

# Recent Papers

## Object-oriented Constraints for XML Schema

Suad Alagić, Philip A. Bernstein, and Ruchi Jairath  
Department of Computer Science,  
University of Southern Maine,  
e-mail: `alagic@usm.maine.edu`, `ruchi.jairath@maine.edu`  
and  
Microsoft Research  
e-mail: `philbe@microsoft.com`

### Abstract

This paper presents an object-oriented representation of the core structural and constraint-related features of XML Schema. The structural features are represented within the limitations of object-oriented type systems including particles (elements and groups) and type hierarchies (simple and complex types and type derivations). The applicability of the developed representation is demonstrated through a collection of complex object-oriented queries. The main novelty is that features of XML Schema that are not expressible in object-oriented type systems such as range constraints, keys and referential integrity, and type derivation by restriction are specified in an object-oriented assertion language `Spec#`. An assertion language overcomes major problems in the object-oriented/XML mismatch. It allows specification of schema integrity constraints and transactions that are required to preserve those constraints. Most importantly, `Spec#` technology comes with automatic static verification of code with respect to the specified constraints. This technology is applied in the paper to transaction verification.

Proceedings of ICOODB 2010, *Lecture Notes in Computer Science 6348* pp. 101-118.

# Verification Technology for Object-oriented/XML Transactions

Suad Alagić, Mark Royer, and David Briggs

Department of Computer Science

University of Southern Maine

Portland, ME 04104-9300

e-mail: {alagic,mroyer,briggs}@usm.maine.edu

## Abstract

Object-oriented schemas are typically lacking declarative specification of the schema integrity constraints. Object-oriented transactions are also typically missing a fundamental ACID requirement: consistency. We present a developed technology based on object-oriented assertion languages that overcomes these limitations of persistent and database object systems. This technology allows specification of object-oriented integrity constraints, their static verification and dynamic enforcement. Proof strategies that are based on static and dynamic verification techniques as they apply to verification of object-oriented transactions are presented in the paper. Most of this work has been motivated by the problems of object-oriented interfaces to XML that have not been able to express typical XML Schema constraints, database constraints in particular. The components of this technology are an object-oriented constraint language, a verification system with advanced typing and logic capabilities, predefined libraries of object-oriented specification and verification theories, and an extended virtual platform for integrating constraints into the run-time type system and their management.

Proceedings of ICOODB 2009, *Lecture Notes in Computer Science 5936*  
pp.23-40, 2010.

# Mapping XSD to OO Schemas

Suad Alagić and Philip A. Bernstein  
Microsoft Research  
One Microsoft Way, Redmond, WA, USA  
e-mail: alagic@usm.maine.edu, philbe@microsoft.com

## Abstract

This paper presents algorithms that make it possible to process XML data that conforms to XML Schema (XSD) in a mainstream object-oriented programming language. These algorithms are based on our object-oriented view of the core of XSD. The novelty of this view is that it is intellectually manageable for object-oriented programmers while still capturing the complexity of the core structural properties of XSD. This paper develops two mappings based on this view. The first one is specified by a set of rules that map a source XSD schema into its object-oriented schema. The second one maps XML instances that conform to an XSD schema to their representation as objects. In addition to mapping elements and attributes, these mappings reflect correctly the particle structures including different types of groups, and type derivation by restriction and extension. The structural properties of identity constraints are also mapped correctly. Formally defined mappings or algorithms of this sort have not been available so far, and existing industrial tools typically do not handle the level of complexity of XSD that our mappings do.

Proceedings of ICOODB 2009, *Lecture Notes in Computer Science 5936*  
pp. 149-166, 2010.

# Ambients of Persistent Concurrent Objects

Suad Alagić

Department of Computer Science  
University of Southern Maine  
Portland, Maine, USA

e:mail: `alagic@usm.maine.edu`  
and

Akinori Yonezawa

Department of Computer Science  
University of Tokyo  
Tokyo, Japan

e-mail: `yonezawa@is.s.u-tokyo.ac.jp`

## Abstract

This paper develops a typed object-oriented paradigm equipped with message-based orthogonal persistence. Messages in this paradigm are viewed as typed objects. This view leads to a hierarchy of types of messages that belong to the core of typed reflective capabilities. Unlike most persistent object-oriented models, this model is equipped with general integrity constraints that also appear as a hierarchy of types in the reflective core. A transaction is naturally viewed as a sequence of messages and it is equipped with a precondition and a postcondition. The presented framework is motivated by ambients of persistent concurrent and mobile objects. The most important practical results supporting the developed model are verification techniques and a virtual platform for constraint management.

Proceedings of DBKDA 2011 (Advances in Databases, Knowledge, and Data Applications), pp. 155-161.

# Algebraic Specification Techniques for Parametric Types with Logic-based Constraints

David Briggs and Suad Alagić  
Department of Computer Science  
University of Southern Maine  
Portland, ME 04104-9300

e-mail: {briggs@usm.maine.edu, alagic@usm.maine.edu}

## Abstract

Mainstream object-oriented languages now offer capabilities of generic types with bounded type parameters, but they typically do not provide support for specifying semantic requirements on the type parameters' methods beyond conformance of signatures. Regrettably, even object-oriented assertion languages, such as JML, have nontrivial limitations in this regard. Yet many interesting parameterized types require additional semantic features if they are to function as intended. We illustrate the issues with a case study of project scheduling based on the Project Management Institute's generic characterization of task breakdowns. We consider algebraic techniques for instantiating parametric types in such a way that the semantic requirements expressed by logic-based constraints propagate to the instantiating types. These techniques argue for more general bindings of actual type parameters for the formal ones which do not have the restrictions of current programming languages. We show that types equipped with constraints should be viewed as theories, and the bindings as morphisms of types as theories. We translate these software specifications into theories in the PVS specification language. These proposals lead to conclusions about language features for more general, semantic bindings of the actual for the formal type parameters, at least in the assertion languages.

Proceedings of the ACM Symposium on Applied Computing, SAC 2009  
pp. 1890-1897.

# Genericity in Java: Persistent and Database Systems Implications

Suad Alagić and Mark Royer  
Department of Computer Science  
University of Southern Maine  
e-mail: {alagic,mroyer@usm.maine.edu}

## Abstract

Lack of parametric polymorphism has been a major obstacle for making Java a viable database programming language. Regrettably, a recently accepted solution for genericity in Java 5.0 has far reaching negative implications for persistent and database systems because of static and dynamic type violations. Severe implications occur in typical database transactions when processing a variety of database collections. Well known approaches to persistence in Java, including Java's own persistence mechanism, do not perform correctly due to incorrect dynamic type information that gets promoted to persistence along with objects. Dynamic checking of types of objects fetched from the persistent store may now lead to unexpected type violations. Further problems occur in reflective transactions as Java Core Reflection now allows dynamic type violations without detecting them or throwing standard exceptions. All of this shows that extending Java with parametric polymorphism has not made Java a more viable database programming language. Both legacy systems such as those based on the Java binding of the ODMG or JDO, and future Java-related persistent and database technologies will be affected. The source of these problems is in an implementation idiom called type erasure. This paper provides formal proofs of the above implications of type erasure along with specific samples of code in Java 5.0 illustrating these violations. The limitations of the virtual platform and extensions required for persistent systems to solve this problem are also elaborated.

*VLDB Journal*, Vol. 17, No. 4. pp. 847-878, 2007,  
<http://dx.doi.org/10.1007/s00778-006-0042-x>.

# Reflective Constraint Management for Languages on Virtual Platforms

Mark Royer, Suad Alagić, and Dan Dillon

Department of Computer Science

University of Southern Maine

Portland, ME 04104-9300

e-mail: {alagic, mroyer}@cs.usm.maine.edu

## Abstract

Extending an object-oriented type system with assertions makes it possible for programs using reflection to rely on semantic information to ensure correct use of discovered types. Using extended reflective capabilities to access assertions in (dynamically) loaded class objects allows a variety of general and flexible verification techniques. The XVP (Extended Virtual Platform) implements these features by extending the Java Virtual Machine with the proposed functionalities. Its architecture and applications are described in the paper. One of the goals of the XVP is to provide a virtual platform that supports JML and the programming by contract methodology.

*Journal of Object Technology*, Vol. 6, No. 10, 2007,  
[http://www.jot.fm/issues/issue\\_2007\\_11/article1/index.html](http://www.jot.fm/issues/issue_2007_11/article1/index.html).

# Temporal Verification Theories for Java-like Classes

Suad Alagić, Mark Royer, and Dennis Crews  
Department of Computer Science  
University of Southern Maine  
Portland, ME 04104-9300

e-mail: {alagic,mroyer,crews}@cs.usm.maine.edu

## Abstract

We consider Java-like object types equipped with assertions as in recent proposals and implementations. The first issue that we consider is the formal notion of an object-oriented type extended with logic-based constraints along with the notions of inheritance and substitutability for such extended types. The second issue is a suitable logic for explicitly expressing properties of sequences of object states, particular cases of which appear for mutator methods in Java-related and other object-oriented assertion languages. The third issue is a suitable prover technology and the required techniques for verifying properties of object types extended with logic-based constraints. We present our solution for these problems based on the view of object types as temporal theories along with a model theory and the required verification techniques. The temporal logic-based approach makes it possible to reason about properties of sequences of object states which allows verification of behavioral subtyping requirements that are based on history properties.

Proceedings of the ECOOP FTfJP06 Workshop on  
Formal Techniques for Java-like Programs,  
<http://www.disi.unige.it/person/AnconaD/FTfJP06/>, 2006.

# An Object-oriented Core for XML Schema

Suad Alagić and Philip A. Bernstein  
Microsoft Research  
One Microsoft Way, Redmond, WA, USA  
e-mail: [alagic@usm.maine.edu](mailto:alagic@usm.maine.edu), [philbe@microsoft.com](mailto:philbe@microsoft.com)

## Abstract

This paper presents the first object-oriented interfaces that capture the essence of the structural complexity of XML Schema. We develop two such interfaces: a lightweight object-oriented interface that hides some of the complexity of XML Schema by simplifying the particle and type hierarchies, and a more complete but more complex interface that contains explicit specification of XSD groups. We define a meta level that can store the full details of XSD schemas, such as content models, type derivations for simple and complex types, and identity constraints, which is available to application programmers via reflection. The applicability of the developed interfaces is demonstrated through a collection of complex object-oriented queries.

Microsoft Research Technical Report, December 2008,  
<http://research.microsoft.com/apps/pubs/default.aspx?id=76533>