# Seventh OOPSLA Workshop on Behavioral Semantics of OO Business and System Specifications

*Doublets*

*RULES*

*1. The words given to be linked together constitute a `Doublet', the interposed words are the `Links', and the entire series a `Chain'. The object is to complete the Chain with the least possible number of Links.*
*2. Each word in the Chain must be formed from the preceding word by changing one letter in it, and one only. The substituted letter must occupy the same place, in the word so formed, which the discarded letter occupied in the preceding word, and all the other letters must retain their places.*
*3. When three or more words are given to be made into a Chain, the first and last constitute a `Doublet'. The others are called `Set Links', and must be introduced into the Chain in the order in which they are given. A Chain of this kind must not contain any word twice over.*
*4. No word is admissible as a Link unless it (or, if it be an inflection, a word from which it comes) is to be found in the following Glossary. Comparatives and superlatives of adjectives and adverbs, when regularly formed, are regarded as `inflections' of the positive form, and are not given separately, e.g., the word `new' being given, it is to be understood that `newer' and `newest' are also admissible. But nouns formed from verbs (as `reader' from `read') are not so regarded, and may not be used as Links unless they are to be found in the Glossary.*

*Lewis Carroll*

Business and system specifications are – or have to be! – technical documents used to describe and understand businesses and specifically business rules and the computer systems that have to support (some of) these rules. Specifications have to express this understanding in a clear, precise, and explicit way, in order to act as common ground between business domain experts, analysts and software developers. They also provide the basis for reuse of concepts and constructs ("patterns") common to all, or a large number of, businesses, and in doing so save intellectual effort, time and money. They introduce precision much earlier than in coding, so that business people – and not the developers – define all business rules.
An example of a set of business rules (for the *Doublets* game) was provided by Lewis Carroll and is reproduced above. These rules are clear, precise and elegant. As to conciseness, perhaps the reader can determine where these rules could have been improved.

Nevertheless, if all our business rules would have been formulated in such a manner (and appropriately structured!), perhaps our workshops would have achieved one of their most important goals.

Adequate specification approaches substantially ease the discovery of business requirements during walkthroughs with business customers, and support clear separation of concerns known since Adam Smith as division of labor. Different audiences are interested in different aspects of "common business components", and correspondingly may want to buy or sell these components based on different criteria.

Precise specification of semantics – as opposed to just signatures – is essential not only for business specifications, but also for business designs and system specifications. In particular, it is needed for appropriate handling of viewpoints which are essential for understanding large and even moderately sized systems, both business and computer ones. (800-page "flat" specifications are neither used nor read by anyone.) In order to handle the complexity of a (new or existing) large system, it must be considered, on the one hand, as a composition of separate viewpoints, and on the other hand, as an integrated whole, probably at different abstraction levels.

Many concepts and constructs used for all kinds of behavioral specifications – from business to systems – have common semantics and thus are good candidates for standardization and industry-wide usage. Various international standardization activities (such as the ISO Reference Model of Open Distributed Processing, ISO General Relationship Model, OMG activities around the semantics of UML and other OMG submissions, (common) business objects, as well as the OMG semantics and reference model working group) are at different stages of addressing these issues.

It is therefore the aim of the workshop to bring together theoreticians and practitioners to report their experience with making semantics precise (perhaps even formal), clear, concise and explicit in OO business specifications, business designs, and system specifications. Both academic (teaching!) and industrial "war stories" will be particularly appreciated. Experience in the usage of various (object-oriented) modeling approaches for these purposes would be of special interest, as would experience in explicit traceability of semantics between a business specification, business design, and a system specification.

The topics of the workshop center include:

- business specifications
- business architectures
- precise specification of semantics …
- semantics of OO modeling approaches …
- semantics-preserving refinement strategies …
- viewpoint modelling …
- standards …
- business patterns (reusable fragments of specification)
- related tool support.

In order not to start from scratch, we include here the conclusions of our previous Workshop at OOPSLA'97 (reprinted from [2]).

- Start from top
- Discover from bottom
- Precision before correctness
- Do not confuse tool use with thinking
- Properties of a complete state space lead to an invariant
- Articulation is essential:
  - "All your terms have to be 5 or 6 words long because you have to pin down the context."
  - Use ontologies including relationships other than subtyping, to ask explicit questions about context
  - Be formal, but don't insist on exposing it
- Separate business from system specifications.
  - In code, separate business from plumbing.
  - Business rules, even detailed, should not be provided by developers
- Have a bidirectional mapping between graphical and formal specifications
- Abstraction (including selection of "appropriate refinement") has to be done by humans
  - Refinement invariants should include "relevant concerns" explicitly
- Open systems change their specifications

## References


1. Haim Kilov, Bernhard Rumpe. Summary of ECOOP'97 Workshop on Precise Semantics of Object-Oriented Modeling Techniques. In: *Object-oriented technology: ECOOP'97 Workshop Reader*. Ed. by Jan Bosch and Stuart Mitchell. Springer Lecture Notes in Computer Science, vol. 1357, 1998.

2. Haim Kilov, Bernhard Rumpe, Ian Simmonds. Object-oriented Behavioral Semantics With an Emphasis on Semantics of Large OO Business Specifications. In: OOPSLA'97 Conference Addendum to the Proceedings. *ACM Press*, 1998.

3. Haim Kilov, Bernhard Rumpe. ECOOP'98 Workshop on Precise Behavioral Semantics (with an Emphasis on OO Business Specifications). To appear in: *Object-oriented technology: ECOOP'97 Workshop Reader*. Springer Lecture Notes in Computer Science, 1999.

**Organizers**

Haim Kilov
Merrill Lynch
Operations, Services and Technology
World Financial Center
South Tower
New York, NY 10080-6105
email: haim_kilov@ml.com or haim_kilov@omg.org

Bernhard Rumpe
Institut für Informatik,
Technische Universität München
80333 Munich, Germany
email: rumpe@in.tum.de

Ian Simmonds
IBM T J Watson Research Center
30 Saw Mill River Road
Hawthorne
NY 10532, USA
email: simmonds@us.ibm.com