Journal of Universal Computer Science, vol. 9, no. 6 (2003), 563-570
submitted: 27/1/03, accepted: 17/3/03, appeared: 28/6/03 © J.UCS

Knowledge Management for Computational Problem Solving

D. T. Lee*
(Institute of Information Science, Academia Sinica, Taipei, Taiwan
dtlee@iis.sinica.edu.tw)

G. C. Lee*
(Institute of Information Science, Academia Sinica, Taipei, Taiwan
gclee@iis.sinica.edu.tw)

Y. W. Huang
(Institute of Information Science, Academia Sinica, Taipei, Taiwan
ywhuang@iis.sinica.edu.tw)

Abstract: Algorithmic research is an established knowledge engineering process that has allowed researchers to identify new or significant problems, to better understand existing approaches and experimental results, and to obtain new, effective and efficient solutions. While algorithmic researchers regularly contribute to this knowledge base by proposing new problems and novel solutions, the processes currently used to share this knowledge are inefficient, resulting in unproductive overhead. Most of these publication-centred processes lack explicit high-level knowledge structures to support efficient knowledge management. The authors describe a problem-centred collaborative knowledge management architecture associated with Computational Problem Solving (CPS). Specifically we articulate the structure and flow of such knowledge by making in-depth analysis of the needs of algorithmic researchers, and then extract the ontology. We also propose a knowledge flow measurement methodology to provide human-centred evaluations of research activities within the knowledge structure. This measurement enables us to highlight active research topics and to identify influential researchers. The collaborative knowledge management architecture was realized by implementing an Open Computational Problem Solving (OpenCPS) Knowledge Portal, which is an open-source project accessible at http://www.opencps.org.

Keywords: Algorithmic Research, Knowledge Management, Problem-Centred, Collaborative Knowledge Management Architecture, Knowledge Flow Measurement, OpenCPS
Categories: H.1, H.3, H.4

1 Introduction

Problem Solving Environments (PSEs) are platforms that provide all of the computational facilities required to solve a target class of problems. PSE features include advanced solution methods, the automatic and semi-automatic selection of solution methods, and easy means for incorporating novel solution methods [Gallopoulos et al. 94]. There are some successful single-user mathematical PSEs that

---

* Also with the Dept. of Computer Science & Information Engineering, National Taiwan University

have been created, including Matlab, MathCAD, Maple, and Mathematica. Researchers are increasingly focusing on the development of collaborative PSEs [Schuchardt et al. 01] [Chin Jr. et al. 02], an extension that enhances human intelligence (HI) by providing a communication infrastructure that encourages collaborative (even synchronous) problem solving among individuals in geographically distributed locations. Despite having a wide variety of domain-specific features, strengths of PSEs can be measured in terms of the following levels of collaboration: data sharing, software warehousing, application sharing, and workflow.

In contrast, Knowledge-Based Systems (KBS) focus on the capture, formalization and application of strong domain knowledge. Two promising KBS candidates are ontologies (concerned with the representation of static domain knowledge) and problem solving methods (PSM) (aimed at describing KBS reasoning processes in a manner that is both implementation- and domain-independent). PSMs are also associated with the dynamic reasoning of knowledge [Pérez and Benjamins 99]. If a KBS supports PSM, then the KBS possesses the capabilities of a PSE. Accordingly, it is possible to measure the KBS's strength using the collaboration levels mentioned above. In addition to adding a certain level of automation to the organization of the global knowledge structure, a PSE (with knowledge engineering support) can assist in problem solving efforts by providing a greater amount of advanced Machine Intelligence (MI).

To show how knowledge engineering can be integrated with PSEs in a manner that combines HI and MI, we choose the computational problem-solving community as a sample target. After identifying and analyzing the community's needs, we set out to capture its ontology and to articulate its knowledge flow. Our efforts were aimed at the eventual design and implementation of the OpenCPS.

2 Identifying the Needs of Researchers

Questions that both junior and senior CPS researchers are familiar with include: How does one determine if a computational problem is a new one? Is the problem well-solved? Do similar or related problems exist? How can one be sure that a result is new? How does it compare with previous results? Who are the experts in this area, and is collaboration possible?

Answering these questions requires literature searches in libraries and digital archives, but the scattered and poorly organized information that exists today is considered a barrier to knowledge acquisition. Researchers often have to spend a lot of time on literature survey, yet duplicated efforts may still occur and go unnoticed well into the peer-review process. We believe that this inefficient procedure can be improved through a dedicated knowledge management system (KMS) that uses advanced information technology.

Well conducted research surveys map the intellectual landscape of existing knowledge, identifying key individuals and ideas, the evolution of important concepts, previous and ongoing research efforts, relationships among topics, and major breakthroughs. A KMS can assist researchers in their survey efforts by providing an infrastructure for organizing and visualizing intellectual landscapes. Designing a KMS with collaboration in mind allows users to contribute feedback or new insights to a shared knowledge structure. The more that users contribute, the greater the comprehensive understanding of a topic or issue. In this matter, a KMS can assist in knowledge acquisition and dissemination, serve as a centralized source of comprehensive domain-specific knowledge objects, and facilitate interactions among online researchers, who would benefit by having more time to focus on actual research rather than tracking down the previous efforts of their colleagues.

3 Extracting CPS Ontology

Ontologies serve as explicit specification of domain concepts and their relations. Standards for describing ontologies include Topic Maps, RDF/RDFS and DAML+OIL. The list of authoring tools for modelling domain knowledge includes Protégé 2000, OntoEdit and OilED. A complete overview of ontological concepts and issues can be found in [Fernandaz-Lopez 02].

We propose using three conceptual spaces—problem, solution, and implementation—to provide a detailed and accurate description of CPS ontology. Problem space objects consist of uniquely identifiable computational problems, solution space objects consist of algorithmic solutions, and implementation space objects assist in carrying out the solutions. A high-level CPS KB abstraction can be modelled as a collection of these objects, their intra-relations within each space, and their inter-relations across the three spaces.

As shown in Fig. 1, equivalent problems are grouped together, sub-problems, super-problems, and variant problems are also indicated. Solutions which solve the same target problem but have different time complexities are shown. Implementations for the respective solutions are also pointed out. Articulating these relationships requires identifying the intrinsic attributes of problems, solutions, and implementations. A computational problem is associated with the following attributes: problem name, description, problem category, equivalent problems, sub-problems, super-problems, variant problems, formal definition, input variables, output variables, output measure, problem status, existing solutions, and related publications. An algorithmic solution is associated with the following attributes: solution name, target problem, description, pseudo code, complexity, problem solving strategy, existing implementations, and related publications.

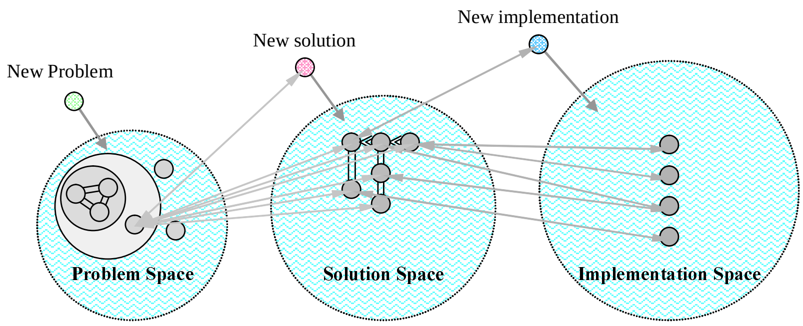

Figure 1: Conceptual Map of CPS

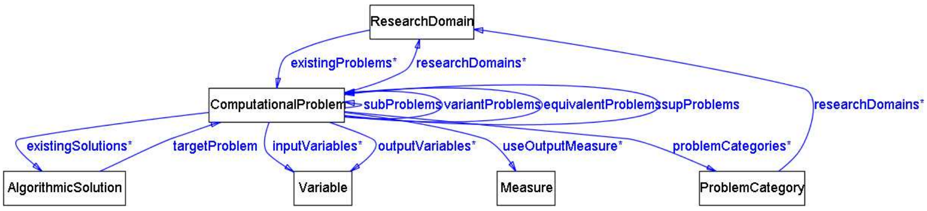

Figure 2: Ontology (partial) of CPS

Finally, an implementation is associated with the following attributes: implementation name, target solution, description, environment, offline execution, online execution, programming language, and related publications. We then model these concepts and describe the cross-relationships among OpenCPS knowledge objects using an RDF schema. A partial visualization generated by Protégé 2000 is shown in Fig. 2.

4 Determining KM Architecture

Following guidelines for running a KM project or creating a KM portal as described by [Applehans 98] and [Mack et al. 01], we propose a KM architecture that consists of the following four parts (see Fig. 3):

1. People, specifically those who produce and use knowledge objects. Member roles include administrator, knowledge author, knowledge reviewer, and technology designer. If no active knowledge authors can be found, it should be considered a warning sign that the KM initiative in question is failing.
2. Knowledge objects, meaning sharable information based on extracted knowledge structures. The three knowledge object types in OpenCPS are computational problems, algorithmic solutions, and implementations. Here a warning sign of KM initiative failure is the lack of understanding of core content.
3. Technical infrastructure. Technology enables the capture, storage, and delivery of content at the user's discretion. The costs associated with constructing, maintaining, and improving technological interfaces are a key issue for enablers. To ensure a quick launch of our KM project, we leveraged the technology layer with existing content management systems (CMSs) that address the processes of creating, managing, and deploying content, but kept content separate from its presentation. A thorough description of what to consider when choosing a CMS can be found in [Merant 02].
4. Knowledge management processes. The KM life cycle models shown in Table 1 [Nissen 2002] can be simplified for our proposed KM architecture for two reasons: we already extracted the well-formed CPS knowledge structure, and content management was delegated to the workflow-enabled CMS. When investigating the essential components of human interaction and creativity, we discovered that [Hneiderman 00] has proposed four core concepts: a) new knowledge is built on previous knowledge, b) powerful tools can support creativity, c) refinement is a social process, and d) creative

work is not complete until it is disseminated. The resulting four-phase "generate excellence (Genex)" framework—collect, relate, create, and donate—provides a perspective that can be applied in the form of non-linear knowledge management processes in our work.

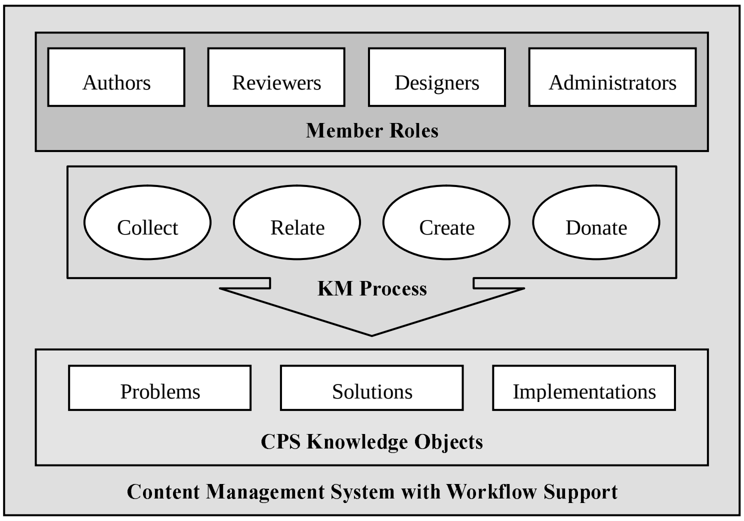

Figure 3: KM Architecture of OpenCPS

Model	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6
Despres and Chauvel	Create	Map/bundle	Store	Share/transfer	Reuse	Evolve
Gartner Group	Create	Organize	Capture	Access	Use	
Davenport & Prusak	Generate	Codify	Transfer			
Nissen	Capture	Organize	Formalize	Distribute	Apply	
Amalgamated	Create	Organize	Formalize	Distribute	Apply	Evolve

Table 1: Knowledge Management Life Cycle Modeles [Adapted from Nissen 02]

5 Measuring the Knowledge Flow

To support research of large scale or hard-core problem solving over a long time period, we can highlight important CPS objects to attract the attention of researchers and new participants. By adopting the commonly accepted axiom— the more valuable the information, the greater its access rate, we established a means of measuring knowledge flow—that is, the process of sharing knowledge among people or knowledge processing mechanisms [Nissen 02] [Hai 02].

Knowledge flow measurement can help identify important topics within a collaborative knowledge portal. Some simple measures are provided by associating the actions of knowledge workers according to the four knowledge management processes—collect, relate, create and donate, which are highly related to human creativity. Tracking these measures with OpenCPS helps in the identification of active subjects, which may take the form of problems, solutions, implementations, and accessible resources within the OpenCPS knowledge portal.

6 Related Work

A considerable amount of information is available on using citation analysis methods to extract "science maps" [Small 03]. Two examples are Author Co-citation Analysis (ACA) [White 03] and bibliographic coupling [Morris 03]. These publication-based approaches have the advantage of being able to utilize existing citation databases such as the Science Citation Index (SCI) [Garfield 84]. By analyzing citation databases, maps and trends of science can be discovered.

The goal of HistSite [Garfield et al. 03] is to extract historiographs that reveal revolutionary changes in scientific viewpoints. ISI's Essential Science Indicator (ESI) identifies "research fronts" and visualizes them using a product called Special Topics that classifies them as either "emerging research fronts" or "fast moving research fronts" [Small 03]. NEC's ResearchIndex (CiteSeer) [Lawrence et al. 99] is a popular publication-centred knowledge portal that explores ways to autonomously create citation indices instead of working exclusively with an existing index.

Several efforts have focused on formalizing and indexing computational problems. The NP optimization problem compendium is a noteworthy electronic book full of useful information on that particular class of problems [Crescenzi and Kann 99]. It contains a large KB built on the formalization of a computational problem, and creates a well-organized map of knowledge items and their relationships. The U.S. National Institute of Standards and Technology has established a "Dictionary of Algorithms and Data Structures" that formally defines over 1,000 computational problems. However, both projects lack adequate CPS knowledge structures and collaboration facilities that are required to become useful knowledge portals.

7 Discussions and Conclusion

In this paper we have proposed a problem-centred collaborative knowledge management architecture that differs greatly from publication-centred approaches [Lawrence et al. 99] [Morris 03] [Willinsky 00] [White 03]. The OpenCPS knowledge portal uses three conceptual spaces to create formal knowledge objects that define the frontiers of CPS research domains. The problem space, which corresponds to well-defined computational problems, is the heart of the CPS research domain. The solution space, which corresponds to existing algorithmic solutions, provides an up-to-date theoretical view of problem solving efforts made by CPS researchers. Finally, the implementation space is a practical view of the CPS research domain that corresponds to existing implementations. CPS researchers can search a problem space to see if a computational problem is well-defined, if algorithmic solutions are available, or if there is a need for a new computational problem object and the potential for a collaborative effort to create it.

Similar to publication-centred efforts, we aim to create an efficient means of extracting, organizing, and visualizing scientific knowledge—but adopt a problem-centred approach that focuses on encouraging collaborative efforts in organizing knowledge. In our view, the automated extraction of scientific maps is useful, but requires human intelligence for accurate reflections. Furthermore, we believe that a problem-centred, online, collaboration-enabled platform avoids problems related to publication lag times. For instance, automated research front extraction based on citation analysis may not provide an accurate reflection of “hot” research topics; a problem may draw tremendous research attention but few papers have been published because a good (or potentially good) solution is not yet discovered. In contrast, our OpenCPS allows for real-time reflections on current topics based on knowledge object access rates. We summarize the following four advantages of our approach:

1. Both knowledge objects (formalized) and content objects (free-formatted) are by default publicly accessible to portal users. Users can employ these objects to create personal research surveys in the form of an online document. The more profuse the content, the more effective these collect and relate processes become.
2. Participants will be encouraged to donate personal knowledge or other portal content objects that will help illuminate a research domain with better materials than any single researchers could ever generate on their own. Donating the most recent results or surveys and other research efforts will make it easier to locate unsolved computational problems or to make use of existing algorithmic solutions.
3. Knowledge objects that represent and visualize solution implementations can be used to mediate interactions between users and conceptual algorithmic solutions. They are useful for both academic and corporate/industrial applications.
4. Access rights and workflows are enforced for content objects, so to allow a personal workspace for each user, which can be used to store content objects and to exchange information with other members.

We have implemented our OpenCPS knowledge management architecture using Plone [Plone], an open-source, workflow-enabled content management system based on Zope. For comparison, see Ontoweb Portal [Ontoweb Portal], an ontology-based information exchange system for KM and e-commerce. While we have built experimental content objects that conform to the specifications of the extracted ontology, an evaluation of the actual usefulness of the OpenCPS based on real-world experiences has to be observed over a longer period of time.

Acknowledgements

This work was supported in part by the National Science Council under the Grants NSC-90-2219-E-001-001 and NSC90-2219-E-001-002.

References

[Applehans 98] Applehans, W., Globe, A., and Laugero, G.: “Managing Knowledge: A Practical Web-Based Approach (Addison-Wesley Information Technology Series),” ISBN 0-21-43315-X, 1998.

[Benjamins 97] Benjamins, V. R., Fensel, D., Chandrasekaran, B.: “PSMs do IT! - summary of track on sharable and reusable problem-solving methods of the 10th KAW'96,” Int. J. of Human-Comput. Studies 47, 1997.

[Crescenzi and Kann 99] Crescenzi, P., Kann, V.: “A compendium of NP optimization problems,” http://www.nada.kth.se/~viggo/problemlist/compendium.html

[Fernandaz-Lopez 02] Fernandaz-Lopez, M. (Ed): “Deliverable 1.4: A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies,” Available: http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/

[Gallopoulos et al. 94] Gallopoulos, E., Houstis, E. N., and Rice, J. R.: “Computer as thinker/doer: Problem solving environments for computational science,” IEEE Comp. Sci. Engr. 1, 11-23, 1994.

[Garfield 84] Garfield, E.: “The Permuterm Subject Index: An autobiographical review,” Essays of an Information Scientist, 7, 546–550, 1984.

[Garfield et al. 03] Garfield, E., Pudovkin, A. I., and Istomin, V. S.: “Why Do We Need Algorithmic Historiography?” JASIST, 54, 5, 400-412, 2003.

[Hai 02] Hai, Zhuge: “A Knowledge Flow Model for Peer-to-Peer Team Knowledge Sharing and Management,” Expert Systems with Applications 23, 23-30, 2002.

[Hneiderman 00] Hneiderman, B.: “Creating Creativity: User Interfaces for Supporting Innovation,” ACM Transactions on Computer-Human Interaction, 7, 1, 114–138, 2000.

[Chin Jr. et al. 02] Chin Jr., G., Leung, L. R., Schuchardt, K., Gracio, D: “New Paradigms in Problem Solving Environments for Scientific Computing,” IUI2002, San Francisco, California, USA, 39-46, 2002.

[Lawrence et al. 99] Lawrence, S., Giles, C. L., Bollacker, K.: “Digital Libraries and Autonomous Citation Indexing,” IEEE Computer, 32, 6, 67-71, 1999.

[Mack et al. 01] Mack, R., Ravin, Y., Byrd, R. J.: “Knowledge portals and the emerging digital knowledge workplace”, IBM Systems Journal, 40, 4, 925-955, 2001.

[Mehta 86] Mehta, U.: “Knowledge Based Systems for Computational Aerodynamics and Fluid Dynamics, Knowledge Based Problem Solving,” ISBN 0-13-516576-8, 183-212, 1986.

[Merant 02] Whitepapers of Merant Collage: “10 Critical Elements in Choosing a Content Management Platform,” Avaliable:
http://www.merant.com/Products/WCM/collage/home.asp#whitepapers

[Morris 03] Morris, S., Yen, G., Wu, Z., and Asnake, B.: "Time Line Visualization of Research Fronts," Journal of the American Society for Information Science and Technology, 54, 5, 413-422, 2003.

[Nissen 02] Nissen, M. E.: “An Extended Model of Knowledge-Flow Dynamics,” Communications of the Association for Information Systems, 8, 251-266, 2002.

[Ontoweb Portal] http://ontoweb.aifb.uni-karlsruhe.de/

[Pérez and Benjamins 99] Pérez, A. G., Benjamins, V. R.: “Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods,” IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, 1999.

[Plone] http://plone.org/

[Schuchardt et al. 01] Schuchardt, K., Didier, B., and Black, G.: “Ecce – A Problem Solving Environment’s Evolution Toward Grid,” Pacific Northwest National Laboratory, Grid Computing Environments 2001 Special Issue of Concurrency and Computation: Practice and Experience

[Small 03] Small, H.: “Paradigms, Citations, and Maps of Science: A Personal History,” Journal of the American Society for Information Science and Technology, 54, 5, 394-399, 2003.

[Willinsky 00] Willinsky, J.: “Proposing a knowledge exchange model for scholarly publishing,” Current Issues in Education [On-line], 3, 6, 2000.

[White 03] White, D.: "Pathfinder Networks and Author Cocitation Analysis: A Remapping of Paradigmatic Information Scientists,” Journal of the American Society for Information Science and Technology, 54, 5, 423-434, 2003.