The development and evolution of enterprise information systems poses numerous challenges that have been researched and documented over the past forty years. Broadly speaking these challenges primarily relate to the problems that organizations continuously face when trying to deliver quality software-intensive systems in a timely and costly manner. Information systems (intended as social systems that are composed of people, technology, processes, data and so forth) are required to mirror the growth and evolution of business organizations. However, business organizations are complex adaptive systems that do not evolve in a step-wise fashion but are best thought of as ‘emergent’ entities whose features are products of continuous social negotiation and consensus building (Truex et al., 1999). Consequently there tends to be a misalignment between what is required of software systems (underlying an enterprise information system) and what these systems actually deliver at a specific point in time. Hence, the software systems themselves are required to possess a certain level of adaptation and flexibility in order to be more easily aligned to the changing organizational requirements.

One way of achieving increased adaptability and flexibility of software systems is through the adoption of Semantic-Based Systems Development (SBSD). SBSD refers to the use of ontologies (i.e. formalized conceptual models) that allow for precise semantic representations of real-world systems. Since software systems, in essence, model real world phenomena, it is necessary to adopt modeling and development techniques founded on semantics. Broadly speaking, semantics enable the precise mapping between complex real world phenomena and their modeled counterparts and enable the (dynamic) mapping/integration between different representations (and understandings) of real world phenomena.

In recent years, ontologies have emerged as the prime focus of semantic modeling, with the main focus on the development of representational languages and the resulting ontologies. Limited work has been carried out within the software development community in relation to the development of semantic-based systems—though research into the modeling, alignment and evolution of ontologies has progressed significantly among the Semantic Web community. It would seem plausible that languages, tools and techniques developed to achieve the objectives of the Semantic Web could be integrated within the software development process as a means to produce more flexible and adaptive systems. Numerous challenges exist however, as the development of semantic-based systems will have to manage things that exist (ontology), specific organizational knowledge of what exists (epistemology) and the required organizational action (pragmatics).
On October 22, 2007, the International Workshop on Semantic-Based Systems Development was held in Montreal, Canada at the ACM International Conference on Object-Oriented Programming, Systems, Languages, and Applications. This special issue contains revised versions of a selection of articles presented at the workshop.

The first article, titled *A Good Role Model for Ontologies: Collaborations* by Pradel, Henriksøn and Assmann, presents an extension of the Web Ontology Language (OWL) by introducing the notion of roles and role modeling as a way to enhance the modularization and reuse of ontologies. As compared to other approaches, the adoption of roles provides an ‘intuitive meaning’ as to why certain units (or parts of ontologies) can be used as semantic components. The approach adopted focuses on the observation that the same thing can interact with other things in different ways depending on the context (for example, a wine can assume or play the role of a drink or a product). The authors show how the adoption of roles provides an improved means of separation of concerns in ontologies as compared to a purely class-based approach.

The second article, titled *Linguistics-based Modeling Methods and Ontologies in Requirements Engineering* by Lautenbacher, Bauer, Sieber and Cabral, proposes the use of methods grounded in linguistics and Speech Acts to semantically model and annotate requirements specifications as a means of improving their understandability and reuse throughout the software development lifecycle. The authors review some linguistic-based modeling methods (such as the functional-positional segmentation method, the function design method and the Darwin Information Typing Architecture) and subsequently demonstrate how such techniques can be applied to annotate requirements specification documents in conjunction with a specialized ontology. The benefit derived from such an approach is the increased level of automation (e.g. improved querying) that can be applied to such development artifacts resulting from the improved structure and machine-understandable semantics of the specification documents.

The third article, titled *Semantic User Interfaces* by Tilly and Porkoláb, presents a new approach to developing applications based on semantic user interfaces (SUI). A SUI is a user interface that exhibits the characteristic of being semantically separate from other software components (such as those executing user requests). This approach allows domain experts to create applications by editing SUI documents (static user editable documents for describing SUIs) that can contain references to arbitrary services whose contracts are specified in domain ontologies. Component developers can then work in an application independent manner by designing and implementing reusable user interface components and service provider components that are based on standard specifications stored in domain ontologies. The approach is subsequently tested via an experimental proof of concept SUI framework. The experiments demonstrate the viability of the approach as well as its benefits. These benefits include separation of concerns between stakeholders, simplification of user interaction and increased learnability and reusability.

The fourth article, titled *Toward UML-Compliant Semantic Web Services Development* by Sánchez, Acuña, Cavero and Marcos, investigates the possibility of adopting the Object Constraint Language (OCL) for the representation of logical axioms expressed in the Web Service Modeling Ontology (WSMO). More specifically, the article focuses on a profile of the Unified Modeling Language (UML) that specializes in the modeling of Semantic Web services. The study conducted within the general area of Model Driven Development and researches the possibility of transforming WSMO models into OCL/UML models that are more readily understandable by software developers. The authors demonstrate the application of their transformation technique with three case studies. The case studies illustrate how the conversion from WSMO to OCL is straightforward providing support to developers without extensive knowledge of Semantic Web technologies to model and implement Semantic Web services.

The fifth article, titled *Integrating Web Portals with Semantic Web Services: A Case Study* by Acuña, Minoli and Marcos, investigates architectural and methodological aspects of integrating services offered by different Web Portals. The authors bring together the Web Portal Integration Architecture (PIASA) with MIDAS-S, a methodological approach for the development of Web portals. The approach proposed is based
Model-Driven Architecture (MDA). In MDA different types of models are refined in succession (from a computationally independent model to platform independent and platform specific models). The approach includes the modeling and semantic description of Web services whereby ontology context models are adopted for the different domains of application and then used to annotate the services themselves. The study is demonstrated through a case study.

The sixth article, titled *Semantics for Accurate Conflict Detection in SMoVER: Specification, Detection and Presentation by Example* by Altmaninger, Schwinger and Kotsis, investigates the adoption of semantics within the context of collaborative software development as a means to identify and resolve conflicts that arise when using Version Control Systems. A Semantically enhanced Model Version Control System (SMoVER) is presented. The conceptual design and implementation of SMoVER is described.

The system is then demonstrated for the Web Services Business Process Execution Language (WSBPEL) and examples are given in order to illustrate the semantic aspects involved in the conflict detection process. These aspects include equivalent concepts, static semantic conflicts and behavioral semantic conflicts. The possible outcomes of conflict detection are then visualized by the approach. SMoVER is then compared with related approaches presented by the previous research.

**ACKNOWLEDGMENT**

The editors of this special issue are grateful to the U.K. Royal Academy of Engineering for partially funding the organization of the workshop on Semantic-Based Systems Development at OOPSLA 2007 where the research papers described in this special issue were presented.

Sergio de Cesare holds a PhD in information systems from LUISS Guido Carli in Rome (Italy). He is currently a lecturer at Brunel University where he teaches information systems modeling and Semantic Web technologies. de Cesare’s broad research interests lay in the areas of business and software modeling, model driven information systems development and the Semantic Web. His current research focuses on the development of ontological models for systems development/re-engineering and the subsequent transformation of such models into platform-independent and platform specific application models. Sergio has authored several papers published in international journals and conference/workshop proceedings related to object-oriented, component-based and ontological modeling.

Mark Lycett is a reader in adaptive information systems at Brunel University, examining the complex dynamics that couple people, process, information and technology. His research interests relate to the principles and mechanisms of adaptation/evolution in the natural and social realm and how they can be brought to fruition using modern technologies, in particular to guide and improve people’s experience with technology. Lycett’s has published his work in a number of leading journals and conferences and is engaged in ongoing research with a number of organizations. Prior to returning to education, he spent a number of years in industry, primarily in project management.