to the resource and, if so, at which price. A review committee consisting of domain experts will provide guidelines on the acquisition of resources and their quality level.

In the disclosure step, the portal organisation adds meta-data such as annotations, cross-references, and navigation aids to the resources in order to prepare for easy access by the end-users. The actual work of adding the annotations is done in the production step. The value of the production step is added in two ways: providing the actual access to the resource (by a hyperlink, by mirroring, or in another way) and by ensuring interoperability of the data stemming from different resources.

Addition of the distribution value again involves two tasks. Physical distribution is implemented by means of known server technology. The other task associated with distribution is that of pricing and marketing. Adding the values to the resources by the portal organisation will inevitably incur costs. Adequate funding has to be found for the portal organisation, either as public funding or direct funding by charging the customers, or a combination thereof. A possible scheme could offer two versions: a minimal version at a low charge or free of charge, provided the funding allows this, and a 'de luxe' version that comes at an additional price. The pricing scheme may involve more modalities, however. The use of some resources will no doubt involve fees. To make matters even more complicated; some users of the portal may already have a subscription to some other resources and do not want to be billed twice. This means that issues of pricing and marketing are an important concern.

The addition of aids for end-user navigation is the main value added by the dissemination step. We believe one attractive option is to allow the user to travel in an environment that portrays the scientific domain. Unlike what is the case in traditional virtual reality, the idea is not to mimick reality as closely as possible. Rather, the visualisations help the user to navigate in resource space by making the required distinctions and showing the important relations in a visual way. Finally, a part of the dissemination value can also be added by a client, such as an institute that wants its own, proprietary data accessed together with other resources through the same interface.

End-usage, finally, is within the scope of the portal organisation insofar as expectations of the kind of end-users and their working practices and needs of course drive the entire design.

4. Management and organisation

Some organisation must run the portal and assume overall responsibility for its proper operation. This organisation should be held accountable for the processes outlined above. This organisation should be able to guarantee its stakeholders sustainable utilisation of the portal and the knowledge available and accessible through the system. A portal federating a number of resources allows a lean organisation. This organisation will be faced with a number of strategic and operational objectives.

There are two main strategic tasks to be performed. A most crucial task is to represent the full international community of users and creators of knowledge sources in the project. This is the representation task. This task can best be fulfilled at two levels. At the highest organisational level there is a senior international representation of the entire community. At the operational level, we envisage user groups that meet regularly. Furthermore, the organisation should be able to develop and implement a clear strategy based on the above meta-level value chain for the portal. This is the executive task. The executive task comprises overall responsibility in managing the portal and laying down and deciding on the overall strategic framework for the tasks.

The portal organisation should be able able to achieve the following strategic and operational objectives:

1. Representing the full international community of users and creators of knowledge sources in the fields of interest for the project. This task should be fulfilled at two levels. At the highest organisational level there is a Supervisory Board (SB) that consists of a senior international representation of the entire community. At the operational level, we envisage user groups that meet regularly.
2. Being able to further develop and implement a clear strategy based on the above meta-level value chain for a federated knowledge ensemble for the project. This is the executive task that is entrusted to a small Executive Board (EB). The EB is hired and fired by the SB. The EB manages the consortium and lays down and decides on the overall strategic framework for the tasks mentioned here. To give an example: the EB sets after due consultations the general conditions for certification of resources, while the certification itsef, including a decision on the admission of a resource to the ensemble, is delegated to another body (see objective nr. 7 below).
3. Being able to operate as an international organisation. A task for the EB.
4. Being able to protect the interests of the ensemble such as but not limited to property rights and sustainable continuity of the services in the future. A task for the EB.
5. Being able to provide conditions furthering the sustainability of the participating resources. A task for the EB.
6. Being able to contact and negotiate with suppliers of these resources. This task is best performed by a small acquisition team (that, of course, reports to the EB and acts upon guidelines issued by the EB).
7. Being able to assess the quality of these resources in an independent way. This is the task of an international Review Board (RB). The RB is the ultimate authority in the organisation to approve of the admission of resources into the ensemble and is composed of international experts in the field. The RB is appointed by the EB and performs its task on the basis of a set of formal certification rules as laid down by the EB. It needs to be seen if the RB should possibly be divided into divisional RB's (DRB) to represent a finer granularity of the different subject fields.
8. Being able to warrant the intellectual integrity of these resources. A task for the EB.
9. Being able to market and sell the ensemble under conditions to be agreed. This is the task of a marketing and sales team reporting to the EB and performing its task on the basis of a set of marketing and sales rules as laid down by the EB.
10. Being able to ensure optimal interoperability between the participating resources, with consequences for the interaction between creators and participating resources, and between users and resources. This is the task of an international Standards Committee (SC). The SC is the ultimate authority in the consortium to ensure optimal interoperability of the resources present in the ensemble and is composed of international experts in the field. The SC is appointed by the EB and performs its task on the basis of a set of formal standardisation rules as laid down by the EB. The SC sets guidelines in the following areas of production: data exchange/XML, data management issues, ontologies, orthology, and other areas.
11. Being able to materially create the ensemble, and to operate, maintain, update and expand the ensemble. This is the task of the production team (PT), possibly comprising of a number of specialised

divisions, reporting to the EB and performing its task on the basis of a set of production rules as laid down by the EB. The production team or one of its divisions is supported by an set of expert groups to realise implementation that guarantees interoperability and error-free distribution.

12. Being able to give support to the users (international helpdesk). In particular for those users who will make of the ensemble for purposes representing high risks, such as in clinics, complicated research set-ups, in connection to patients, etc. This task requires a helpdesk.

An organisation as sketched above will be able to operate the portal in a sustainable way that may count on adhesion from the majority of users. We are convinced that there is a market to warrant the investments needed to realise the portal.

5. Infrastructure

Realisation of the portal is largely possible with existing technology. The main technical decision is whether to design the system of portal and resources as a data warehouse or as a federated information system. The pros and cons of either solution are well-known and can be briefly summarised here. A data warehouse gives guaranteed access to all resources and can guarantee interoperability. Also, a data warehouse can be shielded from the outside world except during the brief intervals in which new data and/or resources are added. Against this, maintenance of a data warehouse constitutes a huge and, for many scientific user communities, prohibitive effort. For institutes that can afford the expenditure, a data warehouse is probably the best solution. Indeed, large pharmaceutical and agrotechnical companies routinely establish data warehouses for their in-house researchers, if only because this way, confidentiality of the data and findings can be safeguarded.

A federated information system, [6] by contrast, is an open environment. Maintenance of resources is left to the groups that make the resource available. Maintenance costs for the portal comprise the implementation and maintenance of middleware, of the navigation interface, and of the interoperability layer. Against this, a federated information system relies on a complex configuration of often implicit agreements. For example, resource providers are required to operate their resource in a predictable way, meaning, among other things, to have their data available round the clock and to deliver their data in a format of which the syntax may be unique to the resource but is always known and the semantics is agreed.[7] It is one of the tasks of the portal organisation to make the necessary agreements explicit. Navigation and interoperability are aided by making use of existing consistent semantics and adding semantics where needed. For biology, this semantic interoperability is served by the Open Biology Ontologies initiative. Portals are considered by such diverse organisations as E-BioSci, ORIEL, and BioASP.

6. Further outlooks

The portal organisation will quite naturally assume other activities in fulfilling its mission as general clearing house for information in the chosen domain or domains.

A natural extension of its tasks is to commission literature reviews and other compilations of a predefined quality level. These compilations are in turn available as resources, i.e. via the graphical interface. More importantly, they are structured using meta-data standards and other guidelines, and they can be heavily linked to other resources. This kind of reviews then far surpasses more traditional kinds in terms of reader value.

The developed standardisation products can be tools for a more disciplined data management and experiment description or annotation than is customary today. An important task for the portal organisation in the biology domain will be to bring together existing ontologies and ontologies that will have to be developed so as to span the entire range from molecules to populations, over molecular complexes, organelles, cells, tissues, organs, body parts, and organisms.

Somewhat further in the future lays the use of consistent semantics developed by the organisation, such as in biology ontologies. Consistent semantics structure content and therefore are important didactical aids. They can also be used as a scaffold for constructing a knowledge representation of a major part of a scientific paper. Specialised authorware would construct the knowledge representation in a way that is transparent to the author. For readers, a knowledge representation enables personalisation of the article.

7. Concluding remarks

Resources multiply every day. They are hard to find and their operation requires knowledge of ideosyncratic instructions for use. User communities depending on the availability of resources waste time and money in collecting and processing data, quite aside from the real possibility of errors creeping into and propagating throughout the system. The disadvantages of this state of affairs are now becoming apparent to a number of user communities. These communities are actively seeking ways to remedy the situation. Often, however, the remedy takes the form of a "roll your own"-portal that is operated with uncertain future by one group, while another group with different ideas offers a portal with an equally uncertain lifetime but divergent operation. This way, the advantages of content syndication are not fully exploited and the diversity of resources is simply echoed at a higher level of aggregation. In science, user communities can start scholarly journals, so there is no reason why they could not also start an organisation whose purpose it is to establish and operate a portal in a sustainable way. For the examples we have considered the organisation is international and will almost inevitably be world-wide.

Portal organisations have a vital role to play in scientific research. They can fulfill this role if managed properly, by an organisation that ensures sustainability and assigns responsibilities where they belong.

References

1. P.E. van der Vet, "Building web resources for natural scientists", in: *Interactive distributed multimedia systems and telecommunication services (IDMS2000)*, H. Scholten, M.J. van Sinderen (eds.), LNCS 1905, Berlin: Springer, 2000, pp. 205-210.
2. L. Stein, "Creating a bioinformatics nation", *Nature* 417 (2003) 119-120.
3. A.V. Malchanau, P.E. van der Vet, H.E. Roosendaal, "Habitable Interfaces: an approach to federating information resources for scientific communication", *submitted*.
4. H.J. Leavitt, "Applied organisational change in industry: Structural, technological, and humanistic approaches", in: *Handbook of Organisations*, J. March (ed), Chicago: Rand McNally and Co, 1965, pp. 1144-1170.
5. A. Nijholt, J. Hulstijn, A. van Hessen, "Speech and language interactions in a web theatre environment", in: *Proceedings of the ESCA workshop on Interaction Dialogue in Multi-Modal Systems*, P. Dalsgaard, C.-H. Lee, P. Heisterkamp, R. Cole (eds.), Aalborg: ESCA/Center for PersonKummunikation, 1999, pp. 129-132.
6. H.E. Roosendaal, P.A.T.M. Geurts, "Scientific communication and its relevance to research policy", *Scientometrics* 44 (1999) 507-519.
7. P.M.D. Gray, G.J.L. Kemp, "Federated database technology for data integration: lessons from bioinformatics", in: *Electronic collaboration in science*, S.H. Koslow, M.F. Huerta (eds.), Mahwah NJ: Lawrence Erlbaum, 2000, pp. 45-72.
8. P.E. van der Vet, H.E. Roosendaal, P.A.T.M. Geurts, "C2M: configurable chemical middleware", *Comparative and Functional Genomics* 2 (2001) 371-375.