First International Workshop on Meta-Models and Schemas for Reverse Engineering ateM 2003

Jean-Marie Favre
University of Grenoble
France
www-adele.imag.fr/~jmfavre/
Jean-Marie.Favre@imag.fr

Mike Godfrey
University of Waterloo
Canada
plg.uwaterloo.ca/~migod/
migod@uwaterloo.ca

Andreas Winter
University of Koblenz-Landau
Germany
www.uni-koblenz.de/~winter/
winter@uni-koblenz.de

Abstract

The International Workshop on Meta-Models and Schemas for Reverse Engineering *aims at utilizing meta-technology for the development of reverse engineering toolsets. The workshop objective is to determine and specify areas in reverse engineering which will benefit from meta-technology and to discover and discuss meta-technologies which will provide significant additional support for developing and using program understanding and analysis tools.*

1. Introduction

Schemas are essential for the development of reverse engineering tools. They define internal (data) structures used by these tools and also specify the underlying semantic model of various analysis services. It is well known in the reverse engineering community that the development of schemas including their appropriate fact extractors is time-consuming and costly. The community would benefit from support for defining and using schemas and schema-based tools. *Meta-technology*, i. e., the use of meta-modeling techniques, promises to provide a convenient framework to efficiently define and apply schemas in reverse engineering.

The *International Workshop on Meta-Models and Schemas for Reverse Engineering* focuses on the application of schemas in Reverse Engineering and on techniques to define and utilize these schemas.

2. Use of Schemas in Reverse Engineering

Research on Reverse Engineering and the development of program understanding and analysis tools results in a large amount of different schemas. They deal with certain programming languages like Cobol, C/C++, and Java on abstract syntax level or with (multi-language) source code landscapes on architectural level [2], [7]. Besides the definition of problem and domain related data structures, schemas enable *Adaptability* and *Interoperability* of reverse engineering tools.

Adaptability:

Explicitly defined schemas allow reverse engineering tools to be adaptable. For generic tools, knowledge of the analysis context such as programming languages, granularity, or problem domain is specified within a problem-related schema. Services and techniques provided by these tools are defined upon these schemas and use standardized analysis facilities, already defined for data matching those schemas. Approaches like these are used e. g. within GUPRO [1], Swagkit [8], or Rigi [12].

Here, schemas aid in customizing general-purpose tools into special-purpose tools. General mechanisms for querying, analyzing and manipulating graph-like structures are used to solve individual problem-dependent program analysis problems.

Interoperability:

An essential prerequisite to interoperability of reverse engineering tools is their agreement on the shared data. Here, schemas are used to specify the interoperability context i. e.the structure of the data to be interchanged.

GXL [11] provides an widely accepted mechanism to exchange (graph-like) data in Reverse-Engineering. Data, matching the interoperability context is exchanged by XML streams. Besides the exchange of instance data, GXL also offers support for describing and exchanging schemas, using the same kind of XML streams.

Adaptability and Interoperability are two examples in Reverse Engineering, which — in parts — already benefit from using meta-technology. Further areas should be identified during the workshop.

Proceedings of the 10th Working Conference on Reverse Engineering (WCRE'03)
1095-1350/03 $ 17.00 © 2003 IEEE

3. Meta-Technology for Reverse Engineering

Within the object-oriented community, the use of meta-models is nowadays also recognized as a natural way to capture software models. It is successfully applied to implement reengineering and design recovery tools. Meta-modeling approaches are used for specifying and extending the Unified Modeling Language UML [6] and meta-modeling techniques like Meta-Object Facility MOF [3] are developed around UML. Furthermore markup languages like the Extensible Markup Language XML [9] are embedded in meta-modeling environments. The structure of XML files is specified by schemas in DTD [9] or XML-Schema [10]. XML Metadata Interchange XMI [4] provides a XML-based interchange format for metadata according MOF.

The emphasis on the meta-level has been pushed further by the recent Model-Driven Engineering paradigm, and in particular by the MDA industrial standard [5]. For example, OMG's Common Warehouse Metamodel (CWM) includes standard schemas for COBOL, C, etc; QVT (Query-View-Transform) is an emerging standard for program transformation.

In fact, many meta-techniques are already available. These include support for defining and dealing with schemas, grammars, ontologies, and meta-models. This workshop will examine how these techniques can be used successfully for developing reverse engineering tools in addition to the already applied meta-technologies.

4. Topics of Interest

The workshop aims enabling reverses engineering by meta-modeling. Thus, the topics of interest include the following, as they relate to reverse engineering:

- Relationship between meta-models, schemas, ontologies and grammars
- Methods and techniques to define meta-models and schemas
- Methods and techniques to reuse meta-models and schemas
- Model Driven Engineering and Model Driven Architecture
- Schemas and Reference Schemas for legacy applications
- Schemas and Reference Schemas for Web Applications and Component technologies
- Schemas and Reference Schemas for Architecture Recovery
- Grammar Reverse Engineering and its relationships to Meta-models
- Meta-technologies and standards such as GXL, XMI, MOF, JMI, CWM, QVT, etc. and their application in reverse engineering
- Meta-environments and meta-tools for reverse engineering
- Schema extraction and evolution applied on meta information about software systems

This workshop is intended for researchers interested in *meta-technologies* and *reverse engineering* to discuss their progress and explore future directions. The goals of the workshop are:

- to exchange experience about meta-models, schemas and associated techniques when used in reverse engineering,
- to discover areas of mutual collaboration, and
- to envision future trends in the field of meta-technologies in reverse engineering.

5. Conclusion

The purpose of this workshop is to bring researchers from different communities and to study how schemas, meta-models, and associated technologies could be used in the context of reverse engineering. The workshop also intends to summarize results of various activities on standard and reference schemas in engineering schemas, exchange of schema information, collections on reverse engineering schemas evolved within the last years (e.g., the WoSEF workshop at ICSE 2000, Dagstuhl seminar #01041).

References

[1] J. Ebert, B. Kullbach, V. Riediger, and A. Winter. GUPRO – Generic Understanding of Programs, An Overview. *Electronic Notes in Theoretical Computer Science (`http://www.elsevier.nl/locate/entcs/volume72.html`)*, 72(2), 2002.

[2] M. Godfrey. High Level Schemas, A Journey through the Bush. `http://www.cs.toronto.edu/~simsuz/wosef/presentations/session2.pdf` (08.05.2002) Presentation at WoSEF 2000, Limerick, Ireland, 6/2000.

[3] Meta Object Facility (MOF) Specification. `http://www.omg.org/cgi-bin/apps/doc?formal/03-03-01.pdf`, 4/2002.

[4] OMG XML Metadata Interchange (XMI) Specification. `http://www.w3.org/TR/2000/REC-xml-20001006.pdf`, 1/2002.

[5] MDA Guide. `http://www.omg.org/docs/omg/03-06-01.pdf`, 6/2003.

[6] OMG Unified Modeling Language Specification. `http://www.omg.org/cgi-bin/apps/doc?formal/02-04-03.pdf`, 3/2003.

[7] Rogues Gallery of Schemata. `http://www.cs.toronto.edu/~simsuz/schema/`.

[8] SWAG Software Toolkit. `http://www.swag.uwaterloo.ca/~swagkit/`.

[9] Extensible Markup Language (XML) 1.0 (Second Edition). `http://www.w3.org/TR/2000/REC-xml-20001006.pdf`, 10/2000.

[10] XML Schema Part 0: Primer, W3C Recommendation, 2 May 2001 . `http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/`, 5/2001.

[11] A. Winter. Exchanging Graphs with GXL. In *P. Mutzel, M. Jünger, S. Leipert (eds.) Graph Drawing, Vienna, Austria, September 23-26, LNCS 2265, Springer*, 485–500. 2002.

[12] K. Wong. RIGI User's Manual, Version 5.4.4. `http://www.rigi.csc.uvic.ca/rigi/rigiframe1.shtml?Download`, 30. June 1998.