DETC2002/CIE-34462

WEB-BASED INNOVATION ALERT SERVICES TO SUPPORT PRODUCT DESIGN EVOLUTION

Satyandra K. Gupta
Department of Mechanical Engineering and
Institute for Systems Research
University of Maryland
College Park, MD 20742
skgupta@eng.umd.edu

Edward Lin
Institute for Systems Research
University of Maryland
College Park, MD 20742
lin@isr.umd.edu

Alexander J. Lo
Department of Computer Science
Rose-Hulman Institute of Technology
Terre Haute, IN 47803
loaj@rose-hulman.edu

Changxin Xu
Department of Mechanical Engineering
University of Maryland
College Park, MD 20742
changxin@glue.umd.edu

ABSTRACT

Technological innovations provide an opportunity to improve product performance and reduce cost. Therefore, design organizations are interested in monitoring technological innovations. A large number of innovations are announced every year. Monitoring them manually is very time consuming. We are developing web-based innovation-alert services that can be used to monitor and communicate information about innovations relevant to a particular product design. In this paper, we discuss the required infrastructure, relevant design issues, and our approach to developing web-based innovation alert services to support product design evolution. We also describe a prototype innovation monitoring service for computer components and an interactive tool to transform semi-structured web contents into semantic representations in XML.

KEYWORDS: product design, design representation, information retrieval, technological innovations, semantic web

1 INTRODUCTION

Each year, innovation providers (e.g., developers of new components or manufacturing technologies) announce thousands of technological innovations (e.g., descriptions of new or improved manufacturing processes, materials, and components). If a design organization can redesign a product to incorporate technological innovations, then this might give substantial benefits to the organization, such as improvements in the functionality and performance of the product, or reductions in its cost [Urba93, Utte94, Ulri95, Ches96].

However, such modifications will also incur substantial costs. To decide whether the benefits outweigh the costs sufficiently to warrant making modifications, the design organization may need to obtain and analyze information about many different innovations. In current practice, there are several significant problems with obtaining and analyzing that information:

- Innovation consumers receive too many innovation announcements to evaluate them adequately. A design engineer may receive several hundred innovation announcements each year, primarily through trade magazines and trade shows (for example, in a quick survey of a year's worth of four engineering magazines, we found more than one hundred new innovations just in the area of electromechanical sensors and actuators). Since several people may need to study each innovation announcement to examine its potential impact, the total amount of time needed for this task over the course of a year may add up to many weeks of effort.

- Evaluating and assimilating innovations may require reasoning about information from semantically heterogeneous sources (e.g., different innovation providers, different design tools, or experts from different disciplines). The ability to share such information is often hindered because of differing concepts, terminology, and assumptions about the world. Often, the loosely defined natural-language definitions associated with this information will be too ambiguous to resolve the differences.

1   Copyright © 2002 by ASME

- Existing search facilities are not adequate. Existing search facilities include general-purpose search engines (Yahoo, Lycos, Infoseek, etc.) and database query engines (at component suppliers' sites). Based on our interactions with designers at several leading design organizations (Northrop Grumman, Lockheed Martin, Raytheon, Hughes Network Systems, etc.), we understand that most designers do not use these facilities, because they find them to be of little use in improving their productivity.

We believe that the above problems can be overcome by developing and combining new techniques for information representation and decision making. In particular, we envision the development of Internet-based innovation-alert services that will be used by scientists, engineers, and organizations to communicate information about innovations relevant to product and process design:

- Organizations will post innovation announcements (technical descriptions of innovations) on their web sites in order to facilitate the use of this information across interdisciplinary and organizational boundaries.

- Organizations will be able to create and deploy programs that will monitor postings about innovations, to search for information that might be relevant or useful.

- To help screen and assimilate the information resulting from these interactions and exchanges, organizations will have sophisticated decision models that capture the information about design decisions and process decisions needed to evaluate what impact an innovation might have on the design.

Watch tools, such as 'price watch' and 'stock watch', are available commercially to help people track price and stock index changes. However, these tools are not useful for tracking innovation announcements because innovation announcements contain much richer information. An announcement generally describes innovation specifications that may have attribute/value pairs and values themselves may be symbols. In addition, different innovation providers may use different terms to describe the same innovation. Sometimes same terms may mean different things in two different innovation announcements. Furthermore, innovation announcements are published in html format on innovation providers' web sites. It requires extra effort to extract relevant innovation announcements from web pages.

In order to facilitate evaluating what impact an innovation might have on a design, decision models should be constructed in order to represent the design options that might potentially be used in design, and the decision-making criteria needed to decide which of those options may be feasible or preferable. From the decision model, design team can identify what changes of design variables may affect product design decisions and what design variables are likely to changed by innovations.

Web-based innovation alert services facilitate the monitoring of innovation announcements with desired changes of design variables. An innovation request specifies the condition or constraints that an innovation may become attractive for incorporation into a design. In order to develop a successful innovation alert service, we need to consider three issues.

1. How to properly represent an innovation monitoring request that contain numerical/symbolic values, constraints, and innovation alternatives?
2. How to extract innovation information from web contents and represent them in proper form?
3. How to determine whether an innovation is worthwhile to re-evaluate the design options? In other words, does an innovation meet the thresholds of design variables?

In this paper, we discuss these issues and describe our approach for constructing innovation alert services. We present a representation scheme based on extended AND/OR trees that incorporate constraints and various types of innovations that might be of interest. We focus on extracting information from semi-structured web documents and transforming the extracted information to AND/OR representation for the convenience of information matching. We also developed an innovation scout agent system for monitoring computer products to demonstrate our approach.

This paper is organized as follows. Section 2 reviews related works on specification representation, wrapper technology, and ontology based approaches for knowledge sharing. Section 3 describes the infrastructure and some design issues that could be encountered in developing innovation alert service systems. Section 4 presents our implementation of a web-based innovation alert service for computer product evolution. Section 5 summarizes our research work and directions for further study.

2 BACKGROUND

2.1 Representations of Product Specification and Design Options

The International Organization of Standardization has been developing standards for the exchange of product data under the STEP effort. The first parts of STEP to reach international status were published in 1994 and include product configuration, and product shape (geometry and topology [ISO93] and form features [ISO94]). These generic resource models have been extended into specific domains that include mechanical design [ISO95c], electrical/electronics connectivity [ISO95a], and design and manufacturing [ISO95b]. A few prototype implementations have been reported that use product standards to exchange and store information across a variety of software applications and business enterprises [Petr92, Broo95, Hard96]. However, to the best of our knowledge, there is no formal representation available for specifying requirements or specifications of a product design.

Many previous approaches to representing and enumerating design options have used parametric formulation. In parametric formulation [Diet00], a set of parameters can be identified for a concept used for realizing the product design. When the set of parameters identified for the concept is assigned valid values, a product design is obtained. The set of parameters changes if a different concept is considered for realizing the product design. However, parametric formulations do not work for the product development process in which different concepts are considered for realizing the product design.

Trichur used AND/OR trees to represent product design alternatives [Tric99]. AND/OR trees provide an abstract model that captures the hierarchical structure of a product design.

Design alternatives are explicitly modeled at each level of the functional decomposition process. Nau et al. described an AND/OR tree application to model manufacturing process templates in an integrated product and process design (IPPS) tool for Microwave Modules [Nau00].

2.2 Wrapper Techniques

With the huge amount of information available online, web information extraction has been a popular research area. Several terminologies, such as web mining, information retrieval (IR) and information extraction (IE), are commonly used in this research area. Kosala and Faloutsos et al. reviewed state of art research in this area [Falo95, Kosa00]. Usually software agents or intelligent agents are used to perform web mining tasks. One vital component of any web-based information agent is a set of wrappers. A wrapper is a piece of software that enables a semi-structured web source (a document is semi-structured if the location of the relevant information can be described based on a concise and formal grammar) to be queried as if it were a database. One of the critical problems in building a wrapper is defining a set of extraction rules that define how to locate information on the web [Knob00]. Muslea developed a hierarchical wrapper induction technique for semi-structured information sources [Musl99]. Knoblock developed a system that is able to learn accurate extraction rules and verify the wrapper to ensure that correct data is extracted [Knob00]. This system can also transfer the results into XML.

For semi-structured web documents, current wrapper techniques are adequate. The only problem is efficiency. A large number of wrappers are required for the large web sources. An automatic or semi-automatic way to generate wrappers is also needed. W4F toolkit is able to help building wrappers that translate HTML pages into XML [Sahu99]. This toolkit provides a visual wizard for user to define web source, extraction rules and test the wrapper. Myllymaki developed another approach that uses reference point (anchor) in the HTML page instead of extraction rule in the case of extracting small amount of information from large sources [Myll01]. His approach can be quickly implemented.

2.3 Ontology-Based Approaches for Knowledge Sharing

In order to share innovation announcements and innovation requests, two organizations must agree to a common set of assumptions that provide a semantic basis for their technical descriptions. The specifications of such common assumptions are known as *domain ontologies* [Grub93a].

The seminal work by Gruber [Grub93a] described the role of ontologies (as annotated logical theories) in supporting knowledge sharing. Since this paper, there have been primarily three approaches to using ontologies for managing heterogeneity:

- The first approach attempts to merge representations and create a common ontology that is shared by the different knowledge sources. The KRAFT project, a consortium of three UK universities in partnership with British Telecom, addressed the problem of knowledge fusion from distributed knowledge bases [Gray97, Shav97]. Both the TOVE project [Grun97a, Grun98] at the University of Toronto and the Enterprise Ontology [Usch97] at the

University of Edinburgh designed integrated ontologies for enterprise modeling that can be used by software applications throughout the enterprise.

- The second approach to managing heterogeneity is to construct an ontology for each knowledge source and then build a network of mediators and facilitators that enable translation among these different ontologies. The Ontolingua project [Grub91] at Stanford provides an architecture for constructing, debugging, and translating ontologies written in a canonical format. Wiederhold proposed a network of mediators for translating among ontologies [Wied92]. Research has begun on several approaches to ontology-based knowledge translation [Grub93b, Baal94, Buva95]. In particular, the work of [Cioc00] presents a more rigorous framework for the problem of semantics-preserving mappings among ontologies.

- The third approach combines aspects of the previous approaches by designing a common ontology that all knowledge sources use as an interlingua. Since this allows the communicating knowledge sources to have their own, local ontologies, this approach is applicable to systems that were built without any prior intent for them to communicate. This approach was taken by initial attempts at standardizing the representation of process information [Catr91, Cain92, Ray92, Lee98]. NIST is also developing an ontology-based standard representation for process specifications in product development [Schl, Schl96, Grun97b].

3 WEB BASED INNOVATION ALERT SERVICES INFRASTRUCTURE

A web-based innovation alert service is a web-enabled application that assists organizations and designers to efficiently monitor innovation announcements posted on the Internet, effectively process innovation announcements that are relevant, and alert users with potential innovations that may benefit organization and designers. In the following sections, we first present general system architecture of web innovation alert services. Then, we discuss important design issues.

3.1 System Architecture

A web based-innovation alert service requires performing three operations. It allows organizations and designers to specify innovation alert requests through web browser. The system manages these requests in data warehouse. It deploys information scouts to search and filter relevant innovation announcements posted on the Internet. Innovations that meet alert conditions are forwarded to requestors. Therefore, a web-based innovation alert service generally contains three essential modules:

- *Web-based Interface*: allows users to enter innovation alert requests. Each request may contain innovation requirements (including innovation alert conditions) and monitoring preferences (monitoring time or frequency, and notification method). This interface program analyzes innovation requests, transforms requests to appropriate representations, and stores them in persistent storage.

3   Copyright © 2002 by ASME

- *Data warehouse*: stores innovation alert requests and user profiles. It also stores retrieved innovation information if necessary

- *Intelligent control center*: is the central part of an innovation alert service. It monitors innovation providers' web contents at the time and frequency specified by requestors, extracts information from web contents, filters out information based on user's innovation alert conditions, and notifies requestors if necessary.
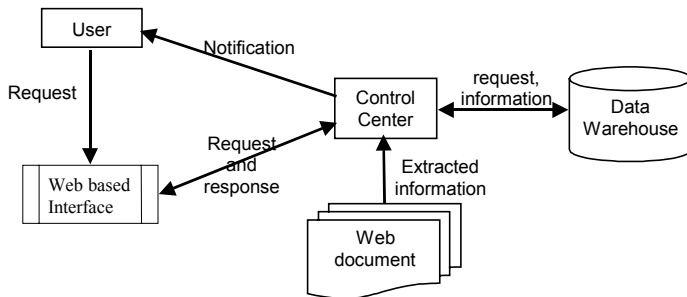

Figure 1: System Architecture

Figure 1 describes the relationship of these modules. In the following sections, we discuss design issues for developing an innovation alert service. First, we describe a generic representation for innovation alert requests. Then, we discuss issues that the control center encounters. In addition, we discuss and present our approach to efficiently extract announcements from a web document. Finally, we describe how the requests are stored in the data warehouse.

3.2 Representation of Innovation Alert Request

In order to facilitate evaluating what impact an innovation may have on a design, we need to represent innovation alternatives that may potentially be used in designs. An innovation can be specified by a finite set of innovation specification items (attributes). Thus, a set of innovation alternatives can be represented as follows:

- Attributes represent product specifications or manufacturing process specifications. Each specification item is represented as an attribute. Note that price is also considered as an attribute because it is also an important decision variable when making product evolution decisions.

- Alternatives represent possible values of an attribute. Values may contain numeric and/or symbolic values. Numeric values may have discrete and continuous values.

- Constraints represent constraints between attributes or constraints applicable to all attributes.

AND/OR tree can be used to represent a set of innovation alternatives. OR nodes represent attributes, whereas AND nodes represent alternatives of an attribute, children of the OR node. However, AND/OR tree doesn't efficiently represent continuous variables. Representing every continuous value as an AND node is not a sound approach. Thus, we have extended basic AND/OR tree and represent continuous values as constraints associated with attribute nodes. For example, an attribute has continuous values ranging from 20 to 30. These values can be represented by "$20 \leq value \leq 30$". This expression can then be further decomposed into the lower and upper limits and an operator pair ("$\leq \& \leq$"). An operator pair ("$< \& <$" or "$< \& \leq$" or "$\leq \& <$" or "$\leq \& \leq$") is specified as a constraint of the attribute while the lower and upper limits are regarded as the children of the attribute node. In addition, complex relationships between attribute nodes can be easily represented in a hierarchical structure using AND and OR nodes. A dependent relationship between attributes is represented as a constraint to the common node of the attributes. An extended AND/OR tree is shown in the Figure 2.
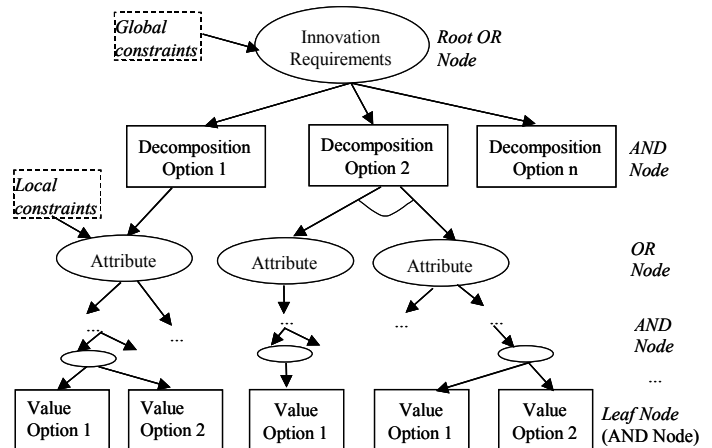

Figure 2: An Example of Extended AND/OR Tree

An innovation request specifies the conditions under which an innovation might have an impact on a design. Thus, a typical innovation request has the following parts:

- Alert conditions specify the nature of the innovation being sought.

- Monitoring preferences include monitoring start time, end time and frequency by which the monitoring job is executed. Monitoring frequency may also be assigned by the system in order to sample the web as few times as possible, without loss of data.

We also use the above extended AND/OR tree to represent innovation alert requests. Alert conditions are specified as constraints of attribute nodes.

3.3 Information Scout Agent Control Center

Control center controls the execution of monitoring, extracting, and filtering of innovation announcements. Once a requestor submits an innovation alert request, the system stores the innovation alert request in persistent storage. The system schedule establishes first monitoring event in the system future event list. Each element in the list is an event that is to be triggered in the future. Each request only has one instance of the request in the future event list. When an event time is up, the system scheduler removes the event from the future event list. Since it's possible that several events are triggered at the same time, the removed events are actually stored in another current events list. An information scout agent is generated for each event in the current event list. Since each event keeps a reference to the corresponding request in the database, the agent can retrieve the request and innovation providers' information from the database and generate a wrapper for each website. After the web document is retrieved, the agent verifies

4   Copyright © 2002 by ASME

each innovation announcement if it satisfies the conditions of the innovation alert request. The system notifies the requestor with matched innovation announcements. A framework of the control center is shown in Figure 3. In the following, we discuss some design issues of the control center.
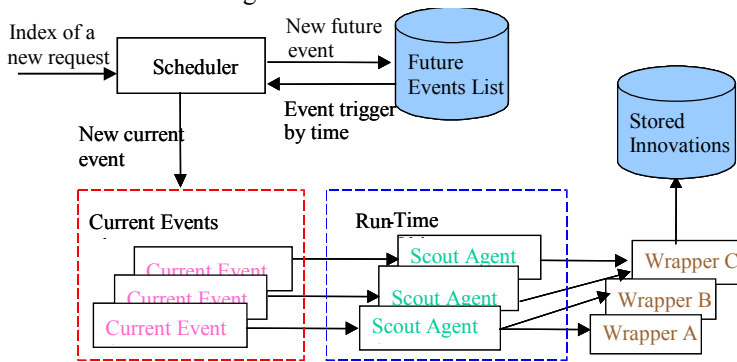

Figure 3: Structure of Control Center

3.3.1 Life Cycle of Monitoring Agent

When an agent is generated for an innovation alert request, an agent of the same request for previous event may still be alive (it didn't finish the job because of internet traffic or some reasons). Different strategies may be adopted to solve this problem:

- Kill the previous agent before generating the new agent. This leads to possible data loss, especially if the information is changing very fast such as stock information. However, for technical innovations, this is not a big concern.

- Allow the current agent to finish its job and generate a new agent when the previous agent completes its task. The next monitoring event for the request in the future event list may still follow the monitoring schedule or follow the delayed schedule.

3.3.2 Efficient Monitoring—Generating Meta-Request for Similar Requests

The control center creates an agent for each request. Since an innovation alert service provider may have millions of requests running everyday, the system may require huge amount of computing power to handle these requests. Efficiency becomes a very important issue.

Different requests may use the same innovation service provider. In other words, same or similar innovation announcements of this innovation service provider may be retrieved several times within a short timeframe. This increases the load on the server. One way to relieve the burden of the server and increase the monitoring efficiency is to generate meta-requests. A meta-request is a general form of requests on some website. It can be viewed as the union of several similar requests (those requests want to monitor the same or similar innovation announcements and at a similar time preference). Whenever a new request is generated regarding the same website, it is first compared to the meta-request of that website. The meta-request is generated when the first user request is generated. The meta-request 'aligns' all the requests towards

that website. In that sense, it is able to update itself when a new request is introduced.

3.3.3 Matching Extracted Information with Requirements

An extracted innovation announcement is an option in the innovation alternatives. We can also use AND/OR tree to represent an extracted innovation. It is an instance of the extended alternative innovation options AND/OR tree in which each OR node has only one child and each attribute node has only one value. Although there are algorithms for comparing two general trees [Kosa89, Dubi90, Peli99], they are not useful for comparing extracted innovation with the extended innovation AND/OR tree.

An innovation announcement satisfies an innovation alert request if it satisfies the following conditions. First, for every attribute node in the extended AND/OR tree, there exists a corresponding attribute node in the extracted innovation AND/OR tree. We don't consider extra attributes in innovation announcements that are not presented in the innovation alert request. Second, if an attribute node of the extended AND/OR tree has value nodes, the corresponding attribute node of the extracted innovation tree must have one of the value nodes. Third, every value node of the attribute nodes in the extracted innovation tree must satisfy the constraints presented in the respective attribute node of the extended innovation AND/OR tree. An algorithm that checks the matching conditions is shown in Figure 4.

- Loop through all options in the requirement tree until match succeeds or loop ends
  - o Select one option
  - o Loop through all attributes in the option until match fails or loop ends
    - Locate one attribute node in the selected option
    - Search the innovation item tree for the same node, if not, match fails
    - Compare the node value with the constraint nodes in the requirement tree

Figure 4: Matching Algorithm of Two Extended AND/OR Trees

3.3.4 Wrapper Generation

Obviously there is no universally applicable extraction rule that works for all web pages. But it is also inadvisable to develop an extraction rule for each page. We observed that most of the semi-structured innovation pages have the following characteristics:

- One web-page consists of several innovation announcements. If there are common attributes shared by all the innovation announcements, these attributes and corresponding values are put on top of all the items.

- Each innovation announcement is listed one by one. Labels or anchors separate each announcement. Table is the most common format.

- In each innovation announcement, interested attributes are mixed but they usually form several groups with rather fixed attributes components. The attribute name or some special anchors, such as the special unit of the attribute, can locate the value of each attribute. Figure 5 shows a typical layout of innovation announcements on a semi-structured web document.

5   Copyright © 2002 by ASME

Attributes	Divider	Attributes	Divider	More Attributes
Attributes	Divider	Attributes	Divider	More Attributes
Attributes	Divider	Attributes	Divider	More Attributes
Attributes	Divider	Attributes	Divider	More Attributes
…	…	…	…	…

Figure 5: Pattern of Semi-Structured Web Document

Thus the extraction rule can be formulated based on the meta-information of the page and predefined tokens to locate the start and end of each innovation announcement. Predefined tokens are those terms or units usually used with the parameters in an innovation specification. Attribute and value need to be searched in each group. However, there are no general patterns to describe how the attributes and values are organized in each group. A reasoning algorithm or procedure is needed to figure out each attribute and value.

Different innovation providers may use different terms to express the same thing. Therefore, mapping the alias of the same token to a unique term is needed. A general extraction procedure is described as follows:

- Step 1: Define tokens and their alias for this category of innovation

- Step 2: Search for metadata information that is the same for all innovation announcements in the same page.

- Step 3: Identify each innovation announcement; locate predefined tokens or their alias in one innovation item; and map them to the attribute-value pairs.

- Step 4: Repeat Step 3 until all innovation announcements are processed.

3.4 Data warehouse

The data warehouse stores future event list, innovation alert requests, and retrieved innovation announcements. Future event list is stored in the data warehouse for the purpose of system recovery. Innovation alert requests and retrieved innovation announcements are stored in XML format. We have developed a DTD file for them. The basic element in the XML file is a 'node' of the AND/OR tree. Children element nodes represent the relationship (parent-children) and constraints. A sample DTD file is shown in Figure 6.

```
<!ELEMENT requirement (node*) >
<!ELEMENT node (label, type, parent?, children?, constraint?) >
<!ELEMENT label (#PCDATA) >
<!ELEMENT type (#PCDATA) >
<!ELEMENT parent (#PCDATA) >
<!ELEMENT children (#PCDATA) >
<!ELEMENT constraint (#PCDATA) >
```

Figure 6: Sample DTD File of Extended AND/OR Tree

4   AN INNOVATION ALERT SERVICE

We have developed an innovation alert service for computer products. One of the reasons that we choose computer products is that computer component innovations happen rather quickly. A tomcat[1] web server hosts our

_____
[1] Tomcat is an open source web server. http://jakarta.apache.org.

innovation alert service. User connects to the server and creates requests using web browser. Servlets are used to verify user's login and process requests. Scheduler, a standalone program, is running continuously in the server. The scheduler is responsible for the monitoring requests. The scheduler keeps an event queue that stores monitoring requests. When an event is triggered, the scheduler retrieves the event information from the queue, gets corresponding request information of the event from an IBM DB2 database, removes the event from the queue, inserts the next event of the request into the queue, and creates an agent for the request. Each event has a unique event ID that is identical to a request ID in the database. The agent generates a wrapper for each innovation provider and starts to retrieve innovation announcements. The agent parses each announcement, converts it to XML format, and determines if it satisfies the request constraints. All matched announcements are stored in the database. An alert is also generated and sent to the user's email box. In the following, we describe our demonstration system.

4.1 XML Representation of Memory Chips Alert Requests

Our innovation alert service monitors computer memory chip innovation announcements. Generally, computer memory chips are classified by a set of attributes such as memory type, packaging type, speed, capacity, voltage, buffered or not, etc. Some attributes have symbolic values such as memory type. Some have discrete numerical values such as voltage and capacity. Though price is not an intrinsic computer memory attribute, it is an important decision variable in product evolution. Therefore, we also regard it as an attribute of computer memory chips. A portion of the computer memory chip classification tree is shown in Figure 7:
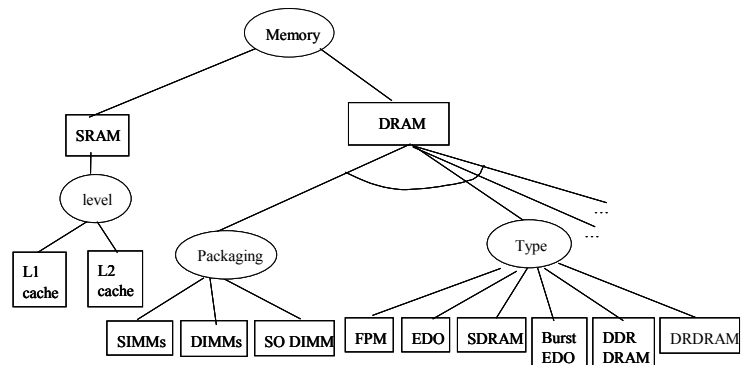

Figure 7: Computer Memory Classification Tree

A computer chip alert request extends the computer memory chips alternative options tree by adding conditions/constraints to attribute nodes. Figure 8 shows an alert request with three constraints.
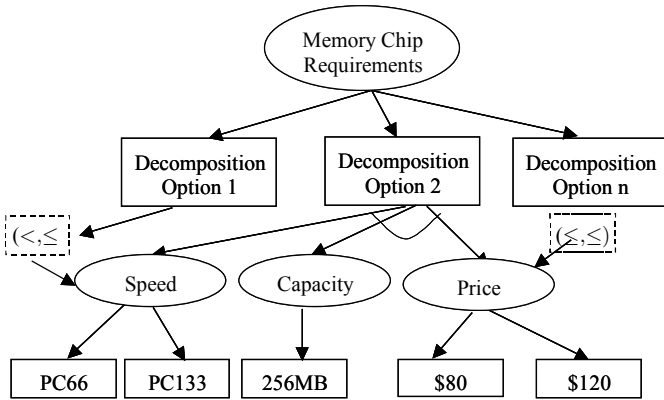
Figure 8: An Example of Alert Request of Computer Memory Chip

In this example, the system alerts the requestor if it finds any computer chips that has a speed between 66MHz and 133MHz, and its capacity is 256Mb, and its price is between $80 and $100. This request is stored in IBM DB2 in XML format. Figure 9 shows a fragment of the stored request.

```
<!DOCTYPE memory SYSTEM " memChipReq.dtd">
<requirement>
<node>
<label>Requirement</label>
<type>OR</type>
<parent>N/A</parent>
<children childrenNames="Option1&Option2&Option3"/>
</node>
<node>
<label>Option1</label>
<type>AND</type>
<parent>Requirement</parent>
<children childrenNames="packaging&capacity&speed&price"/>
</node>
<node>
<label>packaging</label>
<type>OR</type>
<parent>Option1</parent>
<children childrenNames ="DIMMs"/>
</node>
<node>
<label>capacity</label>
<type>OR</type>
<parent>Option1</parent>
<children childrenNames="lowerCap&upperCap"/>
<constraints><&=</constraints>
</node>
<node>
<label> lowerCapacity</label>
<type>AND</type>
<parent>capacity</parent>
<children childrenNames="N/A"/>
</node>
…
</requirement>
```

Figure 9: Fragment of Stored Request

4.2 Innovation Extraction and Matching

Our system generates information scout agents to retrieve computer memory chips announcements from some monitored service providers' web sites. An agent needs a wrapper to retrieve information efficiently. The wrapper requires two configuration files. One file includes some global information of a web document and explains the way data is structured, such as where the information is located in the web page, how the innovation items are organized and whether there is some information to be omitted such as unwanted list of price and so on. We call this 'website configuration'.

The other file describes how to format the output (to fit the DTD file), what information about the product is pertinent (what are it's attributes) and how to identify information as being a certain attribute. We call this 'product configuration'.

In order to improve efficiency and reduce man-made errors, we also developed a wizard for generating wrapper configuration files. This wizard has a step-by-step user interface such that configuration files are constructed correctly. The appendix provides an example of these two configuration files for a fairly structured web site.

Information scout agent also employs some simple pattern recognition algorithms to identify the attribute and values from mixed strings and stores computer chip announcements in XML format as shown in Figure 10.

```
<?xml version="1.0"?>
<!DOCTYPE memory SYSTEM "memoryChip.dtd">
<catalog>
<memory_Chip>
<packaging packagingType="DIMMs"/>
<chip memoryType="SDRAM"/>
<capacity>64MB</capacity>
<speed>PC66</speed>
<price>$45.00 </price>
<errorCorrection CorrectionType="Non-ECC"/>
<buffer bufferType="Unbuffered"/>
<OEMpartnumber>M8*64 PC66</OEMpartnumber>
</memory_Chip>
<memory_Chip>
…
</catalog>
```

Figure 10: Fragment of Computer Memory Chip Announcement

Each extracted innovation announcement is compared to the computer chip alert request using the algorithm introduced in section 3.3.3. An alert will be sent to requestor if there is an announcement that meets the request's constraints. The requestor can login to our alert service web server and review the results.

4.3 User Interface and Examples

A designer visits our system via a web browser. An alert request is specified through a series of Java server pages (JSPs). The first step is to choose a product category and select the monitored component. Figure 11 shows the second step interface that defines innovation alert constraints. The third step is to input monitoring preferences. Figure 12 shows a typical computer chip vendor's website.
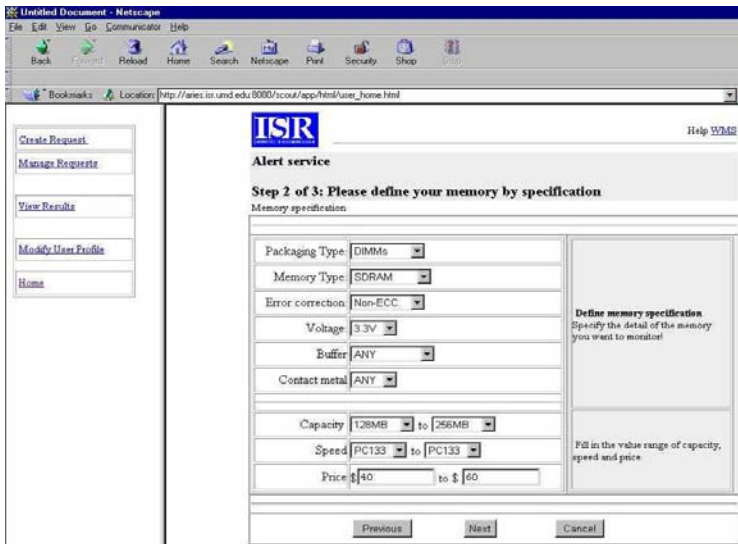
Figure 11: A Step in Definition of Alert Request

Figure 12: Sample of Website Monitored

Most of computer chip vendor' websites list the innovation announcements in a table format. However, location and descriptions of displayed attributes/values vary among these sites. The contents are still semi-structured and our approach can extract announcements correctly. Once innovation announcements that satisfy user's alert request are found, the user gets an alert email. User can also log in to view the result. Figure 13 shows the results from the sample website.

5 CONCLUSIONS

With rapid change in information technology, the Internet has become an electronic communication channel for innovation consumers and providers to share information. In this new era, product design teams require better tools to improve product designs in a timely manner. We envision the development of Internet-based innovation-alert services that will be used to communicate and analyze information about innovations relevant to product design. In this paper, we discuss the required infrastructure, relevant design issues, and our

Figure 13: Sample Result

approach to development of web-based innovation alert services to support product design evolution. Our approach assists designers tracking innovations related to their designs effectively. Three issues are addressed in this paper:

- Representing alternative innovation options, innovation alert requests, and web innovation announcements;
- Generating and controlling information scout agents to retrieve and parse announcements and compare them with requests;
- Representing semi-structured web innovation announcements and generating wrappers.

We also described an innovation alert service for computer components to demonstrate our approach. We believe that our system can be further expended to wider applications of innovation alert services.

Several issues remain to be further studied. First, to facilitate sharing innovation announcements and requests across interdisciplinary and organizational boundaries, we need a robust approach to interpret and translate information into proper semantic representations. Second, in order to improve alert service performance for complicated alert requests, we need a better model to aggregate alert requests. Third, though extracted innovations provide important feedback to the design team, further analysis is still required to determine whether the innovations can be used to make products more profitable.

ACKNOWLEDGMENTS

REFERENCES

[Baal94] J. V. Baalen and R. Fikes. The role of reversible grammars in translating between representation languages. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, 1994.

8   Copyright © 2002 by ASME

[Broo95] S.L. Brooks and R.B. Greenway Jr. Using STEP to integrate design features with manufacturing features. In Proceedings of the *ASME Computers in Engineering Conference*, Boston, MA, 1995.

[Buva95] S. Buvac and R. Fikes. A Declarative Formalization of Knowledge Translation. In *Proceedings of the ACM CIKM: The 4th International Conference on Information and Knowledge Management*, 1995.

[Cain92] W. D. Cain. The use of STEP in an integrated manufacturing environment. *Tech. Report*, Martin Marietta Energy Systems, Inc, 1992.

[Catr91] B. Catron and S. Ray. ALPS: A language for process specification. *Int. J. Computer Integrated Manufacturing*, 4(2):105-113, 1991.

[Ches96] H.W. Chesbrough and D. Teece. Organizing for innovation. *Harvard Business Review*, January-February 1996.

[Cioc00] M. Ciocoiu and D. Nau. Ontology-Based Semantics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference* (KR-2000), April 11-15, 2000, pages 539-546. Breckenridge, CO.

[Diet00] G. E. Dieter. Engineering Design: A Materials and Processing Approach. Irwin McGraw-Hill, 2000

[Dubi90] M.Dubiner, Z.Galil, E.Magen. Faster Tree Pattern Matching. *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, 1990. Page(s): 145 -150 vol.1

[Falo95] C.Faloutsos and D. Oard. *A Survey of Information Retrieval and Filtering Methods. Technical report,* University of Maryland, 1995.

[Gray97] P. M. D. Gray, A. Preece, N. J. Fiddian, W. A. Gray, T. J. M. Bench-Capon, M. J. R. Shave, N. Azarmi, M. Wiegand, M. Ashwell, M. Beer, Z. Cui, B. Diaz, S. M. Embury, K. Hui, A. C. Jones, D. M. Jones, G. J. L. Kemp, E. W. Lawson, K. Lunn, P. Marti, J. Shao and P. R. S. Visser. KRAFT: Knowledge Fusion from Distributed Databases and Knowledge Bases. In The *8th International Conference and Workshop on Database and Expert Systems Applications*, September, 1997. Toulouse, France.

[Grub91] T. R. Gruber. Ontolingua: A Mechanism to Support Portable Ontologies. *Tech. Report KSL 92-66*, Knowledge Systems Laboratory, Stanford University, November, 1991.

[Grub93a] T. R. Gruber. Toward Principles of the Design of Ontologies Used for Knowledge Sharing. In *International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation*, 1993. Padova, Italy. Also available as Report KSL 93-04, Knowledge Systems Laboratory, Stanford University.

[Grub93b] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisitions*, 5(2):199-220, 1993.

[Grun97a] M. Gruninger. Ontologies for Enterprise Engineering. In *Enterprise Engineering and Integration: Building International Consensus*. Springer-Verlag, 1997, pp.

[Grun97b] M. Gruninger, C. Schlenoff and A. Knutilla. Using process requirements as the basis for the creation and evaluation of process ontologies for enterprise modeling. *ACM SIGGROUP Bulletin*, 18(3), August, 1997.

[Grun98] M. Gruninger and M. S. Fox. Enterprise Modeling. *AI Magazine*, 19:109-121, Fall, 1998.

[Hard96] M. Hardwick, D.L. Spooner, T. Rando, and K.C. Morris. Sharing manufacturing information in virtual enterprises. *Communications of the ACM*, 39(2):46--54, February 1996.

[ISO93] International Standards Organization. Integrated resources: Geometric and topological representation. *Technical Report Working Draft*, ISO 10303-42, International Standards Organization, 1993.

[ISO94] International Standards Organization. Integrated resources: From features. *Technical Report Working Draft*, ISO 10303-48, International Standards Organization, 1994.

[ISO95a] International Standards Organization. Electrical/electronics connectivity. *Technical Report Working Draft*, ISO 10303-103, International Standards Organization, 1995.

[ISO95b] International Standards Organization. Electronic PC assembly, design and manufacturing. Technical Report Working Draft, ISO 10303-210, International Standards Organization, 1995.

[ISO95c] International Standards Organization. Mechanical design using boundary/surface/wireframe representations. Technical Report Working Draft, ISO 10303-204/205/206, International Standards Organization, 1995.

[Knob00] C. A. Knoblock, K. Lerman, S. Minton, I. Muslea. Accurately and Reliably Extracting Data from the Web: A Machine Learning Approach. IEEE Data Engineering Bulletin, 2000.

[Kosa89] S.R.Kosaraju. Efficient Tree Pattern Matching. *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, 1989. Page(s): 178 -183

[Kosa00] R. Kosala and H. Blockeel. Web Mining Research: A Survey. *SIGKDD Explorations*, July, 2000.

[Lee98] J. Lee, M. Gruninger, Y. Jin, T. Malone, A. Tate and G. Yost. The PIF Process Interchange Format and Framework. *Knowledge Engineering Review*, 2:1-30, 1998.

[Musl99] I. Muslea, S. Minton, C. A. Knoblock. Hierarchical Wrapper Induction for Semistructured Information Sources. *Autonomous Agents and Multi-Agent Systems*. September, 1999

[Myll01] J. Myllymaki. Effective Web Data Extraction with Standard XML Technologies. In *WWW10,* May 2-5, 2001, Hong Kong. ACM 1-58113-348-0/01/0005.

[Nau00] D. Nau, M. Ball, J. Baras, A. Chowdhury, E. Lin, J. Meyer, R. Rajamani, J. Splain, and V. Trichur. Integrated Product and Process Design of Microwave Modules Using AI Planning and Integer Programming. In *Fourth workshop on "Knowledge Intensive CAD (KIC-4)"*, pp. 186-196, Parma, Italy 2000, IFIP Working Group 5.2.

[Nevi89] J.L. Nevins and D.E. Whitney. *Concurrent design of products and processes: A strategy for the next generation in manufacturing*. McGraw-Hill Publishing Company, 1989.

[Peli99] M. Pelillo, K. Siddlqi, S.W.Zucker. Matching Hierarchical Structures Using Association Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume: 21 Issue: 11, Nov. 1999, Page(s): 1105 -1120

[Petr92] C.J. Petrie. *Enterprise integration modeling: Proceedings of the First International Conference*. The MIT Press, Cambridge, MA, 1992.

[Ray92] S. R. Ray. Using the ALPS process plan model. In *Proc. of ASME Manufacturing International Conference*, 1992.

[Sahu99] A. Sahuguet and F. Azavant. Building light-weight wrappers for legacy web datasources using w4f. In *International Conference on Very Large Databases* (VLDB) (1999), pages 738--741, 1999.

[Schl] C. Schlenoff, A. Knutilla and S. Ray. Process Specification Language Project. <*http://www.mel.nist.gov/psl/*>.

[Schl96] C. Schlenoff, A. Knutilla and S. Ray. Unified process specification language: Requirements for modeling process.

Tech. Report, National Institute of Standards and Technology, Gaithersburg, MD 20899, September, 1996.

[Tric99] V. S. Trichur. *Integer Programming Models for Product Design*. PhD Dissertation, University of Maryland in College Park, 1999.

[Truc87] H.E. Trucks. Designing for economic production. *Society of Manufacturing Engineers*, 1987.

[Ulri95] K.T. Ulrich and S.D. Eppinger. Product design and development. McGraw-Hill, 1995.

[Upto96] D.M. Upton and A.McAfee. The real virtual factory. *Harvard Business Review*, July-August 1996.

[Urba93] G.L. Urban and J.R. Hauser. *Design and marketing of new products*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[Usch97] M. Uschold, M. King, S. Moralee and Y. Zorgios. The Enterprise Ontology. *Knowledge Engineering Review*, 13:47-76, 1997.

[Utte94] J.M. Utterback. Mastering the dynamics of innovation. Harvard Business School Press, Boston, 1994.

[Wied92] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 3(25):38-49, March, 1992.

Appendix: Sample Configuration Files

```
Website configuration file:
==begin file==
==products associated==
[names] memory.prod      [end]
==product==
[look method]    1        comment:   row-by-
row thru table
==table==
[start]
[all words]              true
[table]   12
[row]     2
[end]

[columns]
[number]         5
[1]       ignore
[2]       *multiple*
1         2        3        4        6        7
          8        10
[3]       always 14
[4]       always 5
[5]       ignore
[end]


==end product==
==end of file==
```

```
Product configuration file:
==begin file==
memory.prod
==header==
<?xml version="1.0"?>
<!DOCTYPE memory SYSTEM "memoryChip.dtd">
<catalog>
==ender==
</catalog>
==entry==
<memory_Chip>
</memory_Chip>
==Attributes==
[number]          14
[name]  packaging
[type]    MUST_BE_POSSIBLE_VALUE
[possible values]
SIMMs DIMMs flashcard cache RIMMs
[default value]    DIMMs
[substring values] DIMM = DIMMs
SIMM = SIMMs
RIMM = RIMMs
[attribute output]
<packaging packagingType="        "/>
[end]

[name]  chip memoryType
[type]    MUST_BE_POSSIBLE_VALUE
[possible values] FPM      SDRAMEDO
         DDR_DRAM        DRDRAM
[default value]    SDRAM
[substring values]DDRAM = DDR_DRAM
[attribute output] <chip memoryType="      "/>
[end]
```