

Developing a Knowledge Web for Business Process Redesign

Walt Scacchi* and Andre Valente**

*Information and Computer Science Dept.
University of California at Irvine

**Intelligent Systems Division, USC Information Sciences Institute
University of Southern California

E-mail: wscacchi@ics.uci.edu, valente@isi.edu

(c) Copyright, Walt Scacchi and Andre Valente 1999

Overview

In this paper, we describe our effort at developing and demonstrating a prototype knowledge-based Web environment for modeling, diagnosing and redesigning complex business processes. Our goal was to investigate how a modern knowledge representation system, Loom [MB95], can favorably leverage the development and evolution of a knowledge web that links narrative, informal and formal descriptions of on-line cases on business process redesign. In so doing, we demonstrate concepts, techniques and tools that facilitate the development of a *knowledge web management system* (KWMS) in an application domain of interest to enterprises throughout the world.

We start with a brief background discussion of commercial approaches to knowledge management, as well as to introduce our conception of knowledge webs and knowledge web management systems. We next describe three problems that drive our investigation. The problems are:

- how to acquire global knowledge for how to redesign complex business processes,
- how to develop knowledge representation and operationalization methods that utilize Web resources and technologies, and
- how to facilitate the continuing evolution and improvement of a knowledge web for business process reengineering.

In response, we present a strategy for how to address these problems in a way that combines the capabilities of a large knowledge acquisition team with intelligent systems technologies and the Web. This includes how Web resources and knowledge-based systems were used, the kinds of classification taxonomies that were developed, and how relevant domain and instance-level knowledge was formally represented. In particular, we show how we used Loom and its corresponding Web browser interface, Ontosaurus, in developing a KWMS. Our results demonstrate that an evolving *knowledge web* can be developed and supported by automated reasoning, analysis and classification mechanisms to incrementally integrate new domain facts and to derive implied relations. Our conclusions then follow.

Background

Knowledge management (KM) within modern or virtual enterprises is an emerging area for research investigations, as well as for the development and deployment of commercially available systems [O98]. Commercial systems like Lotus Notes focus on capturing and managing unstructured information that is created and shared by people in enterprises. This information serves as a narrative knowledge base that is to be organized and managed. Other systems from KM tool vendors such as Autonomy Inc. and InXight Inc. provide computational mechanisms that process unstructured information in a Lotus Notes text base or corporate Intranet repository. These mechanisms add value by automatically constructing category (word phrases or co-occurrences) hierarchies and hypertext indices/links that characterize a set of related documents. This is possible in that these mechanisms employ term clustering or classification techniques based on statistical, information theoretic and linguistic analyses of a text database. By their very nature, text analysis techniques such as these provide their value through a shallow syntactic or surface-level processing of a document or text base. However, this value can be enhanced through use of synonym tables, thesauri, domain keyword vocabularies or other weak models to augment the analyses. Nonetheless, these significant capabilities are now commercially available to enterprises that want to begin to manage their text-based knowledge assets.

On the research front, much work in KM remains to be done that builds on results from knowledge engineering, reasoning and problem-solving mechanisms, and other aspects of intelligent systems. Here classic problems in knowledge acquisition, representation and operationalization must be addressed, as they must by commercial KM systems. As noted above, commercial KM system can employ weak models to great success in acquiring and representing (via hypertext links) interrelated text documents that can be operationalized (distributed and navigated) through Web browsers and repository servers. But there is little use of knowledge engineering, reasoning or intelligent systems techniques in commercial KM tools. In contrast, our effort builds from these techniques.

One of the keys to understanding the current emphasis on unstructured information is to realize there is a wide range of levels of formalization in the ways knowledge is represented in KM systems. At the same time, the level of formalization or structure of the knowledge source restricts the mechanisms available to manipulate it, provide services and solve problems. This is presented in Figure 1.

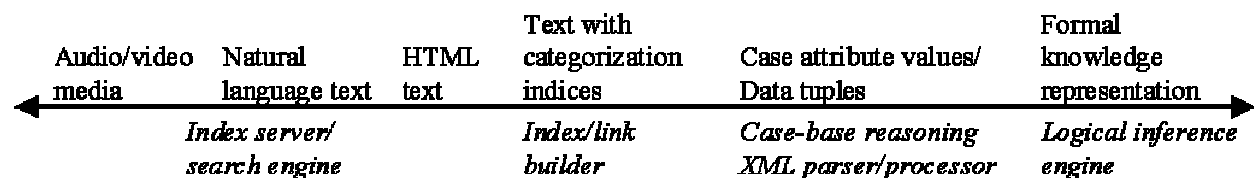


Figure 1. The Knowledge dimension: forms and mechanisms

Along the top is a sort of the notational forms in which knowledge can be captured and

represented. From left to right we increase the formalization and precision of knowledge, while from right to left, we accommodate more informality and ambiguity. Knowledge forms towards the left end are relatively easy for people to create and update, while knowledge forms to the right increasingly demand knowledge engineering, incremental analysis and truth maintenance capability. Along the bottom is a sort of representative knowledge processing tools or mechanisms. From left to right we note an increase in the ability to derive some semantic interpretation from an associated knowledge form: from co-occurrence of specified keywords to logical expressions and deductive assertions. Conversely, from right to left we increase the amount and diversity of information we can search or browse for relevant knowledge. Clearly, most knowledge found on the Web is towards the left end. Further, most of the KM tools concentrate on the left half. In this study, we seek to broaden this scope by organizing, interrelating and managing business process redesign knowledge that *spans* this dimension.

The research in this article is focused on the domain of "business process redesign", or BPR. BPR involves the transformation of enterprise processes, information infrastructure, work situations and surrounding resources into more optimal configurations. BPR is in many ways a precursor of KM in enterprises from historical, organizational and technological perspectives [DP98]. Though much has been written about BPR and hundreds of BPR case studies have appeared (in print and on the Web), there has been little effort at developing an underlying formalization as to what constitutes BPR. For example, an initial effort classify and retrieve relevant cases has been demonstrated using case-based reasoning techniques on a set of eight cases represented as attribute-value tuples [KS96]. Other notable efforts employ knowledge-based capabilities to help redesign specific business processes [N97, ST96]. Nonetheless, there has been progress in developing concepts, techniques and tools for formalizing and engineering enterprise *processes* for business and software development from a knowledge-based perspective [MS96, N97, SM97]. In addition, this progress with enterprise process technology has also led to the integration and use of Web technologies [SN97]. Finally, we have an existing process modeling ontology available to us [MS96]. We thus use our existing enterprise process ontology and domain expertise as the starting point for representing knowledge about BPR. Subsequently, we choose to acquire and represent knowledge of BPR from case studies found on the Web.

By combining formalized knowledge representations for enterprise processes with Web technologies, we make a move towards the development of intelligent support tools for managing BPR knowledge over the Web. Our goal in this study is to acquire and represent a knowledge web for BPR that is operationalized and managed by a prototype knowledge web management system (KWMS). The knowledge web is the target knowledge base we wish to acquire and represent; the environment for operationalizing and managing this knowledge works as a KWMS.

Problems

In order to realize our research goal, we must address three problems. The first problem is *how to acquire global knowledge for how to redesign complex business processes*. Our solution entails a number of activities. We start by using popular search engines and index servers available on the Web. Using keywords such as "business process redesign", "process reengineering" and "case studies", we quickly locate hundreds of links to possible sources of BPR knowledge cases. Of course, given the variability in the precision and recall of these information retrieval services, a

substantial effort is needed to browse, filter and select BPR narratives that can subsequently be coded and classified as useful cases. As a result, we employed a group of 30 graduate students participating in a graduate seminar on BPR to browse, filter, select, code and classify the 200+ BPR cases we found on the Web. The commercial KM tool capabilities noted earlier were not available to us at the time. Thus we could not avoid this problem. Furthermore, even if these tools were available to us, browsing a rough sample of retrieved BPR links reveals that some BPR narratives are short (1-2 printed pages), while others bring up long reports or online books (100-200+ pages). Similarly, some narratives describe *experience reports*, while others tell of *lessons learned* or prescribe *best practices* [O98]. Subsequently, there is a great deal of variability and uncertainty with respect to the kind of results one gets through a surface level analysis of BPR case texts found on the Web, depending on what cases are used for analysis and categorization, and what order you choose to analyze them. As such, our team-based knowledge acquisition effort could produce models of BPR knowledge that might be inconsistent, logically incomplete, or inadequately traceable when integrated. We therefore needed ways and means to enable the construction of a consistent, logically complete and traceable knowledge web for BPR.

The second problem we faced was *how to develop knowledge representation and operationalization methods that utilize Web resources and technologies*. This problem is related to the preceding one. In particular, our graduate student collaborators were organized into six teams who could openly discuss and share their emerging knowledge as to what knowledge they were finding in the Web-based BPR cases. Cases came from all over the world, though for practical reasons, we limited our efforts to those available in English and accessible without fee. All students were trained for what to examine in each case, and each team was assigned to develop a specific BPR case taxonomy. Cases were examined for information that could minimally conform to the ontology of objects, attributes, relations and constraints specified in the *process meta-model* [MS96]. In simple terms, students had to identify the organizational roles of the participants identified in the case, the information processing tasks these participants performed within the business process, the resources used or consumed in the tasks, any off-line or on-line tools or applications, any Internet/Web elements, and what changes occurred to any of these as a result of the process redesign described in the case. Beyond this, the students then had to identify one or more process redesign heuristics that could characterize each selected case. Open discussion and collaboration was again encouraged in identifying the process redesign heuristics, including a lecture discussion of one case study that identifies ten process redesign heuristics [SN97]. Overall, more than 30 process redesign heuristics were identified and classified.

Next, six taxonomies were identified for grouping and organizing access to the BPR cases discovered on the Web. These taxonomies classified and indexed the cases according to:

- *Generic type of industry process*: financial, manufacturing, research and development, marketing, etc.
- *As-is "problems" with existing process*: off-line information processing, workflow delays, lack of information sharing, etc.

- *To-be "solutions" sought for redesigned process:* automate off-line information processing tasks, streamline workflow, use email and databases to share information, etc.
- *Use of intranet, extranet or Web-based process redesign solutions:* build intranet portal for Human Resources information, use HTML form-EDI translators for procurement from Internet-based suppliers, etc.
- *BPR how-to guidelines or lessons learned:* explicit methods or steps for how to understand and model the as-is process, identify process redesign alternatives, involve process users in selecting redesign alternatives, etc.
- *BPR heuristics:* parallelize sequence of mutually exclusive tasks, unfold multi-stage review/approval loops, disintermediate or flatten internal management structures, require and verify all data before processing, etc.

In turn, each of these taxonomies could then be represented as hierarchically nested indices of Web links to the corresponding cases in ways familiar to Web users. Typically, each taxonomy indexes 60-120 cases out of the total set of more than 200. This means that some cases could appear in one taxonomy but not another, while other cases might appear in more than one, and still others might not appear in any of these taxonomies if they were judged to not possess the minimal information needed for characterization and modeling.

The last problem we addressed was *how to facilitate the continuing evolution and improvement of a knowledge web for BPR*. Proposing technically elegant solutions to the preceding two problems that would be costly, take a long time, or produce inconsistent results would defeat our purpose. Similarly, we cannot expect that a no-cost effort requiring little time will produce dramatic results, at least not in a research study. Nonetheless, we wanted to find a means that could incorporate and integrate knowledge acquired from new BPR cases that might later be posted on the Web in a streamlined, evolutionary and incremental manner.

With these problems at hand, we now describe our approach to solving them using a formal knowledge system.

Approach

A key factor in achieving the goals describe in the previous section is the ability to handle knowledge in the more formal end of the spectrum shown in Figure 1. Representing knowledge about a business process using an informal process model, graphic diagram, or in textual form is not always adequate. If we want to automate reasoning with the model to perform some task, we need to use a richer and more structured representation. There are a variety of languages and formalisms where we can represent a business process model. For example, we find business process models expressed using a markup language like XML or even HTML. This provides enough structure to build simple applications that can browse, summarize or search the model contents. However, for complex tasks like evaluating, diagnosing or simulating a process model, however, we may need to perform truly automated reasoning, and for that we need a formal knowledge representation language [SM97].

One of the reasons why formal knowledge representations are not used more frequently is that developing a formalized knowledge base is generally not a casual undertaking. Instead, it is one where circumstances motivate the use or reliance on formal models of domain knowledge and problem solving expertise. In a domain such as Air Campaign Planning [VRMS99], the inherently complexity of the task at hand together with the multiplicity of system applications users employ underscores the need to invest in a developing a formalized knowledge based. Similarly, in a domain like BPR, there is widespread opportunity to practice and reuse known concepts, techniques and tools to realize economic gain, organizational transformation or reinvention. This justifies the investment in developing and engineering a formal knowledge base for BPR interlinked to supporting cases and navigational indices on the Web.

In particular, this points to the need to develop knowledge bases that can be reused easily. Thus, our approach consisted of developing a generic, reusable knowledge base or *ontology* [VRMS99] of business processes and then using it for creating specific knowledge bases of processes in a given industry or company. In turn, the reusable part of these some of these knowledge bases can be transformed into reusable ontologies, in a layered approach.

Once we have the knowledge of business processes formalized in a knowledge representation system we can use the reasoning mechanisms provided by that system to develop applications that analyze a specific process model. In order to test these ideas, we used the ontology of business processes to build a tool to diagnose business process models. The tool reasons with a given process model using a classification mechanisms and lists the problems found. Its knowledge base contains a description of a taxonomy of types of problems in business process models. We then tested this tool by applying it to model cost accounting business processes in a large utility company in California.

Tools for handling formal knowledge

In developing knowledge bases and ontologies for the domain of BPR we used two tools. First, in order to represent BPR knowledge formally and reason with it, we selected the Loom knowledge representation system [MB95]. Loom is a language and environment for constructing ontologies and intelligent applications [VRMS99]. It has been distributed to more than 80 universities and corporations, and is being used in numerous DARPA-sponsored projects in planning, software engineering and intelligent integration of information. Declarative knowledge in Loom consists of *definitions* of concept, relations and instances. A deductive engine called a *classifier* utilizes forward-chaining, semantic unification and object-oriented truth maintenance technologies in order to compile the declarative knowledge into a network designed to efficiently support on-line deductive query processing. By using Loom to re-implement the process meta-model ontology, we can construct formal models of business processes, classification taxonomies and process redesign heuristics. In turn, we can then analyze, query, browse and identify relevant redesign alternatives for processes that have been modeled in Loom and linked over the Web. Finally, we can take advantage of Loom's deductive classifier so that as new BPR cases are acquired, taxonomically classified and formally modeled, our knowledge web can evolve with automated support.

Second, in order to support the visualization of the knowledge bases constructed, we used Ontosaurus [ONTO], a Web browser interface to the Loom system. Ontosaurus is a client-server tool in which a server (written in CommonLisp) with Loom and one or more knowledge bases loaded replies to queries and produces Web pages describing several aspects of the knowledge base. It is also able to provide simple facilities for producing general queries and editing the contents of knowledge bases. Figure 2 shows a typical browser window accessing Ontosaurus. The display consists of three window panels; Toolbar (top), Reference (left side) and Content (right side). The Toolbar panel consists of buttons to perform various operations such as *select domain theory*, *display theory*, *save updates*, etc. The Reference and Content panels are designed to display contents of a selected ontology. Links in both panels display their contents in the Content window. This facilitates exploring various links associated with a word or concept in the Reference window without the need to continuously go back and forth. The bookmark window holds user-selected links for Web pages to Ontosaurus pages, and is managed by the buttons in the bottom of the bookmark window.

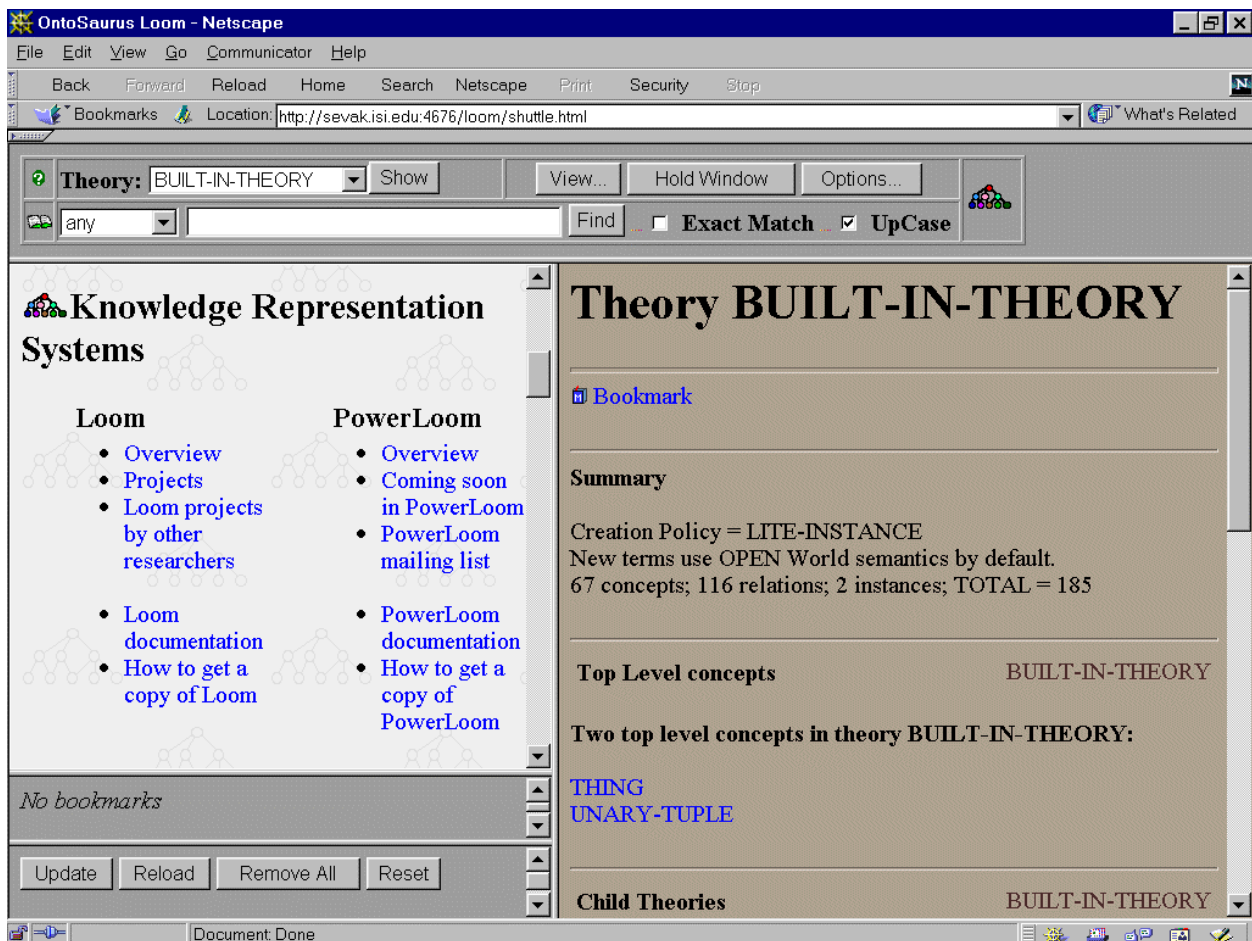


Figure 2. An Ontosaurus browser interface display.

Formally Representing Business Process Models and Process Redesign

We now describe how we built a knowledge-based system to represent and diagnose models of BPR. The system is based on an ontology of business processes expressed in Loom. Loom provides a semantic network knowledge representation framework based on description logics. In Loom it is possible to define *concepts* that have roles or slots to specify its attributes. A key feature of description logic representations is that the semantics of the representation language are very precisely specified. This precise specification makes it possible for the classifier to determine whether one concept *subsumes* another based solely on the formal definitions of the two concepts. The classifier is an important tool for evolving ontologies because it can be used to automatically organize a set of Loom concepts into a classification hierarchy or taxonomy based solely on their definitions. This capability is particularly important as the ontology becomes large, since the classifier will find subsumption relations that people might overlook, as well as modeling errors that could make the knowledge base inconsistent.

Modeling Business Process Knowledge in Loom

The specific process ontology (or meta-model) we employed is centered on the notion of resource [MS96]. There are three basic types of resource: process, agent and simple resource (which is a placeholder for all non-agent resources that are not processes). In order to facilitate communication with a user community that is accustomed to object-oriented modeling, we chose a modeling style in Loom that is closest to object-orientation. For example, information about concepts is stored into roles, similarly to classes and properties in an object-oriented language. The definition of the concept `process` appears in the left frame of Figure 3.

We also made use of Loom's superior expressiveness. For instance, in Loom, roles, slots or properties are first-class objects called *relations*. Relations are defined separately, and may contain not only type restrictions but also cardinality restrictions. It also allows the definition of inverse relations, such as in the example below, where we define the relation `intelligence-collective-role-own-resource`. Note also how we define the domain of the relation as a logical composition over some existing concepts (types).

```
(defrelation intelligence-collective-role-own-resource
:domain (:or intelligent-agent process-role collective-agent)
:range resource
:inverse resource-belong-to-intelligence-collective-role)
```

To create a business process model entails creating a number of instances of the available concepts. This is done in a manner similar to how objects that are instances of classes are created in object-oriented languages. Instances are created in Loom with the command `tellm`. For example, the following is the definition of a business process in a company where we applied the system to support process modeling, analysis and redesign. The definition specifies, for example, that the instance `Produce-work-order` has the attribute `process-require-resource` filled by the value `work-order-preparation-info` (which is itself an instance defined elsewhere).

```
(tellm (:about Produce-work-order
process
(process-require-resource work-order-preparation-info)
```



```
(process-require-resource work-order-data)

(process-provide-resource work-order)
(process-assigned-to-agent-role business-unit)
(process-require-tool-resource accounting-computer-system))
```

Modeling Types of Problems in a Business Process Model

We used the taxonomy of problems in business processes models described above and elsewhere [MS97]. There are three types of problems that arise when modeling business processes. First, consistency problems refer to conflicts in the specification of several properties of a given process. For example, a typical consistency problem is to have a process with the same name as one of its outputs (something that occurs surprisingly often in practice, perhaps because the output is often the most visible characteristic of a process). Second, completeness problems cover incomplete specifications of the process. For instance, a typical completeness problem occurs when we specify a process with no inputs (a "miracle", which can produce outputs with no inputs) or no outputs (a "black hole", where inputs disappear without generating any output). Third, traceability problems are caused by incorrect specification of the origin of the model itself: its author and responsible. Subsequently, a process model that is consistent, complete and traceable can be said to be internally correct.

One of the main reasons that we selected Loom as a representation language was its capability to represent easily and naturally abstract patterns of data that are the very definition of the problems we discussed above. This capability is very handy to produce simple and readable representations of the types of problems with process models. For example, we can define a `black-hole` in plain English as "a process with no outputs". This can be easily described in Loom as a process that provides exactly zero resources:

```
(defconcept black-hole
:is (:and process
(:exactly 0 process-provide-resource)))
```

Many of the simple types of problems can be specified similarly. A more interesting example is the definition of a process with the same name as one of its inputs. Here we use logical expressions to describe conditions that an instance must satisfy in order for a concept to apply. For instance, below we show the specification of `process-and-input-with-same-name` in the ontology. The definition reads (roughly) that a process has input with the same name if and only if it is a process and it satisfies the condition that it requires itself.

```
(defconcept process-and-input-with-same-name
:is (:and process
(:satisfies ?x (process-require-resource ?x ?x))))
```

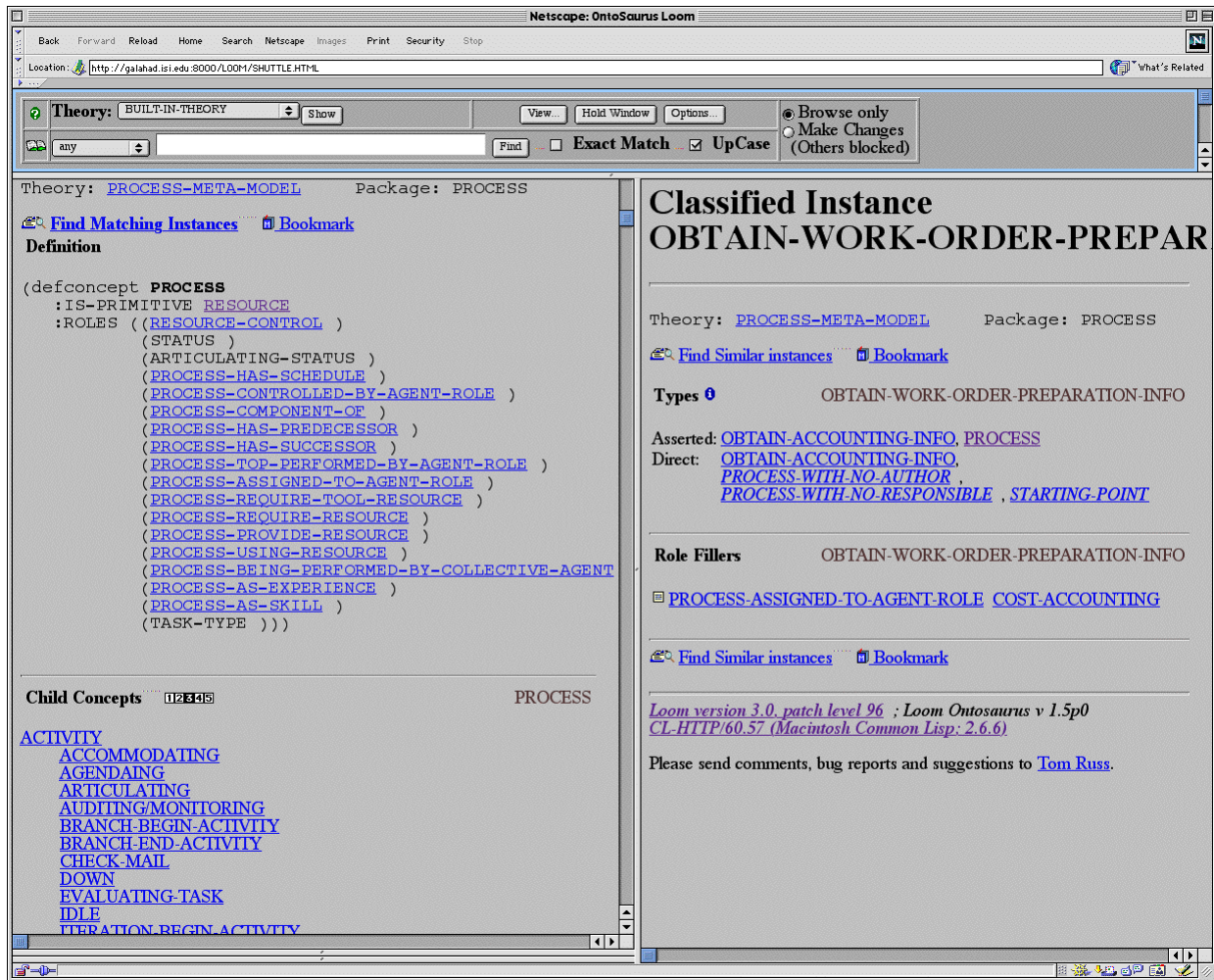


Figure 3. Ontosaurus display with Process concept definition loaded in the Reference window and a process redesign instance in the Contents window

The diagnosis tool

Using the representations discussed above, we built a system that diagnoses business processes. The system operation is very straightforward. The user describes a process model through Ontosaurus for processing by Loom as discussed above. Then the system diagnoses the model provided. One of the advantages of using Loom is that once we define an instance, Loom immediately (and automatically) applies its reasoning engine (the classifier) to find out what concepts apply to that instance. This offers a big advantage, since there is no need to specify an algorithm for the diagnosis process: the diagnosis occurs automatically as we define the model. In addition, the classifier performs truth-maintenance: if we redefine a process to correct a problem found by the system, the classifier will immediately retract the assertion that the problem applies to that process. Thus, we do not need to keep track of the state of the diagnosis as the model changes; instead, the classifier handles the processing activities.

In order to provide a more direct interface to the diagnosis system, we extended the Ontosaurus browser to display two new types of pages. The first displays a description of process in a less Loom-specific way (e.g., for reporting purposes). The second displays a list of all problems found in the current process model we input. The extension was rather simple: all we had to do was to design two additional output page templates and code the appropriate responses in the Loom server. Figure 4 shows a screenshot of the Web page constructed by the server to describe the problems found in a model of a cost accounting process (see below).

Testing the system

We tested the system with a team assigned the task of redesigning the cost accounting processes of a utility company in California (the name is omitted for confidentiality reasons). The team had to acquire knowledge of the processes, formally represent them, and then eventually propose a to-be solution that redesigned the processes. The team interviewed process experts, met with several employees, examined documentation, organizational charts, etc. The first value-added provided by the KWMS was caused even before using the process diagnosis module. By modeling the process in more or less formal terms (i.e. using the process ontology embedded in the system), the team was able to construct a more coherent model. The KWMS helped the BPR team to be more precise and detailed than when they just used graphic diagrams and narrative text. The second value-added was provided by the diagnosis system, since it helped catch errors early in the modeling process. An example appears in Figure 4. In fact, in the beginning the diagnosis module helped the team to learn how to construct the formal representation of the process model by providing feedback in what they should not do. The number of errors generated by entering new processes steadily diminished with time, showing the value of the system.

Discussion and Conclusions

Our investigation demonstrated and prototyped approach to integrate knowledge management and business process reengineering with global resources accessible over the Web. A knowledge web for BPR was acquired, represented and operationalized by a team of collaborating graduate students. We prototyped a Loom-based knowledge web management system that supported the development, use and incremental evolution of a knowledge web grounded in informal BPR case studies found on the Web. This KWMS provides the capability to browse, query, model, and diagnose a knowledge base of formal models of business processes, multiple BPR classification taxonomies, and process redesign heuristics. Subsequently, our team learned and practiced Web-based knowledge management and BPR through participation and contribution. However, the KWMS at this stage lacks the capability to automatically redesign formalized business processes.

Automated redesign is a problem-solving task that represents the next stage of development of our prototype. However, there are BPR situations where automated redesign may not be a suitable goal or outcome. This is in organizational settings where people seek empowerment and involvement in designing and controlling their process structures and workflow. In settings such as these, the ability to access, search/query, select and evaluate possible process redesign alternatives through a KWMS may be more desirable [cf. SN97]. Thus the ultimate purpose of the KWMS we describe may be in *supporting and empowering* BPR participants rather than in

automating BPR.

Overall, the problems identified and addressed by this research are generic and independent of specific business process types. Nonetheless, we believe the approach we pursued is highly reusable and can for the most part be replicated in other settings. Subsequently, we find there are classes of research and practical problems whose solution may span a knowledge dimension that covers informal, semi-structured, and formal representations of knowledge that must then be interlinked across the Web and formal knowledge systems. This work thus represents one such solution.

References

- [DP98] T.H. Davenport and L. Prusak. *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, Boston, MA, 1998.
- [KS96] S. Ku, Y.-H. Suh, and G. Tecuci. Building an Intelligent Business Process Reengineering System: A Case-Based Approach. *Intelligent Systems in Accounting, Finance, and Management*, 5(1):25-39, 1996.
- [L96] D.B. Leake (ed.). *Case-Based Reasoning: Experiences, Lesson and Future Directions*, AAAI/MIT Press, Menlo Park, CA, 1996.
- [MB95] MacGregor, R., and Bates, R. Inside the Loom description classifier. *SIGART Bulletin* 2(3):88-92
- [MS96] P. Mi and W. Scacchi. A Meta-Model for Formulating Knowledge-Based Models of Software Development. *Decision Support Systems*, 17(3):313-330, 1996.
- [N97] M.E. Nissen. Reengineering the RFP Process through Knowledge-Based Systems. *Acquisition Review Quarterly*, 4(1):87-100, 1997.
- [O98] D.E. O'Leary. Enterprise Knowledge Management. *Computer*, 31(3):54-61, 1998.
- [ONTO] Ontosaurus Web Browser home page. <http://www.isi.edu/isd/ontosaurus.html>
- [SM97] W. Scacchi and P. Mi. Process Life Cycle Engineering: A Knowledge-Based Approach and Environment. *Intelligent Systems in Accounting, Finance and Management*, 6:83-107, 1997.
- [SN97] W. Scacchi and J. Noll. Process-Driven Intranets: Life Cycle Support for Process Reengineering. *IEEE Internet Computing*, 1(4):42-49, 1997.
- [ST96] P.G. Selfridge and L.G. Terveen. Knowledge Management Tools for Business Process Support and Reengineering. *Intelligent Systems in Accounting, Finance and Management*, 5:15-24, 1996.
- [VRMS99] A. Valente, T. Russ, R. MacGregor, and W. Swartout. Building and (Re)Using an Ontology for Air Campaign Planning. *IEEE Intelligent Systems*, 14(1):27-36, 1999.

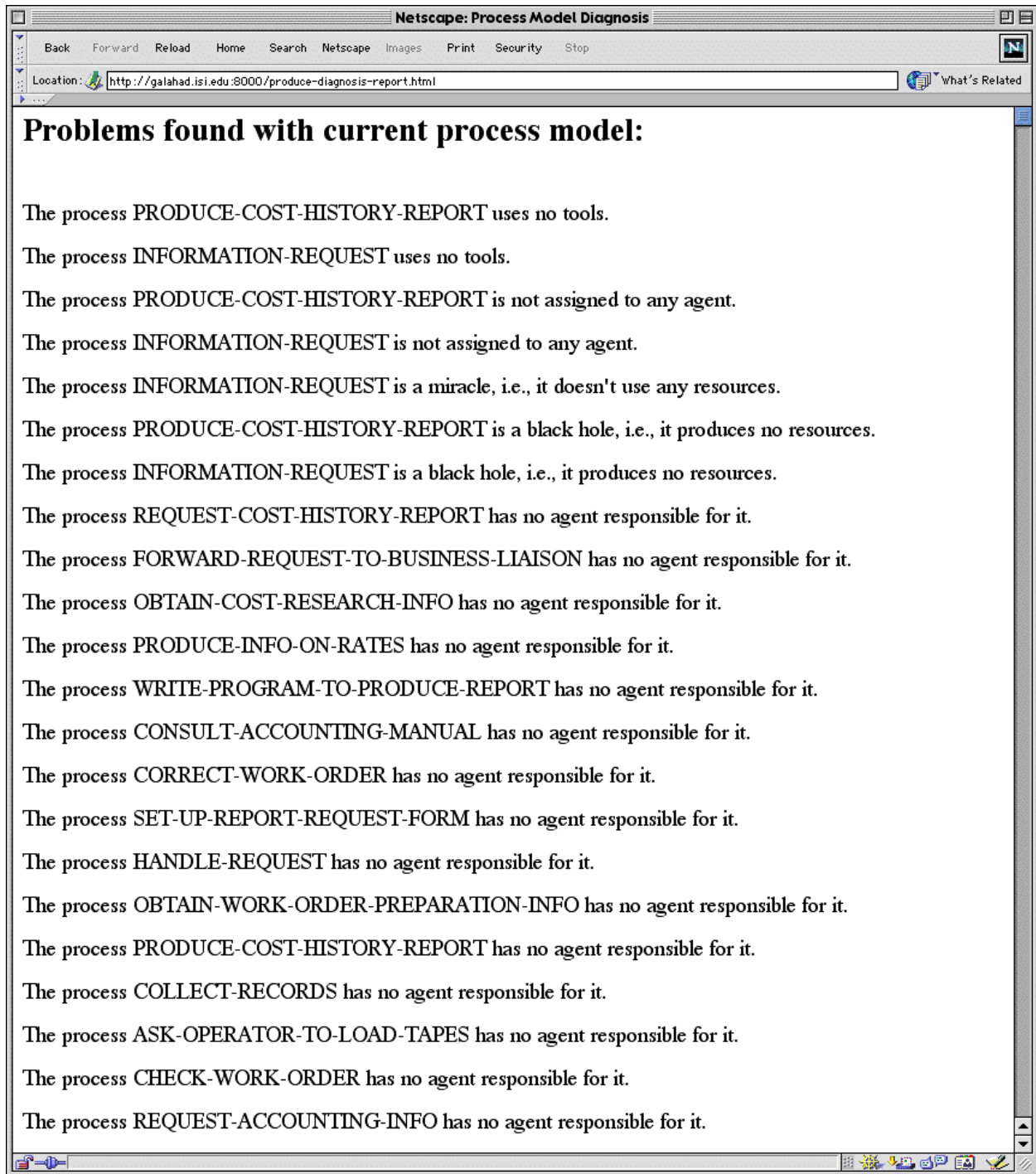


Figure 4. Generated report from Loom diagnosis of a process redesign case