
Editorial

Joachim Baumeister*

Intelligent Systems (Computer Science VI),
University Würzburg,
Am Hubland, Würzburg 97074, Germany
E-mail: joba@uni-wuerzburg.de

*Corresponding author

Grzegorz J. Nalepa

Institute of Automatics,
AGH – University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
E-mail: gjn@agh.edu.pl

Biographical notes: Joachim Baumeister received his PhD in 2004 for the research on the agile development of knowledge systems (dissertation award for the best applied PhD thesis). Current research focuses on the engineering of intelligent information systems, especially semantic Wikis. He is leading a number of application projects in the technical, medical and biological domain. He has authored and co-authored more than 70 reviewed publications in the field of knowledge-based systems and the semantic web. He also organised several national conferences and workshops and is serving as a Program Committee Member for many international conferences and workshops.

Grzegorz J. Nalepa holds a position of Assistant Professor in AGH UST in Krakow, Poland. Since 1995, he has been involved in number of research projects. He has co/authored over 100 publications, from the domains of knowledge engineering, intelligent systems and software engineering. He formulated a new design approach for rules called eXtended Tabular Trees (XTT). He is involved in several conferences and workshops, including *DERIS RuleApps*, *AISEW* and *KESE*. Since 2008, he has been co-organising the *Knowledge and Software Engineering Workshop (KESE)* at KI, the German AI conference. He is the President of the Polish Artificial Intelligence Society (PSSI).

Intelligent systems have been successfully developed in various domains based on techniques and tools from the fields of knowledge engineering and software engineering. Declarative software engineering techniques have been established in many areas, such as knowledge systems, logic programming, constraint programming, and lately in the context of the semantic web and business rules.

Starting with the 1st Workshop on Knowledge Engineering and Software Engineering (KESE) in 2005 at the 28th German Conference on Artificial Intelligence (KI-2005) in Koblenz, Germany, the KESE workshops developed to a series of annual meetings, where researchers and practitioners exchange their experiences and research results from both fields of software engineering and artificial intelligence. Between 2005 and 2009,

five KESE workshops took place (see <http://kese.ia.agh.edu.pl>), and 35 papers have been presented. In the past years, we have seen a fruitful exchange on various topics that show the interactions between software engineering and knowledge engineering. In particular, the following topics were discussed thoroughly:

- knowledge engineering approaches inspired by model-driven software engineering architectures
- refactoring and unit testing techniques known from software engineering that were applied to knowledge engineering
- software systems inspired by agent-based architectures
- quality measures from software engineering applied to rule-based systems
- web service architectures for intelligent systems
- mashups of intelligent applets
- advanced UML research applied in knowledge engineering.

In this issue, we collected a selection of papers that clearly show the symbiotic relationships between the disciplines of knowledge engineering and software engineering.

Bach et al. introduce a multi-agent architecture for sharing and collaboratively developing community experience. The SEASALT approach successfully applies results from software engineering research and combines them with knowledge engineering techniques. An example system taken from travel medicine demonstrates its practical applicability.

Search is a key technology in today's software systems. In the context of the MEDICO project, Roth-Berghofer and Forcher evaluate the understandability of the semantic search engine RadSem that employs multimodal access and explanation to the users. The experiments were conducted by retrieving and assessing medical documents.

Tsai and Chan investigate the development of an intelligent system for sentence retrieval and novelty mining. Here also, the successful combination of software engineering and knowledge engineering techniques is demonstrated by the implementation and evaluation strategies of the system.

Advanced business systems employ rule-based expert knowledge into their control process. Lukichev focuses on evaluation aspects of such systems. He presents verification measures for rule-based systems by introducing the flexible approach of verifier rules. Such meta-rules are easy to maintain and flexibly can adapt to different application systems.

Interaction and collaboration between advanced software systems is also a key technology. Iftikhar et al. address methods to obtain better interoperability between advanced semantic software systems. Their work build on standards developed in software engineering and semantic web research.

In recent years, software engineers seek advanced conceptual tools to support the software engineering process. They often turn their attention to the classic knowledge engineering methods and tools. Some prime examples of this trend are knowledge representation techniques that improve visual design of software systems, and rule-based components that help in the modelling of extensible business logic. The business community is especially in need of new design approaches. Initiatives such as business rules try to meet its requirements. Represented by the OMG, the software industry

embraces these developments with new standards such as the SBVR that supports the alignment of business rules vocabularies as well as BPMN that provides a formalised syntax for business process modelling. These are built on top of the OMG's MOF and composed into the MDA (or more recently MDE) platform.

Another important trend is the rapid development of the so-called semantic modelling tools. These are provided by the development of the semantic web vision aiming at building the next generation of the web where intelligent agents can support the user with automated reasoning. The vision successfully builds on the achievements developed in knowledge engineering research. By providing ontologies for taxonomies and classification as well as lightweight reasoning methods, it allows for the integration of semi-intelligent components into classic software systems. Semantic modelling tools also support software engineers in the design of software modules.

The knowledge engineering community also benefits from the developments in the field of software engineering. In a number of cases, new software tools and design techniques (e.g., UML) accompany or even replace classic modelling methods and programming techniques. Programming platforms and frameworks provide a friendly ecosystem to develop high-level knowledge-based applications. The recent growth in the area of semantic wikis shows, how a simple yet flexible technology can be developed into a full-fledged collaborative knowledge engineering solution.

While this synergy of software engineering and knowledge engineering methods develops, some persistent challenges remain. These are related to the effective design, automated implementation, constant maintainability, and high quality of knowledge-based software systems. For this reason, the research in the field is still active and further developing.