

Design of an Integrated Educational Software Development Environment Oriented in Reuse and Quality Assurance of Products and Processes

*Prof.Dr.Halaris Ioan.,Petropoulos P., Geropoulos S.-T.
Department of Informatics, Technological Educational Institute of Athens
Ag. Spiridonos Str., Egaleo GR 122 10, Athens, Greece
Tel.: 0030-(0)1-5910974, Fax: 0030-(0)1-5910975,
E-mail: info@atlas.teia.ariadne-t.gr*

Abstract

In this paper, an Educational Software Development Environment capable of storing and processing reusable objects of any kind is proposed. This Environment covers the whole life-cycle (LC) of a software product, it supports different life-cycle themes, user-defined store and identification characteristics, semantic similarity mechanisms and linear and hierarchical relations among the stored objects. It also supports modelling of Life-Cycle Procedures and Products in simple and complex reuse structures, based on the needs of every-day development tasks. A successful Educational Reuse Environment is based on a powerful Reuse Database ("Repository"), on the proper education of software engineers (SEs) - in order to adopt a Reuse culture - and on the proper preparation of work environments, organisational/educational infrastructures and reuse motives.

Keywords

Software Engineering, Software Development Environment, Software Reuse, Software Engineering Education, Repository, Quality Assurance.

1 INTRODUCTION - REUSE AND QUALITY ASSURANCE PHILOSOPHY

All software developers know the benefits of software reuse: i)reduction of the overall development cost and time of a project, ii)standardisation of the development methods, iii) use of continuously improving components and iv) knowledge accumulation Mittermeir and Rossak (1990). Also it's well-known that i)the adaptation of education - methodologies, practices, tools - to new evolution of software engineering is not efficient, ii)it's essential to follow the well-established standards during software project development and iii)the achievement of Quality Assurance is a critical goal for the evaluation of the final result.

All these focus on the creation of high quality reliable software, with the minimum possible cost. The nature of Reuse requires to understand the application problems in every-day software development and to follow the guidelines in order to achieve Quality Assurance. It means that software engineers(SEs) have to: a)increase the quality of products, b)define models and quality controls for software development and it's deliverables, c) increase productivity, d) create new or adopt existing standards for the new product line (normal transition to the new approach), e) promote new developers in development process and f) refine the development process based on assessment of process metrics.

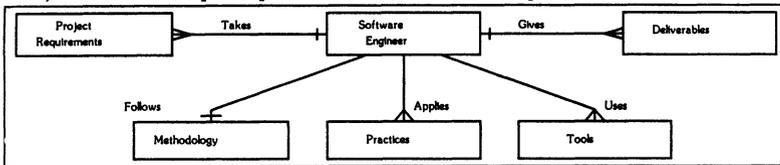


Figure 1 Software Development ERD.

(SEs) do not have to change their Methodologies, Practices and Tools (see Figure 1) but to adapt them to the Reuse Philosophy. This adaptation is effective when they search for such ready-made usable components into their projects and create new general-purpose components, usable to many projects. Also it's essential not to ignore but to give emphasis on Standards. Therefore, the addition of a new entity 'Standards' and a new relationship 'follows' into the ERD of figure1 is essential.

The basic principles of Reuse for a general-purpose Reuse Educational Environment, can be summarised to the following: a) Most-if not all-of the products within a software Life-Cycle(LC) can become reusable, as well as the procedures applied in the phases of the LC. b) There are tasks with their specific structures, phases and work-flows. Reuse has to be flexible to adapt in different kinds of life-cycles. c) Every every project belongs to an application field. Maturity Levels can be defined for every reusable object, showing the fields each object can be reused in. d) The education process in software engineering is not adapted on new development requirements. The process of "Quality Assurance" as one of the main goals in the proposed approach. For such a Reuse environment to be successful, special care has to be taken for the objects to be inserted into its Repository through procedures as domain analysis Prieto-Diaz (1987), evaluation, improvement, extension, creation etc., as well as for the SEs who will use it, through proper training, adoption of a suitable work culture and educational infrastructures for use and motivation for Reuse Tracz (1987).

2 REPOSITORY DESIGN OF THE REUSE EDUCATIONAL ENVIRONMENT

This design is based on software Life-Cycles and a hierarchical representation of the information used in them. The basic information elements are the Reusable Objects (ROs), belonging in Reusable Object Classes. Every RO belongs to a RO Class and the RO Classes are derived from the applied procedures and used products in Life-Cycles. Every applied procedure or product used in a Life-Cycle defines a used object by SEs in software development of the real world. All these objects can be modelled in the Repository as RO Classes and each category of applied procedures and used products in a Life-Cycle can be seen as a potential RO Class. The RO Classes in the Repository are grouped together, according to the Life-Cycles they are derived from. Every Methodology encloses one or more Life-Cycles. Every Life-Cycle defines a number of possible RO Classes. Each RO Class (and every RO in it) is uniquely identified by a set of Facets (characteristics or key-words). For every Facet there is a domain of Possible Values that can be assigned to the Facet and a mechanism for semantic similarities among those Values,

in order to avoid synonyms. In Figure 2, the fragments and their hierarchy of the described information that is important for software Reuse based on Life-Cycles are represented. In the following, a more detailed description of the informational structure and architecture of the Repository is presented, from the point of view of the basic subsystems. These basic subsystems can be seen in Figure 3.

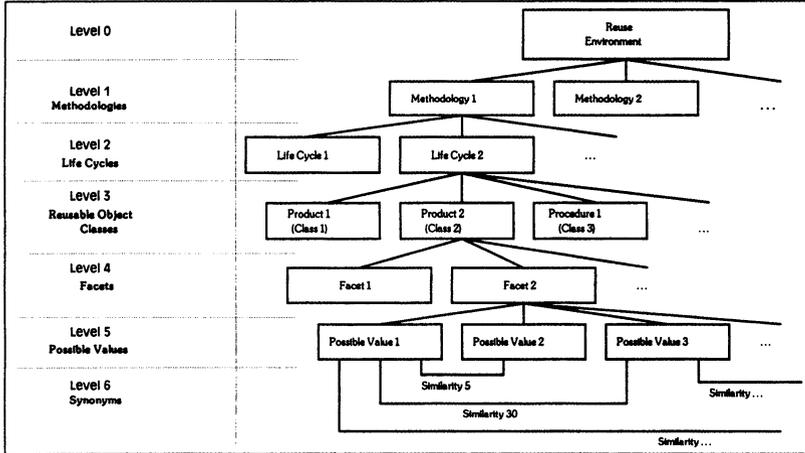


Figure 2 Information hierarchy of the Reuse Model.

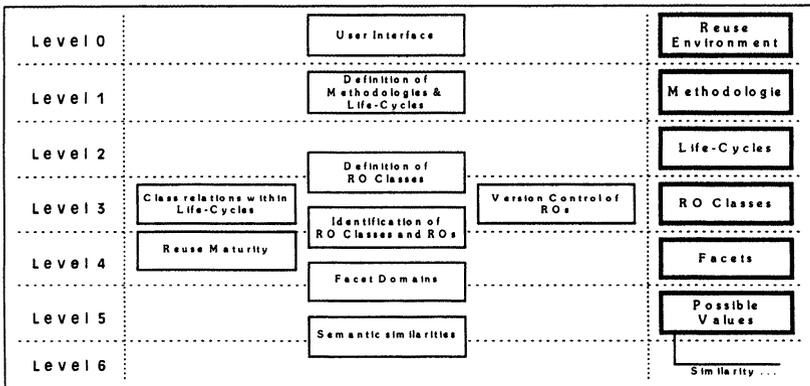


Figure 3 Information hierarchy of the Reuse Model and the basic subsystems.

3 REUSE AND QUALITY ASSURANCE PROCESS

The basic object is Methodology. Every single applied procedure or used artifact can be modelled, as a RO belonging to a Class of similar objects. In order to enlist an object to a RO Class, it has to be identified, classified to and, finally, acquired by the Class. Through time and as special needs arise for generalised and specialised ROs, new versions of the original ROs will be created and inserted in the

Repository, expanding the available ROs' database. Whenever SEs need a specific object for a project - for the phase of the Life-Cycle they are currently working on, they can have a look at the contents of the according RO Class at the Repository. Searches in the Repository are conducted according to the Facets of the RO Classes and their Possible Values in the ROs of each Class; Halaris, Ioan. and Geropoulos, S.-T. (1996).

4 PREPOSITIONS AND PROPOSALS

The proposed Repository is the technical of the two basic factors for successful Reuse. The other factor is the proper training of the SEs and a suitable organisational/infrastructure infrastructure to support it. This "infrastructure" denotes not only the proper training of SEs to adopt a reuse culture in their work, but also reuse motives and prizes for strong reuse to challenge SEs to adopt this new technology Tracz(1987), Grechenig and Biffel (1993). Specific Reuse tasks have to be assigned to specialised students. There has to be at least one SE as a Reuse Librarian, apart from the other SEs of the education department, who will be in charge of the maintenance of the Repository. It means that he/she will be controlling the creation and insertions in the Repository of new ROs, conducting searches and/or evaluating their results and creating new versions of ROs. It might take more than one SE to perform the above tasks, depending on the size of the education department and the variety of Methodologies, LCs and RO Classes in the Repository.

5 REFERENCES

- Grechenig, T. and Biffel, S. (1993) Reusing a S/w Product: What can we do in practice? in *Proceedings of the 4th Hellenic Informatics Conference*, vol. 2, Session: Information Systems III p. 577-588, Athens.
- Halaris, Ioan. and Geropoulos, S.-T. (1996) Reuse Concepts and a Reuse Support Repository in *Engineering of Computer Based Systems*, IEEE Symposium and Workshop, p. 27 - 34, Friedrichshafen, Germany.
- Mittermeir, T.R. and Rossak, W. (1990) Reusability, in *Software Engineering: Foundations and Current Perspectives* (ed. P.A.Ng and R.T.Yeh), Van Nostrand Reinhold, New York.
- Prieto-Diaz, R.(1987) Domain Analysis for Reusability in *Proceedings of COMSAC 87*, Tokyo.
- Tracz, W. (1987) Software Reuse: Motivators and inhibitors in *Proceedings of COMPCONS 87*.

6 BIOGRAPHY

Ioannis Halaris is a professor of Informatics at the Technological Educational Institute of Athens (TEIA). His primary interests are Software-Engineering Education, Development Environments, Reuse and Quality Assurance. He received a B.Sc. in Physics from Univ.of Athens and a Ph.D.(1983) in Informatics from Technical Univ. of Berlin. He has worked as Secretary General in National Center of Public Administration and in Greek Computer Society, and as Head of the Depart.of Informatics in TEIA.

Panagiotis Petropoulos is a software engineer. His primary interests are Software-Engineering Education, Development Environments, Reuse and Quality Assurance. He received a B.Sc. in Informatics from TEI of Athens. He has been working as teacher and trainer in several seminars.

Spyros-Theodoros Geropoulos is a software engineer. His primary interests are Software-Engineering Education, Development Environments, Reuse and Quality Assurance. He received a B.Sc. in Informatics from TEI of Athens. He has worked as software developer in several companies and implemented various tools for software development.