# Toward Intelligent Web-based Redesign Support

**Mark E. Nissen**

Naval Postgraduate School
555 Dyer Road, Code SM/Ni
Monterey, California 93943-5000
MNissen@nps.navy.mil

## Abstract

The reengineering phenomenon continues to be very important in business and management, but the consultant-supported reengineering practice to date has been very expensive and questionable in terms of success. One problem with reengineering at present is the absence of intelligent tools to support process redesign, but recent work on a knowledge-based system called KOPeR has addressed this problem through the design and deployment of the capability for intelligent reengineering "advice." The research described in this paper extends KOPeR to incorporate Web-based connectivity to enable Internet access and use by remote clients. This paper traces through the motivation and background behind intelligent redesign support using rule-based systems and discusses the extension of KOPeR to accommodate Web interface and access. The paper closes with a set of future direction for research along these lines.

## Reengineering Knowledge-Based Systems

With nearly all major corporations—and many other enterprises in the military, government, universities and elsewhere—actively engaged in business process reengineering (BPR) projects (Bashein et al. 1994), the reengineering phenomenon continues to be very important in business and management. However, the reengineering practice to date reflects a questionable record of success and as currently practiced, BPR consulting is a labor-intensive activity, which makes these external consulting services very expensive—particularly given the prevalence of billing rates amounting to several hundreds of dollars per hour. Although reengineering consultants and practitioners are supported by a plethora of tools for *representing* enterprise processes, such tools are devoid of intelligence, yet the reengineering domain has been described as offering good opportunity for AI (Hamscher 1994, Yu et al. 1996), particularly through knowledge-based systems (KBS) technology. Moreover, some researchers are concentrating on automating the activities associated with process *redesign*—which represents the central activity performed by external reengineering consultants—through knowledge-based systems.

Two primary approaches to automated redesign problem solving have been proposed: 1) case-based reasoning (Rock and Yu 1994, Yu and Myopoulos 1996, Yu et al. 1996), and 2) measurement-driven inference (Nissen 1996). Case-based reasoning (CBR) mirrors the kind of human problem solving accomplished by reengineering consultants and a relatively large number of redesign cases now exist for knowledge representation and indexing. However, organizational processes represent very complex systems, each of which has a great many idiosyncrasies and details that can be critical to redesign efficacy, but rarely are such factors expressly recorded in terms that can be incorporated into a casebase; hence a reengineering CBR system would have considerable difficulty adapting previous redesign cases to the needs of novel and dissimilar process instances and problems. This represents a textbook problem with CBR (Rich and Knight 1991). A more serious difficulty with this CBR approach, perhaps, involves the automation of an analytical method that fails more than half the time in practice (e.g., see Hammer and Champy 1993).

Alternatively, a data-driven method can address such idiosyncrasies and details directly (Talebzadeh et al. 1995) and techniques for standardized measurement promote cross-process comparability and inferential robustness. Moreover, some classic problem-solving systems such as MYCIN (Shortliffe 1976) and SOPHIE (Brown et al. 1982) have long been implemented around measurement-driven inference and based on straightforward rule-based reasoning. Specifically, MYCIN uses *counts* of white blood cells to drive inference oriented toward the diagnosis of blood disease and SOPHIE uses electrical *measurements* such as voltage to guide inference oriented toward diagnosing faults in electronic circuits. Both systems perform heuristic classification (Jackson 1990) and suggest that analogous *redesign* problem solving may be feasible.

An analogous KBS to support process redesign has been designed to provide intelligent reengineering "advice" by diagnosing the pathologies or faults associated with *organizational processes* and applying reengineering knowledge to guide the selection of appropriate treatments and repairs (i.e., enabling technologies and redesign transformations) to generate a set of redesign alternatives (Nissen 1997). The background information pertaining to this KOPeR design is presented in the next section.

## KOPeR Background

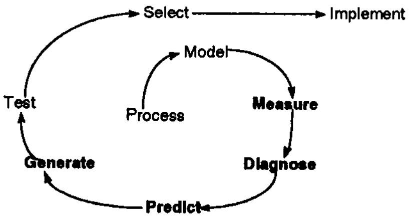

**Figure 1** KOPeR Requirements

KOPeR (pronounced "cope-er") is a KBS for Knowledge-based Organizational Process Redesign that is predicated on measurement-driven inference and designed following the techniques employed to develop the KBS exemplars from above. Figure 1 delineates the principal problem-solving steps required for measurement-driven process redesign, which are used to describe the requirements of KOPeR. From the figure, problem solving begins with the selection of some process in the enterprise for redesign. The redesign steps contemplated for automation through KOPeR are highlighted in bold, which include process measurement and reasoning to diagnose pathologies, predict appropriate technologies and transformations and generate one or more redesign alternatives to improve process performance. For automated problem solving, the process is first represented in terms of a *model* to support measurement, diagnosis, prediction and generation; clearly if one can obtain measurements and reason about the process directly, however, then this representation step can be omitted and the redesign problem solving can be performed (slower and less-reliably) in a manual fashion (i.e., as in current BPR practice).

**Table I** Taxonomy of Process Pathologies

| Pathology Class |
|---|
| Bureaucratic organization |
| "Checking" approach to quality |
| Problematic process structure |
| Inadequate IT infrastructure |
| Fragmented process flows |
| Centralized authority |
| Under-utilized human potential |
| Inhibitive leadership |
| Centralized information |
| Deficient core competency |

The KOPeR design integrates one taxonomy of common process pathologies with a second taxonomy of transformations to redesign the process, and the design is based to a large extent on the two-taxonomy approach employed by Mi (1992) for *general* process diagnosis and repair through an implementation called the Articulator (Mi and Scacchi 1990, 1993). Like the Articulator, both KOPeR taxonomies are organized into classes and subclasses of pathologies/transformations to support abstraction and refinement. The ten major problem types (i.e., classes of pathologies) are presented in Table I and the seven classes of transformations are presented in Table II; clearly these taxonomies are extensible and incorporate only a subset of all potential pathologies and transformations associated with enterprise processes. The reengineering literature provides the source of both taxonomies, as it is replete with instances of suboptimal processes and redesign transformations. The integration and operationalization of these taxonomies is accomplished using a graph-based measurement scheme based on the process-resource ontology of (Mi and Scacchi 1995).

**Table II** Taxonomy of Process Transformations

| Transformation Class |
|---|
| Organizational design (OD) |
| Workflow reconfiguration (WF) |
| Information technology (IT) |
| Human resources (HR) |
| Information availability (IO) |
| Interorganizational alliance (AL) |
| Management and culture (CU) |

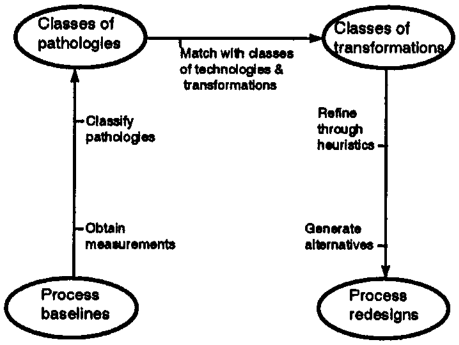

**Figure 2** KOPeR Problem Solving Cycle

The problem solving cycle of KOPeR is delineated in Figure 2. Briefly, the cycle begins with measurements obtained from a process representation, which are abstracted into a class of known process pathologies from the first taxonomy. Reengineering knowledge is then applied to match these pathologies with the appropriate technologies and transformations from the other taxonomy. One or more enabling technologies is applied to transform the represented process, generating the corresponding redesign alternatives, which can then be examined statically and simulated to support comparison and evaluation. Simulation, evaluation and selection of the most appropriate redesign alternative occur outside of KOPeR, as many established tools and techniques are available to support these activities (e.g., WITNESS 1995).

**Table III** Filtering Rules

Structured English

```
RULE 1: IF: high specialization
     THEN: OD transformation class

RULE 2: IF: high feedback
     THEN: OD transformation class
           WF transformation class

RULE 3: IF: low parallelism
     THEN: WF transformation class

RULE 4: IF: low IT support
       OR: low IT communication
       OR: low IT automation
     THEN: IT transformation class
```

A small sample of rules from KOPeR is presented in Tables III-V, and correspond to three rule groups: 1) filtering rules, 2) specialization rules and 3) refinement rules; for purpose of exposition, the rules are formatted in structured English (cf. code). As shown in Table III, the first filtering rule (RULE 1) uses a specialization measurement (associated with the "bureaucratic organization" problem type) to direct problem solving toward the organizational design (OD) class of transformations. The next rule (RULE 2) uses a feedback measurement (associated with the "'checking' approach to quality" problem type) and accomplishes a similar filtering role by directing problem solving toward both the OD and workflow (WF) classes of transformations. Workflow reconfiguration is also the target of RULE 3 and the fourth rule accepts any low measurement associated with the IT-infrastructure problem type to direct problem solving toward the information technology (IT) class of transformations.

**Table IV** Specialization Rules

Structured English

```
RULE i1: IF: OD transformation class
       AND: high specialization
      THEN: combine jobs

RULE i2: IF: OD transformation class
       AND: high specialization
      THEN: case manager position
            IO transformation class
            IT transformation class

RULE j1: IF: OD transformation class
       AND: high feedback
      THEN: delegation
            IO transformation class

RULE j2: IF: OD transformation class
       AND: high feedback
      THEN: empowerment
            IO transformation class

RULE k: IF: WF transformation class
       AND: low parallelism
      THEN: de-linearize process activities
            TEST: decomposable activities
```

Following the firing of a particular filtering rule (the OD result of RULE 1 for example), one or more corresponding specialization rules (e.g., RULE i1 and i2) focus the problem solving upon more-specific transformations (e.g., combine jobs, case manager, multidimensional work) and chain to related classes of transformations (e.g., IO, IT). For example, the case-manager transformation (RULE i2) also requires a change in the distribution of information (i.e., the IO class of transformation) for a generalist case manager to perform effectively; the case manager may also benefit from an IT transformation, say to access this information or to consult with process specialists and experts (human or machine). As another example, the firing of a WF rule (e.g., RULE 3) activates a specialization rule (e.g., RULE k) for de-linearization of process activities, which also chains to a test for whether the target sequence of activities is decomposable.

**Table V** Refinement Rules

| Structured English |
|---|
| RULE x: IF: case manager position<br>THEN: replace specialist agents<br>FIND: specialist agents |
| RULE y: IF: informate<br>THEN: increase information available |
| RULE z: IF: de-linearize process activities<br>AND: decomposable activities<br>THEN: decouple serial process steps |

The refinement rules (such as RULE x) are more specific still, and detail how to effect a particular transformation (e.g., case manager position). Notice that the rule itself does not indicate *which* specific agent-positions to convert into a generalist case-manager role; rather, selection of these specific agent roles becomes a subgoal (i.e., FIND: specialist agents) addressed by the appropriate knowledge (e.g., separate agent-roles assigned to adjacent process activities). It is also possible to link these refinement rules to other knowledge sources for still-deeper analysis. For example, many of the textbook OD issues associated with the case manager position above are addressed quite thoroughly through external information sources (e.g., texts such as that by Szilagyi and Wallace 1987, Web searches, online help), and in a sufficiently-rich networked environment, KOPeR rules can even be linked with other *knowledge-based systems* such as ACTION (Gasser et al. 1993) that specialize in OD problem solving. We return to this idea in the final section on future directions, but with this KOPeR background information for reference, we now introduce the current extension being made to develop a capability for Web-based redesign support.

## Web-based Redesign Support

The explosion of interest and activity on the World Wide Web (Web) is making this Internet (and Intranet) area the method of choice for dissemination of business information, and the broad acceptance of Java applets as an effective means for network computing highlights a natural extension for an intelligent system like KOPeR: Web-based redesign support. The basic architecture for this extension is depicted in Figure 3.

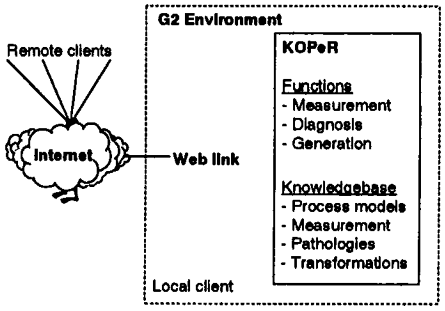

**Figure 3** Basic Architecture

The basic KOPeR system is depicted as a monolithic box in the figure, with three functions listed for 1) process measurement, 2) diagnosis of process pathologies, and 3) generation of redesign alternatives; four knowledgebase components are also listed that include 1) process models, 2) diagnostic measures and methods, 3) classes and instances of process pathologies, and 4) classes and instances of redesign transformations. KOPeR delivers the redesign functionality described above to the Web interface provided through a commercial tool (GENSYM 1997); this tool (labeled "G2 Environment" in the figure) provides direct access to a local client and includes a "Web link" interface for Internet access by remote clients. KOPeR is currently being re-implemented through this tool to move toward an "industrial strength" application that includes the capability to access process data and deliver reengineering support through ordinary html-based forms. With this, local use of KOPeR is unchanged and remote access can be accommodated through two modes: 1) table submission and 2) form input.

In the table submission mode, a remote user creates the same kind of input table that has been used to represent process data for KOPeR in the past and sends this table (e.g., via ftp or diskette) to a human on the local end who uses it to exercise the system; this represents a residual capability from the original (i.e., not Web-based) design that is not particularly innovative, but has proven to be very useful where travel and IT-infrastructure constraints inhibit KOPeR access. In the latter mode, however, the process data itself is passed directly to KOPeR through html forms that carry the same, table-based process input data; that is, the input tables are the same, only their delivery is now accomplished automatically using the Web through html forms. An area for future research extends this functionality to send *graphical process flows* to KOPeR (e.g., using IDEF notation), but this requires an additional interpretation step to be accomplished by KOPeR and relatively strict standards to be enforced for process format and syntax. We return to this extension in the section on future work below.

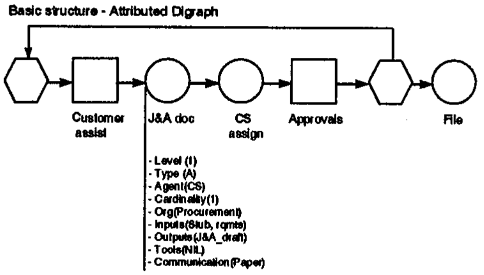

**Figure 4** Level-1 Process Flow

The level-1 process flow diagram for a procurement subprocess is presented in Figure 4. This representation adheres to the conventional attributed digraph notation, in which process activities are represented by nodes connected by task-precedence edges; attributes associated with process activities include factors such as the performing organization, agent role, types of tools and communications employed and other heuristically-valuable process variables. In this scheme, the circle icons are used to represent atomic tasks, whereas squares represent task chains that are decomposable into lower-level subprocess activity flows. The cycle represents a feedback loop that denotes a quality-control step in the process instance shown above. This reflects the representational scheme implemented through the Articulator.

An example of a remote KOPeR input table is presented in Figure 5; this input table corresponds to the level-1 process flow delineated above. Although the actual procurement subprocess (J&A) is comprised of activities represented down through three hierarchical levels, for space considerations, only the top level is shown in the figure above and included in this input table. As a (third form) normalized table, it possesses the minimal properties demanded by most database interfaces and carries the essential process data delimited into a minimum of the eight columns shown. These include 1) the activity type (e.g., task chain (TC), atomic activity (A), iteration branch (IB, IE) pairs), 2) activity level, 3) activity name, 4) predecessor activity(ies), 5) agent role(s) (and cardinality), 6) activity inputs, 7) activity outputs and 8) tool(s) employed to support each activity. From the table entries, notice that the first activity (J&A) is a task chain that is marked as level 0 to designate the process root. As shown in the figure above, the level-1 process flow begins with an iteration branch (Approve_beg), which indicates that the entire J&A sequence of activities must be repeated if a J&A document does not successfully complete the approvals process (four separate organizational approvals required at level-2 are not shown in the figure or included in the table); hence the first "real" J&A process activity is (labeled "Cust_assist") the customer assistance step, which represents the interface required for the procurement organization to capture and document the customer requirements (i.e., for some product or service to be procured).

The table shows the iteration branch as the predecessor and lists Contract Management Assistant (CMA) as the agent role assigned to perform this activity; the unit cardinality (1) indicates that only a single CMA agent is required for this particular instance of the J&A process. As the first activity, this step requires no inputs, but the output of this activity takes the form of documented customer requirements (labeled "Rqmts"). No tools (e.g., information technology) are listed for this activity, even though some common support artifacts such as telephones are presumed to exist and be used in the process; the idea of the process representation is to abstract away from details that are unnecessary to support redesign inference and to capture only those tools that are heuristically-useful for the diagnosis of process pathologies. Note the predecessor-successor relationships that flow through the table and the correspondence between outputs of certain "upstream" activities (e.g., "Rqmts," "Cust_assist") and inputs required "downstream" by others (e.g., "Rqmts," "J&A_doc"). Input tables such as this can have arbitrary length and number of hierarchical levels, with a thousand-plus activities not uncommon for process models captured in the field (see Nissen 1996).

As noted above, KOPeR is being re-implemented to effect a Web interface, but its basic input, output and functionality (e.g., reengineering "advice") remain essentially the same. Output html forms are designed to match local KOPeR execution as closely as possible. An example of the reasoning from above pertaining to redesigning the (complete, 3-level deep) J&A process is presented in Figure 6, where the system summarizes the key measurements and diagnoses for the J&A process, along with a summary of the eight redesign transformations generated by KOPeR. Although this interface extension is not technically challenging, the enhanced functionality enabled through the Web-based interface now allows KOPeR to be accessed and used remotely by anyone (authorized) with a Web browser.

Each measurement can be obtained automatically from the process representation and is defined in graph-based terms (e.g., nodes, edges, attributes, paths, cycles, etc.). As examples, process size is obtained by counting the number of activity nodes in a representation; length (not shown) is measured by the number of activity nodes comprising the longest path through the process; depth (also not shown) is defined as the largest number of discrete levels of decomposition in a process model. The footprint measure combines depth with the number of leaves (i.e., atomic tasks) in the representation to create an area-like measure (i.e., depth x leaves); parallelism is calculated as the ratio of process-size divided by length, and the IT and other "fractions" (e.g., feedback fraction, handoffs fraction) are obtained by dividing attribute counts by process-size as a normalizing factor. The Articulator facilitates this measurement step through a number of basic counting functions (e.g., nodes, edges, attributes).

The joint-reviews transformation is triggered primarily by the high feedback-fraction and handoffs-fraction measurements, along with unit parallelism. Feedback indicates the presence of many review activities in the process flow and handoffs suggest that substantial cross-functional work and communication is involved with the baseline process; parallelism denotes a linear, sequential process flow that offers potential to be conducted in parallel (i.e., through concurrent process activities). Together these correspond to pathologies that principally drive process cycle time. The joint-reviews transformation draws from the familiar practice of joint application design (JAD) in information systems development, where representatives with diverse functional interests meet synchronously to evaluate a design; in this present case, the purpose of a joint meeting is to review a particular J&A documentation package.

Similarly, KOPeR rules and knowledge are used to generate the other seven transformations, three of which focus on empowering different organizational players (i.e., the CMA, Contract Specialist and Contracting Officer) and one that calls for creation of a case team to work together on the entire J&A process. The last two transformations pertain to the IT infrastructure and lead to another redesign alternative being selected as "superior" by the process experts: workflow systems. The workflow-systems transformation subsumes a document database and is triggered primarily by the low values for IT-infrastructure (e.g., IT-Support, IT-Communication, IT-Automation) fractions, as well as the high handoffs-fraction measurement. The IT-based measurements indicate that the process is performed in a labor-intensive, paper-based, manual fashion—with both cost and cycle-time implications—while the direct interaction between handoffs-fraction and cycle time receives the same interpretation as above. The workflow systems intervention combines mechanisms for the indexed storage and retrieval of digitized documents with facilities for electronic communication and automatic document routing; workflow systems now represent a popular and effective redesign technology for the business domain (White and Fischer 1994). To reiterate, this represents the same KOPeR functionality that has existed for some time; the key extension addressed in this paper pertains to Internet access now supported for remote clients.

## Future Directions

Although not technically challenging, this extension of KOPeR functionality to support Internet access by remote clients is significant in that it demonstrates one approach to embedding intelligent system capabilities into Web-based applications, which marks a powerful new extension of AI and knowledge-based systems technology with good potential to leverage the current explosion of Internet connectivity and network computing. It also addresses many of the classic problems associated with isolated AI systems—remember the dedicated LISP machines—and in particular, the Web interface helps to embed knowledge-based systems not only in distributed *applications*, but also through distributed *organizations and processes* as well; this may prove to be a key element for effective reengineering in the future.

Future work can also address the limitations associated with the primitive table-based input now required by the system, as a capability to accept and employ *graphical* process models could be very useful. For example, such a capability would obviate the additional conversion step (i.e., from graphical representation to tabular input) now required for most *existing* process representations. This would also enable the huge number of processes developed through reengineering projects to date (esp. using the IDEF model and tools) to be accessed and evaluated by KOPeR for pathologies in a totally automated manner (e.g., evaluated at night by a Web "crawler"). Imagine coming to work in the morning with an e-mail message identifying the major pathologies afflicting your key enterprise processes (e.g., sequential process flows, manual, paper-based, labor-intensive activities, etc.) and summarizing a number of redesign transformations (e.g., joint reviews, workflow systems, etc.) generated to improve process performance! Although such a scenario remains somewhat speculative at present, this present work toward intelligent Web-based redesign support represents a step in this direction.

To close the paper with another speculative future extension of developmental work along this line, the functionality of KOPeR can also be augmented to encompass more of the redesign activities depicted in Figure 1. As noted above for example, linking KOPeR's refinement rules to external (esp. online KBS) information sources represents an approach to increasing its inferential power and both the breadth and depth of knowledge available to the system. Recall also from the figure that the KOPeR design currently stops with the *generation* of redesign alternatives; another clear line for extension would be to integrate KOPeR with simulation utilities for the *automatic testing* of these alternatives, and possibly even something like a multi-attribute utility procedure to also perform the *decision-making* activities associated with the selection step. Further, we may contemplate introducing an automated means to generate *computational*, task-level process steps and details for the redesign alternatives, where in a sufficiently-networked and -automated environment they could actually *call or execute* the physical processes through enactment of the process activities, tools and attributes that are represented as KOPeR models; such enactment could be used to perform "real" process work in an organization, for example, through interfaces with workflow systems, process-support tools (e.g., databases, communication systems) and even linkages between various organizational agents (both human and machine). Although very ambitious in scope, the technical feasibility of these extensions is nearly here today and such an integrated functionality would all but eliminate the present distinction between the activities associated with process redesign and implementation. Perhaps this represents the next "high bar" for research along these lines.

**KOPeR Input Form**

Enter the process information for all activities using the Articulator Input Guide for format, syntax and column definitions. Remember that each record represents the information for a single process activity and that all eight columns must have entries for every activity; use the symbol "n/a" where appropriate, but **leave no blank entries.**

**Process Table**

| Type | Level | Activity | Predecessor | Agent(no) | Input | Output | Tool |
|---|---|---|---|---|---|---|---|
| TC | 0 | J&A | n/a | n/a | n/a | n/a | n/a |
| IB | 1 | Approve_beg | n/a | n/a | n/a | n/a | n/a |
| TC | 1 | Cust_assist | Approv_loop | CMA (1) | n/a | Rqmts | n/a |
| A | 1 | J&A_doc | Cust_assist | CS (1) | Stub | J&A_Draft | Word_proc |
| n/a | n/a | n/a | n/a | n/a | Rqmts | n/a | n/a |
| A | 1 | CS_assign | J&A_doc | KO (1) | Rqmts | Assign | n/a |
| TC | 1 | Approvals | CS_assign | CS (1) | J&A_Draft | Approval | n/a |
| n/a | n/a | n/a | n/a | n/a | Assign | n/a | n/a |
| IE | 1 | Approve_end | Approvals | n/a | n/a | n/a | n/a |
| n/a | n/a | Approve_beg | n/a | n/a | n/a | n/a | n/a |
| A | 1 | J&A_file | Approve_end | CS (1) | Approval | J&A_complete | n/a |

**Figure 5** KOPeR Input Form

**KOPeR Output Form**

The first table summarizes the key measurement and diagnoses for the process.

| Measure | J&A | Diagnosis |
|---|---|---|
| Process size | 31 | Small |
| Parallelism | 1.00 | Sequential |
| IT-support fraction | 0.03 | Manual |
| IT-communication fraction | 0.00 | Paper-based |
| IT-automation fraction | 0.00 | Labor-intensive |
| Feedback fraction | 0.35 | Checking |
| Handoffs fraction | 0.58 | Friction |

The second table summarizes the generated redesign alternatives for the process.

| Transformation Class | Redesign | Description |
|---|---|---|
| WF | R1A | Joint reviews |
| WF | R1B | Asynchronous reviews |
| OD, IO | R2A | Empowerment - CMA |
| OD, IO | R2A | Empowerment - CS |
| OD, IO | R2A | Empowerment - KO |
| OD | R3 | Case team |
| IT | R4A | Document database |
| IT | R4B | Workflow systems |

**Figure 6** KOPeR Output Form

## References

Bashein, B. J.; Markus, M. L.; and Riley, P. 1994. Preconditions for BPR Success: And How to Prevent Failures. *Information Systems Management* Spring:7-13.

Brown, J. S.; Burton, R. R.; and De Kleer, J. 1982. Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II and III. In Sleeman, D. and Brown, J. S. eds. *Intelligent Tutoring Systems.* London, UK: Academic Press.

Gasser, L; Hulthage, I.; Leverich, B.; Lieb, J.; and Majchrzak, A. 1993. Organizations as Complex, Dynamic Design Problems. in: M. Filgueiral and L. Damas (eds.), *Progress in Artificial Intelligence, Lecture Notes in AI Number 727.* Verlog, Portugal.

GENSYM. 1997. Gensym G2 and related product documentation.

Hamscher, W. 1994. AI in Business Process

Reengineering: A Report on the 1994 Workshop. *AI Magazine* 15(4):71-72.

Hammer, M. and Champy. J. 1993. *Reengineering the Corporation: A Manifesto for Business Revolution.* New York, NY: Harper Business Press.

Jackson, P. 1990. *Introduction to Expert Systems.* Wokingham, UK: Addison-Wesley.

Mi, P. 1992. *Modeling and Analyzing the Software Process and Breakdowns.* Ph.D. diss., Dept. of Computer Science, University of Southern California.

Mi, P. and Scacchi, W. 1990. A Knowledge-Based Environment for Modeling and Simulating Software Engineering Processes. *IEEE Transactions on Knowledge and Data Engineering* 2(3):283-294.

Mi, P. and Scacchi, W. 1993. Modeling, Integrating, and Enacting Complex Organizational Processes. *Proceedings of the Fifth International Symposium on Intelligent Systems in Finance, Accounting, and Management,* December.

Mi, P. and Scacchi, W. 1995. A Meta-Model for Formulating Knowledge-Based Models of Software Development. *Decision Support Systems.*

Nissen, M. E. 1996. *Knowledge-Based Organizational Process Redesign: Using Process Flow Measures to Transform Procurement,* Ph.D. diss., School of Business Administration, University of Southern California.

Nissen, M. E. 1997. Reengineering Support through Measurement-Driven Inference. Forthcoming in *International Journal of Intelligent Systems in Accounting, Finance and Management.*

Rich, E. and Knight, K. 1991. *Artificial Intelligence.* (Second Edition) New York, NY: McGraw-Hill.

Rock, D., and Yu, D. 1994. Improving Business Process Reengineering. *AI Expert* October:27-34.

Shortliffe, E. H. 1976. *Computer-Based Medical Consultations: MYCIN.* New York, NY: Elsevier.

Szilagyi, A. D. and Wallace, M. J. 1987. *Organizational Behavior and Performance (Fourth Edition).* Scott Foresman and Company, Glenview, IL.

Talebzadeh, H.; Mandutianu, S; and Winner, C. F. 1995. Countrywide Loan-Underwriting Expert System. *AI Magazine.* 16(1):51-64.

White, T. E. and Fischer, L. 1994. *The Workflow Paradigm: New Tools for New Times.* Alameda, CA: Future Strategies.

WITNESS. Discrete event simulation software. Distributed by AT&T ISTEL, UK, 1995.

Yu, E. S. K. and Mylopoulos, J. 1996. Using Goals, Rules and Methods to Support Reasoning in Business Process Reengineering. *International Journal of Intelligent Systems in Accounting, Finance and Management,* 5:1-13.

Yu, E. S. K.; Mylopoulos, J.; and Lesperance, Y. 1996. AI Models for Business Process Reengineering. *IEEE Expert Intelligent Systems & Their Applications,* 11(4):16-22.