INTERNATIONAL STANDARD

ISO/IEC 24744

First edition
2007-02-15

Software Engineering — Metamodel for Development Methodologies

Ingénierie du logiciel — Métamodèle pour les méthodologies de développement

Reference number
ISO/IEC 24744:2007(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Table of Figures

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 24744 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

Introduction

Development methodologies may be described in the context of an underpinning metamodel, but the precise mechanisms that permit them to be defined in terms of their metamodels are usually difficult to explain and do not cover all needs. For example, it is difficult to devise a practice that allows the definition of properties of the elements that compose the methodology and, at the same time, of the entities (such as work products) created when the methodology is applied. This International Standard introduces the Software Engineering Metamodel for Development Methodologies SEMDM, a comprehensive metamodel that makes use of a new approach to defining methodologies based on the concept of powertype. The SEMDM is aimed at the definition of methodologies in information-based domains, i.e. areas characterized by their intensive reliance on information management and processing, such as software, business or systems engineering. The SEMDM combines key advantages of other metamodelling approaches with none of their known drawbacks, allowing the seamless integration of process, modelling and people aspects of methodologies. Refer to Annex B where other metamodels are mapped to SEMDM and a brief synopsis of problems is provided.

Various methodologies are defined, used or implied by a growing number of standards and it is desirable that the concepts used by each methodology be harmonized. A vehicle for harmonization is the SEMDM. Conformance to this metamodel will ensure a consistent approach to defining each methodology with consistent concepts and terminology.

Software Engineering — Metamodel for Development Methodologies

1 Scope

This International Standard defines the Software Engineering Metamodel for Development Methodologies (SEMDM), which establishes a formal framework for the definition and extension of development methodologies for information-based domains (IBD), such as software, business or systems, including three major aspects: the process to follow, the work products to use and generate, and the people and tools involved.

This metamodel can serve as a formal basis for the definition and extension of any IBD development methodology and of any associated metamodel, and will be typically used by method engineers while undertaking such definition and extension tasks.

The metamodel does not rely upon nor dictate any particular approach to IBD development and is, in fact, sufficiently generic to accommodate any specific approach such as object-orientation, agent-orientation, component-based development, etc.

1.1 Purpose

This International Standard follows an approach that is minimalist in depth but very rich in width (encompassing domains that are seldom addressed by a single approach). It therefore includes only those higher-level concepts truly generic across a wide range of application areas and at a higher level of abstraction than other extant metamodels. The major aim of the SEMDM is to deliver a highly generic metamodel that does not unnecessarily constrain the resulting methodologies, while providing for the creation of rich and expressive instances.

In order to achieve this objective, the SEMDM incorporates ideas from several metamodel approaches plus some results of recent research (see [1-7] for details). This will facilitate:

- The communication between method engineers, and between method engineers and users of methodology (i.e. developers);
- The assembly of methodologies from pre-existing repositories of method fragments;
- The creation of methodology metamodels by extending the standard metamodel via the extension mechanisms provided to this effect;
- The comparison and integration of methodologies and associated metamodels; and
- The interoperability of modelling and methodology support tools.

The relation of SEMDM to some existing methodologies and metamodels is illustrated in Annex B.

1.2 Audience

Since many classes in the SEMDM represent the endeavour domain (as opposed to the methodology domain), it might look like developers enacting the methodology would be direct users of the metamodel. This is not true. Classes in the SEMDM that model endeavour-level elements serve for the method engineer to establish the structure and behaviour of the endeavour domain, and are not used directly during enactment. Only methodology elements, i.e. classes and objects created by the method engineer from the metamodel, are used by developers at the endeavour level, thus supporting both the creation of “packaged” methodologies as well as tailored, project-specific methodologies.

Here the term “method engineer” refers collectively to either a person constructing a methodology on site for a particular purpose or a person creating a "packaged" methodology as a "shrink-wrapped" process product.