

Semantic Document Processing with xfy technology

Sunao Takafuji

Justsystem Corporation
1-2-3 KitaAoyama, minato-ku
Tokyo, Japan

sunao_takafuji
@justsystem.co.jp

ABSTRACT

In this paper, we describe how xfy technology [1], an XML handling platform, is a pioneer application of the next generation of document processing that has the potential to create a new document paradigm in this new era of semantic computing. The current document processing paradigm focuses on WYSIWYG interfaces to convey ideas through nice-looking documents. However, as attractive as their presentation might be, such documents rarely allow readers to fully appreciate their meaning, since appearance only partially reflects semantic content. We focus on the semantic side of document processing with XML-centric technology that adds value to documents. We see the need for a new document paradigm powerful enough to process documents semantically – even compound documents of arbitrary complexity – and process the information in those documents at any desired granularity. We embodied this new paradigm in the core functionality of xfy.

Categories and Subject Descriptors

H.1.1 [Systems and Information Theory]: *General systems theory, XML, document structurization, reorganization of information*

General Terms

XML, document processing, document management, knowledge management, metadata

Keywords

xfy technology, xfy framework, XML, mental model, meta-information

1. INTRODUCTION

Pervasive use of information technology in the enterprise and over the internet ensures an ever increasing number of documents. The need for efficient communication has never been greater; yet ironically, massive production of documents degrades the quality of communication, and wide distribution complicates management and reuse.

Some software companies develop database systems to manage unstructured documents, modeling them as either indivisible objects or collections of properties stored in predefined schemata. This approach makes it difficult to retrieve information accurately, inhibits reuse, and is inflexible when changes in the

business environment make parts of the system obsolete.

Other companies have concentrated on knowledge management [2] (KM), in an attempt to synchronize KM theory (knowledge sharing and utilization) with the practice of information technology. Effective knowledge management systems encourage document reuse and re-purposing and support discovery of information in existing documents as a rich source for new knowledge creation. While knowledge management technology encompasses information retrieval, document classification, and text mining, current systems make insufficient use of semantic processing.

Alongside such traditional document management techniques, use of XML vocabularies (e.g., UBL [3], xCBL [4], and XBRL [5]) for document and business process management is increasing. MPEG-7 [6], for instance, proposes a standard for the annotation of all multi-media data. As these standards proliferate and help to clarify document structure, we can look forward to more accurate communication and general improvement of business processes.

XML tags can indicate semantics and thereby facilitate semantic processing on computers, as in a question answering retrieval system [7] that will soon be available. This means XML could improve the quality of automatic text processing systems. Current natural language processing technology, developed over the course of several decades, can also benefit from semantic annotation of free text.

The need for specific tools and applications to support each vocabulary has so far limited the proliferation of XML technology. Additionally, current natural language processing technology has not reached a level of maturity sufficient to rival human understanding. We cannot define in advance all of the tag sets necessary for semantic processing.

This paper discusses how xfy avoids current XML processing bottlenecks and fully exploits the latent capabilities of XML to embody the new document processing paradigm described above. In section 2, we consider the multi-layered structure of documents and study the utility of semantic information in a document according to the mental models [8] of both the writer and the reader. In section 3, we propose a framework for dynamic configuration of meta-information using semantic processing techniques. Then, in section 4, we describe the conceptual design of xfy and review one example of an xfy application in light of the issues raised in sections 2 and 3. The paper concludes with a description of future work in section 5.

2. Meta-information in business documents

2.1 Re-thinking document structure

We conceive the multi-layered structure of documents as shown in Figure 1.

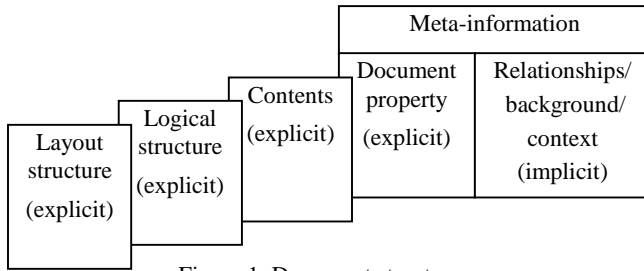


Figure 1. Document structure

The layout structure, or presentation layer, includes such features as the document format and publishing layout. The logical structure is specified by logical elements in languages such as SGML or XML. The meta-structure refers to any additional information conveyed with the document, including its intrinsic semantics.

A compound document embeds other documents within its own logical structure, each of which can be conceived as a monolithic entity in the presentation layer.

Traditional compound document technology integrates layout, processing, and data all together in the document object, so that flexible handling of individual pieces of information is difficult. The meta-structure is also inflexible, typically a fixed set of document properties.

By contrast, XML technology can represent any arbitrary information as an XML document element, and can add meta-information using XML vocabularies for general metadata description (such as RDF [9]).

2.2 Miscommunication through documents

We write documents to convey our ideas to other people, and read documents to share other people's ideas. Only when these ideas are faithfully conveyed can we reach mutual understanding, add value, and benefit from the communication.

After all parties to a written contract agree to the content, value is created through implementation of the contract. A manager in receipt of a report from his/her subordinate can only take appropriate action if the document accurately and completely represents the subordinate's understanding of the issue at hand.

Document standardization and business templates for word processing allow business people to share understanding, rationalize a stream of information, and accelerate the business process. Though these techniques may be effective, they cannot completely eliminate the risk of miscommunication. The gap in understanding is caused by the superficial description of a complex underlying reality; the key to closing this gap is sufficient variety of meta-structure, in particular the semantic structure of document content.

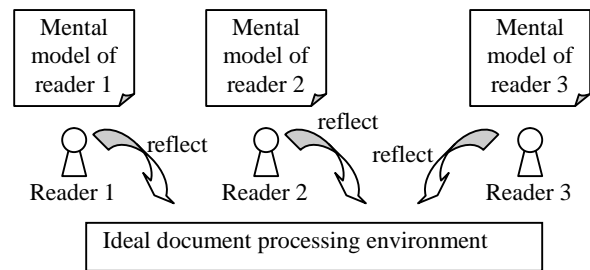
Discord between the mental models of writer and reader arises in part from the diversity of semantic structures in a document. For example, information very important to the writer may be perceived as insignificant by the reader. Or an expert writer may

assume subject matter knowledge or jargon unfamiliar to a general audience.

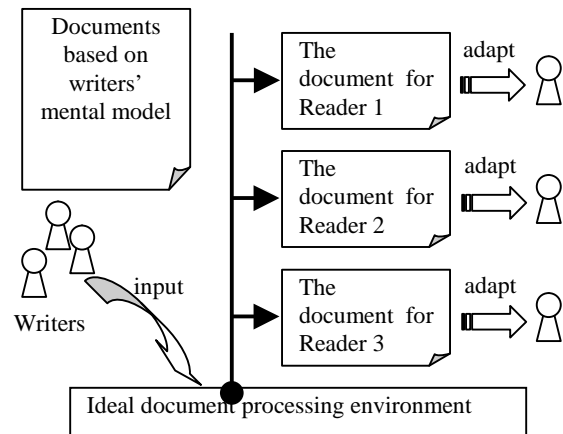
We can see more concrete cases of this miscommunication in offshore outsourcing and the failure of novice engineers. Engineers in the same field living in different countries may not share specification standards. Similarly, a novice engineer may miss critical information in a document that would be readily apparent to a veteran well-versed in the domain.

The mental models of writer and reader may be very different, which often results in a frustrating one-way communication in which the reader struggles to adapt his/her mental model to that of the writer.

An ideal document processing environment would have functionality geared toward synchronizing the different mental models of writers and readers.



2a. Reflection of mental model of readers



2b. Cooperation of mental model

Figure 2. Ideal document processing environment

Figure 2 shows two phases of an ideal document processing environment. The environment receives the readers' mental models as shown in Figure 2a, and generates documents for them based on these mental models – that may be quite different from the writers' – as shown in Figure 2b.

2.3 Implicit links between documents

Electronic documents exist everywhere. From the standpoint of document structure, there are a great number of relational links among them. The hyperlinks in HTML documents form a gigantic "web" that yields great productivity benefits. Even a business

document with no explicit links may nevertheless contain an implicit structure of virtual links.

Factories usually produce many technical documents, such as specification sheets and design diagrams. In the example below, pieces of technical information in one document may be quoted in an order sheet to a manufacturer, and in a presentation sheet for a sales person to a customer. Any monetary values would appear in accounting documents as well, and must agree with the figures in the technical document, order sheet, presentation sheet, and any

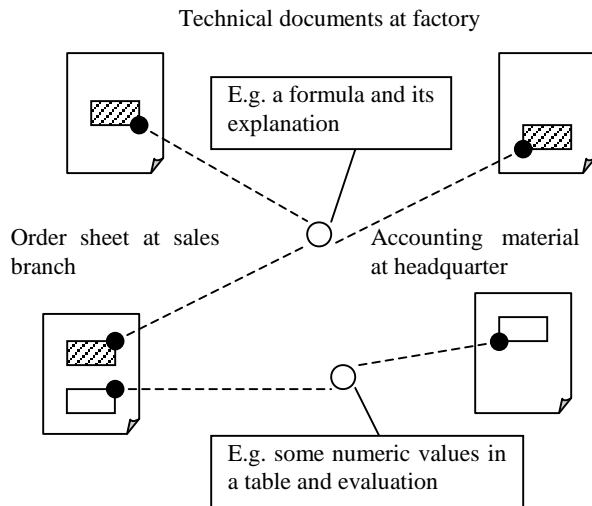


Figure 3. Implicit links between business documents

other related documents. (See Figure 3.)

In such a case, we define an intrinsic hyperlink structure in which each document has a co-relation to or mutual relationship with small semantic pieces of information in “e-space.”

If we ignore these semantic structures within document objects, we have a reference problem: logically identical document parts are scattered throughout the enterprise, their identity unrecognized.

Presently, Japanese manufacturing companies face changes in their business environment, working structure, and demographics. As a result, they have a great need to transfer expert knowledge from one generation to the next, and to reduce the risk of product liability. Although knowledge management is one way to deal with the problem, few even attempt this solution. For one thing, knowledge is scattered everywhere: in e-mail boxes, local hard disks, on network file systems, etc. Also, the word processing and spread sheet applications currently in use are not capable of managing small pieces of information combined with context and background.

Given this state of affairs, it is natural to regard electronic documents as a coherent virtual document space and process them on-demand.

2.4 Synthesis of recognition and maintenance of consistency

Those who communicate through written documents need to improve upon the old fashioned framework. They must find a new way to convey ideas, achieve a synthesis of understanding with

their readers, and gain mutual benefit from the communication. In other words, given the proper framework, writers can build a variety of reader models right into the document generation process, so that the details of tailoring the representation of information to individual readers can be left to the system.

This framework is composed of three elements: a base representation system, a dynamic mapping mechanism, and a presentation system. The base representation system is embodied by single or multiple XML vocabularies. The dynamic mapping mechanism is the system that flexibly composes arbitrary collections of XML tags. The presentation system is the XML document generated by this mapping mechanism.

It is necessary for the system to maintain the integrity of individual pieces of information, even as they become distributed over different documents in e-space. The new document processing framework also includes functionality for consistency management, dependency resolution, and data validation.

3. Semantic document processing with meta-information

3.1 Using meta-information

Section 2 claimed that XML supports reuse through the rearrangement of document components. This is most effective when the document conforms to a well-designed XML tag set or schema.

However, it is impossible to prepare a tag set that fulfills the needs of all document users, and there is always meaningful free text that escapes annotation. Furthermore, tags may impose unnecessary constraints on document configuration.

To cope with this situation, we try to optimize meta-information for semantic content to enable more flexible document reuse.

3.2 Automatic processing of meta-information

While there are advantages to semantic processing, manual annotation comes at great cost. Adding detailed meta-information to text is generally impractical.

Research in information extraction [10, 11] and automatic metadata extraction [12, 13, 14] has contributed to the solution of the annotation problem. In some cases, the technology is practical; the word processor “Ichitaro” [15] and other business intelligence products [16, 17] employ entity extraction technology [18] for the extraction of business entities from documents, and some text mining software [19] uses the dependency parsing technique [20] to discover latent relationships in documents.

In section 2.1, we described the meta-structure of a document. We attach document properties to original documents as needed. Some documents, such as academic papers, have explicit document structure, so meta-information can be extracted rather easily.

While we can seldom predict what kind of tags should be defined for unstructured documents, advanced technology may help automate processing of person name, date/time, place name, and relationship annotation. If the result of such an automation process is insufficient, we can manually correct errors.

We consider such pieces of meta-information components of the writer's mental model for reconfiguration or later reuse. We define certain standardized information sets as "context" (see Figure 4), to be extracted from documents.

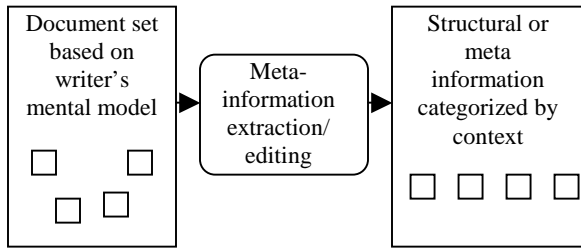


Figure 4. Meta-information added to the original document

We can categorize the types of extracted meta-information from the perspective of semantic processing. Typical meta-information types include fact, theme, and aspect. The program for fact extraction literally extracts facts, such as events with related entities. The program for theme extraction identifies the theme or topic of a text. Aspect extraction extracts the general tone or purpose of a text, such as critical, explanatory, positive/negative, etc.

3.3 How to manage meta-information

Here we consider two methods of managing meta-information. One considers single meta-information objects, and the other expands the model to multiple objects.

The former method may generate a very large DOM and consume a corresponding amount of resources, so we have to design the system carefully to handle small information granularity. The latter method involves managing separated meta-information objects as context sets, and adding or combining some units of context according to the situation.

Given a sufficiently recursive definition of the meta-information set, all meta-information in a document can be represented as a single composite context layer.

A sales manager reading a sales report can recognize certain entities immediately, such as persons, organizations, activities, and causal relationships. These entities and the relationships between them compose the meta-information set. Meta-information that is composed of entities organized from a particular perspective, e.g., a 3-tuple of person, activity, and place, is equivalent to a context layer as shown in Figure 5.

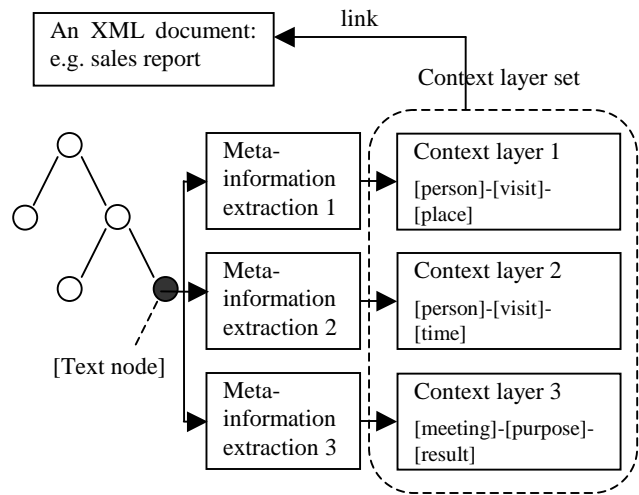


Figure 5. Context layer set

3.4 Recognition using meta-information

Managing a document with its context layer set allows us to configure other documents based on meta-information. We can manage context layer sets by storing them in a repository with links to the original documents. We may use XML-DB as a repository, and access meta-information through an API.

As shown in Figure 6, a reader presents his mental model – the perspective based on his self-context – to the document processing system. This means that he/she edits the range of information, along with its granularity, quantity, and form through a GUI. The document processing system then dynamically generates documents by binding each piece of information to the correct element according to the configuration rules.

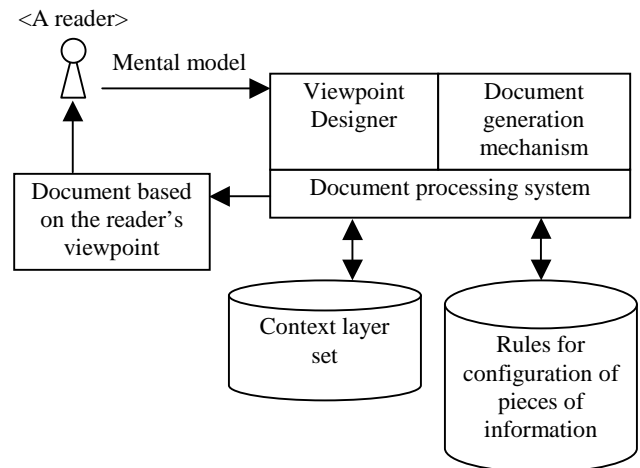


Figure 6. Document generation based on the reader's mental model

This framework enables us to synthesize information at any desired granularity, and to harmonize disparate mental models in a comprehensive manner.

For example, the manager of a business unit can view the document from a high-level perspective to establish the annual business plan using information from past sales reports, while the

manager of human resources can view the “same” document from a resource management perspective, to assign the right salesperson to the right place.

Each manager must explicitly express his/her mental model as shown below.

- a) Mental model of the manager of a business unit
 - The document should be three or fewer pages in length.
 - Annual goals and results of the year for each business section should be described in a table, and illustrated by a graph.
 - The document should include a list of the top three salespersons along with a summary of their activities.
 - It should also include the proposals and risk analysis of each section manager.
- b) Mental model of the manager of human resources
 - The document should be ten or fewer pages in length.
 - It should display a table of current assignments near the top.
 - Next, it should display a summary of each salesperson’s goals and results.
 - Finally, it should represent their sales skills.

Although those mental models are described here in natural language for convenience, they would actually be given as computer-readable terms and numeric values.

In a distributed document environment, the unification of documents may enable us to use them transparently and semantically.

4. xfy framework

4.1 Basic concepts

A basic goal of xfy is the provision of a unified platform for semantic document processing that can handle *any* XML document.

We call this total environment, consonant with the ideas underlying XML, the “xfy framework.” The xfy framework embodies all the functionality we identified in previous sections with the new document paradigm.

That is, xfy is an environment that drastically improves document-based communication between writers and readers. Built upon the flexibility and power of XML technology, it supports new knowledge creation while maintaining consistency of existing distributed information.

4.2 Conceptual design of the xfy framework

We show a conceptual design diagram of the xfy framework in Figure 7.

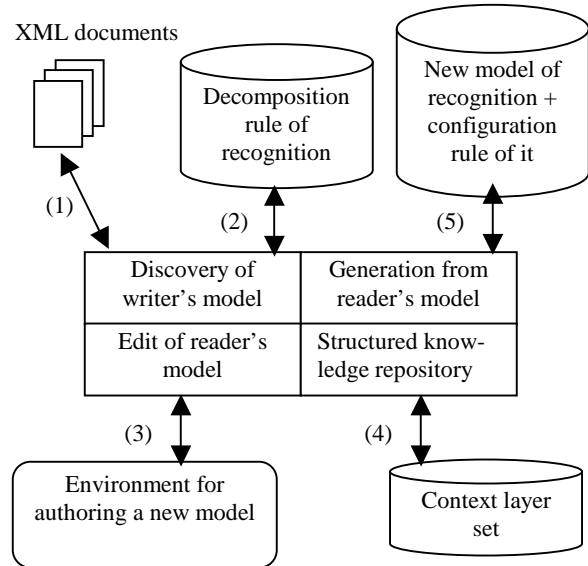


Figure 7. Conceptual model of xfy framework

Four basic functions of xfy are shown at the center of the Figure 7; discovery of the writer’s mental model, maintenance of a structured knowledge repository, edit of the reader’s mental model, and dynamic document generation based on the reader’s model. Numbers in the figure represent interactions between the components.

(1) represents the input to the system: all XML documents. Then xfy discovers the writers’ mental models using extraction rules at a fine granularity through process (2). Extraction rules are represented as XML vocabularies or software modules that extract meta-information. Extracted information is stored in the structured repository as a context layer set through process (4).

The reader may construct his/her own mental model in the xfy framework based on fine-grained semantic and syntactic information through a WYSIWYG interface through process (3). The reader may also create custom plug-in modules suited to particular mental models by applying document configuration rules.

The reader or user of information can create custom views as compound XML documents using extraction and configuration models (5) that conform to particular mental models.

4.3 Features of xfy technology

xfy technology features: 1) extensibility, through support for combining any set of custom XML vocabularies; 2) flexibility, through arbitrary composition of formerly independent DOM trees, 3) user-friendliness, through interactive editing of XML documents.

The Vocabulary Connection (VC) mechanism enables these features. VC is the core technology of xfy that enables the composition of arbitrary XML sub-trees, which may be derived from different namespaces, into a single hybrid DOM tree. This DOM tree is then a single compound document that provides transparent editing access to any node within it.

Since xfy also provides a user-friendly, word processor-like GUI for editing hybrid DOM trees, we can edit XML documents bi-directionally through both GUI and XML text elements.

When a user needs to integrate a new vocabulary into xfy, there are two options. One is to write a plug-in program for the vocabulary in Java; the other is to use Vocabulary Connection Descriptor (VCD) scripting. The former method is for non-XML data, or for advanced annotation functions such as semantic processing. The latter method suffices for most applications, and provides easy integration of XML documents on one comprehensive platform – without requiring users to have engineering expertise.

4.4 How xfy technology works in the xfy framework

Having separately described xfy technology and the xfy framework, we now show how they work together.

xfy is a framework for processing XML documents. As such it has the capacity to integrate any XML related technology. When a task requires too much special processing to be implemented using traditional XML technology, the xfy framework manages the complexity by importing and using an appropriate software module.

The four basic functions of the xfy framework (discovery of the writer’s model, maintenance of a structured knowledge repository, authoring of the reader’s model, and dynamic generation of reader-centric documents) are implemented as follows.

First, “discovery of the writer’s model” combines the original XML document with the XML vocabulary for discovery rules through a VCD. The resulting XML document invokes the processing module (metadata extractor) through the VCD mechanism, and extracts meta-information from the target text nodes. The extracted meta-information is in turn represented as an XML document. If necessary, we can visually edit meta-information through xfy’s GUI.

Meta-information is stored in a knowledge repository managed by the xfy framework (the “structured knowledge repository”).

While authors edit documents, the readers can also edit the presentation form through xfy’s authoring function. “Presentation of the reader’s model” means authoring the form of presentation and the conditions of use. Instead of editing in isolation, we can share and reuse the other people’s models expressed as XML documents.

The last element, “reader-centric document generation” generates the XML document most suitable for the reader’s mental model. Generation rules are described by a VCD script in the XML document. If the generation process is more complicated, a program module plug-in can be written to implement any functionality required. The xfy framework will invoke it through the VCD as necessary.

4.5 An example of application of xfy framework

This section illustrates xfy framework behavior through a concrete example: an application that generates a personalized software magazine from a selection of general monthly software magazines. We suppose the writer is a magazine editor, and the reader is a software engineer assigned to a new project that

requires J2EE technology. The new engineer especially needs to learn about the lightweight container (LWC).

4.5.1 The mental model gap

In order to attract a broad cross-section of readers in the industry, magazine editors need to mix several kinds of articles on a variety of topics: special features, general interest articles, serials, editorial columns, and advertisements. Individual readers, of course, often have highly specific information needs.

This unavoidable difference in mental models creates a gap that may be very difficult to bridge. It is likely that only a few articles correspond to the focus and interests of a particular reader, who may despair of satisfying a specific information need by leafing through a pile of general purpose magazines.

In the current example, the software engineer only wants articles that deal with the LWC in J2EE, yet may not have the time to mine back issues to meet this specific need.

4.5.2 Model discovery: extracting structure from a magazine

Since each software magazine must have a logical structure, it can be described as an XML document. Consider the sample model shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<MentalModelW>
  <Source>HelloJavaWorld</Source>
  <Article type="special_feature">
    <Author>John</Author>
    <Title>Introduction to Caesar</Title>
    <SubTitle>Advanced J2EE technology</SubTitle>
    <Body>Caesar is a lightweight container for J2EE...</Body>
    <Price>$3.0</Price>
  </Article>
</MentalModelW>
```

When xfy extracts meta-information from the XML document, it will invoke plug-in software modules previously incorporated into xfy. The following sample VCD script (not complete code), provides such a mechanism, and will result in the generation of an XHTML document with the extracted meta-information. This sample is self-documented with XML comments.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Declaration of VCD script -->
<vcd:vcd
  <!-- Here, name spaces are abbreviated for convenience. -->
  version="0.1">
  <!-- Extracted meta-information will be represented as RDF. -->
  <vcd:vocabulary match="rdf:RDF" label="View" call-
  template="root"/>
  <!-- Command definition for an extractor button -->
  <!-- The following command will invoke an extraction program
  against indicated text node of DOM. -->
  <vcd:command name="extractorButton">
```

```

<extractor:extractMetadata
  xmlns:extractor="http://xmlns.xfytec.com/extractors/extractor"
  select="{<!-- The target text node should be described. -->}"/>
</vcd:command>
<!-- Template definition to extract meta-information -->
<vcd:template name="root">
<html>
<body>
<!-- If you click this button, xfy extracts meta-information. -->
<ctrl:trigger command="extractorButton"/>
</body>
</html>
</vcd:template>
</vcd:vcd>

```

We assume that the extraction program outputs result data in RDF format. For instance, the snippet below means that the theme of articleID1 is LWC.

```

<rdf:Description rdf:ID="articleID1" >
  <predicate:theme>LWC</predicate:theme>
</rdf:Description>

```

4.5.3 Structured knowledge repository: storage of a magazine's structured information

xfy will store structured information from the software magazine in an XML-DB managed by xfy. This phase includes the process of adding link information between the original and extracted data.

4.5.4 Edit of the reader's model: authoring of constraints to generate a personalized magazine

We suppose that the mental model of the software engineer can be represented as the following.

- I would like to collect LWC articles on J2EE from "HelloJavaWorld" and "JavaLife" over the past two years, and include them in my magazine.
- I want to see special features first. Serials should be in date order, most recent first.
- I don't want to read more than 20 pages.
- I need a glossary of technical jargon related to the technology.

The following XML document captures these concrete constraints, and is sufficient to generate a personalized software magazine for the software engineer.

```

<?xml version="1.0" encoding="UTF-8"?>
<MentalModelR>
  <Source>HelloJavaWorld</Source>
  <Source>JavaLife</Source>
  <Theme>light weight container</Theme>
  <Theme>J2EE</Theme>
  <Volume type="LT">20p</Volume>

```

```

<Order no="1">special feature article</Order>
<Order no="2">general article</Order>
<Constraint>
  <Filter type="time">
    <FromDate>2003.4.1</FromDate>
    <ToDate>2005.3.31</ToDate>
  </Filter>
  <Filter type="category">
    <Item>explanation</Item>
  </Filter>
</Constraint>
</MentalModelR>

```

This XML document will be edited though an xfy GUI specially designed for this purpose.

4.5.5 Synthesis of recognition: generation of a personalized magazine

xfy can generate the personalized software magazine with the following VCD script sample. (Note that it is not complete code, and corresponds to the XML document described in section 4.5.4.)

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Declaration of VCD script -->
<vcd:vcd
  <!-- Here, name spaces are abbreviated for convenience. -->
  version="0.1">
  <!-- Command definition for synthesizer button -->
  <!-- The following command will invoke the program that scores
  each piece of information which match the condition. -->
  <vcd:command name="synthesizerButton">
    <inst:for-each select="{<!-- target meta-information nodes to
    calculate the score -->}">
      <synthesizer:synthesizeMetadata
        xmlns:synthesizer="http://xmlns.xfytec.com/synthesizer"
        select="{<!-- a target node -->}" return-to="result"/>
    </inst:for-each>
  </vcd:command>
  <!-- Template definition to display the view -->
  <vcd:template name="root">
  <html>
  <body>
  <!-- If you click this button, xfy will selects appropriate pieces of
  information and display them. -->
  <ctrl:trigger command="synthesizerButton"/>
  <!-- Display synthesized recognition -->
  <vcd:for-each select="{<!-- scored pieces of information -->}">
  <!-- The following code selects pieces of information with their
  score, and locates them in ascending order on an XHTML
  document. -->
    <vcd:sort select="{<!-- a scored piece of information -->}" data-
    type="text" order="ascending"/>

```

</vcd:for-each>
</body>
</html>
</vcd:template>
</vcd:vcd>

Thus xfy can harmonize the editors' mental models with that of a specific reader.

5. Conclusion

xfy technology is a new document processing platform that combines intelligent, dynamic compounding with semantic processing, and provides this advanced functionality through a user-friendly interface.

In the future, we intend to focus on the following research issues: 1) What is the appropriate level of granularity? What is the best, most flexible information mapping strategy? 2) How best to integrate and control methods of automatic meta-information extraction? 3) How best to manage the original document, meta-information, and derived document (with semantic and mental model markup)?

6. ACKNOWLEDGMENTS

The authors would like to especially thank Hatsuko Ukigawa, executive vice president of Justsystem Corporation, for her thoughtful advice. We also thank Naoya Arakawa, Naoya Uematsu, Masayuki Hayase, Chie Kawano, Shigeki Hagiwara, Nobuyuki Otomori, and Shingo Hisanaga of Justsystem Corporation for their help in reviewing this paper. In addition, we greatly appreciate Jeffrey Bennett and David Evans of Clairvoyance Corporation for reviewing and helpful comments.

7. REFERENCES

- [1] Justsystem Corporation, 2004. xfy technology-authoring and editing compound XML documents. <http://www.xfytec.com/>
- [2] Nonaka, I. (1991). "The knowledge Creating Company." Harvard Business Review, November-December.
- [3] OASIS, 2004. Universal Business Language 1.0. <http://docs.oasis-open.org/ubl/cd-UBL-1.0/>
- [4] Commerce One, LLC, 2004. <http://www.xcbl.org/>
- [5] XBRL International, 2004. <http://www.xbrl.org/Home>
- [6] ISO/IEC JTC1 SC29/WG11. <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>
- [7] <http://trec.nist.gov/data/qa.html>
- [8] Johnson-Laird, P.N. (1983). Mental models: Toward a cognitive science of language, inference, and consciousness. Cambridge, MA: Harvard University Press.
- [9] W3C, 1999. RDF Vocabulary Description Language 1.0: RDF schema. <http://www.w3.org/RDF/>
- [10] Oren Etzioni et al. Web-Scale Information Extraction in KnowItAll(Preliminary Results). World Wide Web Conference, New York, 2004.
- [11] Hang Cui et al. Unsupervised Learning of Soft Patterns for Generating Definitions from Online News. World Wide Web Conference, New York, 2004.
- [12] H. Alani et al. Automatic Extraction of Knowledge from Web Documents. In Workshop of Human Language Technology for the Semantic Web and Web Services, 2nd International Semantic Web Conference, Sanibel Island, Florida, USA, 2003.
- [13] Y.Matsuo, H. Tomobe, K. Hasida, M. Ishizuka. Mining Social Network of Conference Participants from the Web. In Proceedings of the International Conference on Web Intelligence, 2003.
- [14] Junichiro Mori, Yutaka Matsuo, Mitsuru Ishizuka, Boi Faltings. Keyword Extraction from the Web for FOAF Metadata. In 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web, 2004.
- [15] Justsystem Corporation, 2005. <http://www.ichitaro.com/>
- [16] Verity, Inc, 2005. <http://www.verity.com/products/extractor/>
- [17] Inxight Software, Inc, 2005. <http://www.inxight.com/products/smartdiscovery/>
- [18] P.Mcamee and J. Mayfield. Entity extraction without language-specific resources. In D. Roth and A. van den Bosch, editors, Proc. of CoNLL-2002, 2002.
- [19] Nomura Research Institute, Ltd, 2004. <http://www.trueteller.net/>
- [20] Joakim Nivre and Mario Scholz. Deterministic Dependency Parsing of English Text. In the 20th International Conference on Computational Linguistics, 2004.