Working Session on Interoperable Reengineering Services
IWPC 2005 Working Session Proposal

Dean Jin
Dept. of Computer Science
University of Manitoba, Winnipeg, Canada
djin@cs.umanitoba.ca

Andreas Winter
Institute for Software Engineering
University of Koblenz-Landau, Germany
winter@uni-koblenz.de

Abstract

Interoperability is the challenge involved in getting software components to work together. This working session discusses in depth the state of the art in reengineering tool interoperability. A distinction is made between data interoperability and service interoperability. The limitations and promise of each of these aspects of interoperability will be introduced. Examples of existing solutions and an examination of the ways they can be improved to further enhance the reengineering process will be discussed. An evaluation of open problems will show future research perspectives in reengineering tool interoperability.

1 Introduction

The past decade has seen an increased awareness of the challenges involved in maintaining and reengineering software systems. This has led to the development of a broad array of tools and tool components that provide support for program comprehension, software understanding, and software evolution tasks. Most of these tools address a specific problem, offering specialized functionality that assists software practitioners in carrying out a particular task in an automated or semi-automated fashion.

Despite this progress, very few tools have been developed that provide any means for sharing their analyses or the results they provide. To assist in reengineering tasks, tools must be able to work together to provide effective and consistent support for software evolution.

An ideal reengineering environment would allow the use of a suite of tools, each providing support for a particular task. *Interoperability* among these tools would let practitioners leverage the results of different analyses and help speed up the reengineering process. Lack of interoperability among tools continues to be a serious obstacle to the adoption and use of automation in the reengineering process [10, 6].

2 Interoperability in Reengineering

The aim of this working session is to discuss in depth two aspects of interoperability among reengineering components. *Data interoperability* provides a means for exchanging data between different tools and enables coupling of tools to form tool chains. *Service interoperability* involves sharing the functionality provided by a set of tools and tool components in a collaborative, integrated environment.

Data interoperability addresses the problem of information formulation or ‘packaging’ for transfer among tools. Service interoperability addresses the problem of applying well-defined, self-contained tool component behavior to the information stored by another tool. The ability to access and extract data stored by tools participating in a shared-service collaboration is an integral part of service interoperability. In this way, service interoperability builds on the exchange techniques provided by data interoperability. In this working session, the use of data interoperability technology in conjunction with service interoperability will be explored.

2.1 Data Interoperability

Data interoperability forms the foundation from which all kinds of tool and service interoperability can be derived. Tools and components that operate independently must have a means for packaging and transferring data to other tools. An important aspect of this process involves the additional transfer of information about the data, commonly known as a *schema* or *metadata*. The ability to transfer both data and schema allows more meaningful interaction to take place between tools.

The *Graph eXchange Language (GXL)* [12, 5] was developed to provide an easy to use, general purpose mechanism for packaging and exchanging graph-based data with it’s corresponding schema information. GXL originated in discussions held at various international reengineering conferences including IWPC [9] and WCRE [6] and was ratified as a standard exchange format in reengineering at Dagstuhl Seminar 01041 “Interoperability of Reengineering Tools” in January 2001 [2]. In the years since GXL ratification, groups in reengineering, graph transformation, graph visualization, business process modelling and other areas of software engineering have added support for GXL to their tools and collected experiences in working with GXL. These experiences and further requests for extending GXL will be summarized during the working session. In particular, the use of GXL in the context of service interoperability will be explored.

2.2 Service Interoperability

The primary barrier to achieving interoperability among reengineering tools can be attributed to differences in *syntax* and *semantics* in the representation of software knowledge that each tool maintains. While syntactic differences are easily handled through representational mapping and data translation, semantic differences are more difficult to resolve. No single information model captures all the views of software supported by tools currently available [3, 4, 11]. Achieving reengineering tool interoperability necessitates the development of a means for mediating the syntactic and semantic differences that exist between tools. This points to a solution that operates above the level on which data is packaged and exchanged.

A *service* is the functionality provided by a tool (or more often a tool component) that, when given a set of one or more inputs, generates a corresponding output that is relevant to a user. Services are typically viewed independently from the tools that implement them. In a collaborative tool environment, services are described in terms of their:

- *address*, showing where they can be found,
- *interfaces*, describing how they can be requested,
- *business protocols*, describing the order in which they perform subtasks, and
- *semantics*, specifying how they function [1]

Service interoperability focuses on finding ways of sharing services among tools.

One approach recently investigated enables tool interoperability by sharing services among tools that represent software in a *conceptually equivalent* manner. The *Ontological Adaptive Service-Sharing Integration System (OASIS)* [8, 7] makes use of specially constructed, external tool adapters and a domain ontology to facilitate interoperability among a set of tools. The tool adapters extract and filter software facts, addressing the syntactic aspect of integration process.

The domain ontology stores representation and service concepts shared among tools participating in an integration. A service offered by a tool can be shared only when the concepts required by the service intersect with the concepts supported by another tool. The construction of the domain ontology and the determination of the conceptual equivalencies that exist among tools addresses the semantic aspect of the integration process.

3 Goals and Expected Results

The main goal of this working session is to promote a *lively discussion* on data and service interoperability technologies. In particular, we are interested in discussing past experiences that participants have had with these technologies and how they can be modified or enhanced to best address the tool integration needs of software practitioners. Within this context, we expect to achieve the following:

- Summarize the state of the art in interoperability in reengineering,
- Discover problems and deficiencies of current solutions,
- Postulate further research perspectives, and
- Find and evaluate new solutions from areas outside of the reengineering community

4 Working Session Format

Our goal is to split the working session into two parts: a tutorial and a discussion. The intent of the tutorial (ca. 30 minutes) is to introduce the working session topics and to pose controversial remarks and questions meant to stimulate further discussion. A short presentation on the current status of GXL for data interoperability in reengineering will be provided. This will include a report on the successes of GXL and the lessons learned while using it for inter-tool information exchange. Open questions on applying GXL for exchanging data will be discussed.

Following this, *service interoperability* will be motivated as the next step towards enabling interoperability among reengineering tools. We will introduce ways of describing services in relation to addressing, interfacing using data exchange and filtering, business protocols and semantics. A discussion of the requirements for exchanging data and identifying conceptual equivalencies in a service-oriented environment will be provided. Examples drawn from the OASIS implementation will be used to demonstrate the concepts being discussed.

The intent of the discussion (ca. 60 minutes) is to formulate requirements and possible solutions for service-oriented reengineering environments. This will include discussions on:

- how reengineering services can be described
- how service-oriented interoperability can be established in reengineering
- how reengineering services can be extracted from existing tools
- how service-sharing can be used to combine reengineering services into interoperable tool environments

Issues related to ontology-based approaches to service-sharing will be introduced and critiqued. The goal here is to identify how the existing OASIS implementation can be enhanced to more effectively support reengineering tool integration. Extensions, enhancements and changes to enable the use of GXL in service interoperability environments will be assessed.

The discussion part will be held as a moderated *Park Bench Panel*. In this type of discussion, four panellists are permitted to discuss an item of controversy with each other. To allow the audience participation, the "oldest" panellist is identified and must leave the panel if someone from the audience wishes to join in the discussion. This enables an initial intensive discussion among experts while facilitating the input of new ideas and controversies by new participants to the discussion. The initial panel will set up with experts on reengineering tools, interoperability, software components, and service-orientation integration.

5 Working Session Organizers

Dean Jin is an Assistant Professor in the Department of Computer Science at the University of Manitoba in Winnipeg, Canada. He developed OASIS as part of his Doctoral studies at Queen's University. His current research interests include tool support for developing, evolving and maintaining software, service-oriented approaches to systems integration, and the application of categorization and ontology-based methods to interoperability.

Andreas Winter is an Assistant Professor in the Department of Computer Science at the University of Koblenz-Landau in Koblenz, Germany. He was involved in the development of GXL. Current research topics are the specification and coupling of small, repository-based reengineering components, meta-modelling, and software-(re-)engineering processes. He is spokesman of the reengineering interest group of the German computer society (GI-SRE).

6 Conclusion

A myriad of tools and tool components support all kinds of software analysis, exploration and visualization techniques. Enabling service interoperability among these tools would provide reengineering practitioners with an effective means for taking full advantage of the rich functionality that these services provide. This working session will lay the groundwork for further discussion leading to the realization of a service-interoperable reengineering tool suite.

References

[1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer, Berlin, 2004.

[2] J. Ebert, K. Kontogiannis, and J. Mylopoulus. Interoperability of Reverse Engineering Tools. Dagstuhl Seminar 01041 Report, January 2001. URL: `http://www.dagstuhl.de/files/Reports/01/01041.pdf`.

[3] J. Ebert, B. Kullbach, and A. Winter. "GraX: Graph Exchange Format". In *Proceedings of the Workshop on Standard Exchange Formats (WoSEF) at ICSE'00*, Limerick, Ireland, 2000.

[4] R. Ferenc, T. Gyimóthy, S. E. Sim, R. C. Holt, and R. Koschke. "Towards a Standard Schema for C/C++". In *Proceedings of the 8th Working Conference on Reverse Engineering (WCRE'01)*, Stuttgart, Germany, October 2001.

[5] R. Holt, A. Schürr, S. Sim, and A. Winter. Graph eXchange Language. URL: `http://www.gupro.de/GXL/index.html`.

[6] R. C. Holt, A. Winter, and A. Schürr. "GXL: Toward a Standard Exchange Format". In *Proceedings of the 7th Working Conference on Reverse Engineering (WCRE'00) Panel on Reengineering Exchange Formats*. IEEE Computer Society Press, November 2000.

[7] D. Jin and J. R. Cordy. "A Service Sharing Approach to Integrating Program Comprehension Tools". In *Proceedings of the ESEC/FSE 2003 Workshop on Tool Integration in System Development (TIS 2003)*, pages 73–78, Helsinki, Finland, September 2003.

[8] D. Jin and J. R. Cordy. Ontology-Based Reverse Engineering Tool Integration: The OASIS Service-Sharing Methodology. Submitted to the 13th IEEE International Workshop on Program Comprehension (IWPC 2005), May 2005. St. Louis, Missouri.

[9] T. C. Lethbridge. "Report from the Dagstuhl Seminar on Interoperability of Reengineering Tools". In *9th International Workshop on Program Comprehension (IWPC'01)*, page 119, Toronto, Canada, 2001.

[10] H. A. Müller, J. H. Jahnke, D. B. Smith, M.-A. Storey, S. R. Tilley, and K. Wong. "Reverse Engineering: A Roadmap". In A. Finkelstein, editor, *The Future of Software Engineering*, International Conference on Software Engineering (ICSE'00), Limerick, Ireland, June 2000. ACM Press.

[11] S. E. Sim, R. C. Holt, and R. Koschke. WoSEF – Workgroup on Standard Exchange Format, Progress Towards a Format. URL: `http://www.cs.toronto.edu/~simsuz/wosef/`.

[12] A. Winter. "Exchanging Graphs with GXL". In S. L. P. Mutzel, M. Jünger, editor, *9th International Symposium on Graph Drawing (GD'01)*, volume 2265 of *Lecture Notes in Computer Science*, page 485. Springer-Verlag, 2002.