ERW'97 Session Report: Reuse Adoption Experiences Across a Large Corporation

Sergio Bandinelli

ESI – European Software Institute
Parque Tecnológico #204 Zamudio, Bizkaia, Spain
Sergio.Bandinelli@esi.es

Abstract. This paper briefly reports on the ERW'97 (European Reuse Workshop '97) by providing an overview of its sessions. The session regarding the adoption of reuse practices across a large organisation is presented in more depth. In this session, experiences and lessons learned in the ROADS project (Reuse Oriented Approach for Domain based Software) were presented and discussed. The ROADS project consisted of the realisation of four pilot experiments of reuse adoption in four different Business Units at Thomson-CSF and put domain architecture issues in the context of the overall reuse processes.

1. Introduction

The European Reuse Workshop (ERW '97) was held in the Royal Windsor Hotel (Brussels, Belgium) on 26th and 27th November 1997. It was the first of a series of workshops with the aim of sharing and promoting reuse adoption initiatives across Europe and world-wide. ERW gathered together almost 60 applied researchers, industrial software practitioners and managers involved in software process improvement and in the adoption of reuse practices.

This paper reports on some of the reuse experiences presented in the workshop. We first describe the workshop organisation and provide an overview of all the sessions. This paper, however, does not address all of the workshop sessions, but focuses the attention on one specific session regarding the application of reuse across a large organisation

This session consisted of the presentation of a series of reuse experiments carried out in Thomson-CSF within the project ROADS (a Process Improvement Experiment project partially funded by the European Commission). The project was carried out in collaboration with the European Software Institute (ESI) and Prosperity Heights Software (PHS). The author took active part in the project and thus the session report is enriched with first hand information collected during the execution of the project.

The rest of the paper is structured as follows. Section 2 provides an overview of all the ERW'97 sessions. Section 3 introduces the ROADS project, the

Frank v. d. Linden (Ed.): ARES '98, LNCS 1429, pp. 249-256, 1998.

assessment and improvement experiences. Finally, Section 4 concludes by deriving some useful lessons from the project.

2. The European Reuse Workshop (ERW'97)

At ESI (European Software Institute) we observed that there is an increasing interest in Europe in understanding and applying reuse practices and that much work was already being carried out in this field. A measure of this is the number of initiatives funded by the Commission under the ESSI programme and the interest demonstrated by ESI members in reuse technology. For this reason ESI decided to organise a European Reuse Workshop to review the state of the art and practice and to foster the interchange of reuse experiences among software practitioners.

The workshop had an excellent representation from the European industry (more than 60% of the participants), complemented with high quality representation from the academia. A significant number of European countries were represented, including Austria, Belgium, Finland, France, Germany, Italy, Spain, Sweden and UK and there was also participants from overseas (Canada and USA).

The workshop theme, "Process and architectural issues in reuse adoption" captured two key aspects of software reuse today: on the technical side, the architectures as a main reusable asset across projects and on the organisational side, the processes needed to consolidate reuse practices in a developing organisation.

Most of the workshop time was dedicated to discussion-oriented experience sessions. Each of these sessions gathered a set of position papers addressing related issues. There were six experience sessions structured as follows:

- Experience session 1: Reuse in the context of process improvement: models and current practices, chaired by Mike Mannion, Napier University.
- Experience session 2: Development for reuse: from software components to domain analysis and product family architectures, chaired by Magnus Nilsson, Ericsson.
- Experience session 3: Reuse co-ordination and experiences in a large corporation, chaired by Jean-Marc Morel, Bull S.A.
- Experience session 4: Reuse in the Information Systems domain, chaired by Bob Smith, ESI.
- Experience session 5: Reuse projects in SMEs and large companies, chaired by Jean-Marc De Baud, Fraunhofer IESE.
- Experience session 6: Reuse beyond the software development cycle: non-technical factors in reuse adoption, chaired by Sergio Bandinelli, ESI.

In addition to this, the workshop included two keynote presentations and a final panel session.

Mehdi Jazayeri, Professor of Computer Science and Head of the Distributed Systems Group at the Technical University of Vienna, gave the first keynote presentation. The title "The promises and the premises: a critical look at software reuse" anticipated a controversial presentation. Mehdi Jazayeri argued that

nowadays the emphasis on reuse is misguided, that is, reuse distracts from the real goals, promises more than can be delivered and is based on wrong premises. Finally he pointed out that reuse is not always the right thing to do. It must always be the result of a good engineering judgement and practice.

Paul Bassett, Senior Vice President of Research in CAP-Netron Inc. and author of the book "Framing Software Reuse: Lessons From the Real World", gave the second keynote presentation entitled "Ushering in the Era of Software Manufacturing". Paul Bassett defined reuse as the process of adapting generalised components to various contexts of use. In other words, he stated that "reuse" is considered at construction time, while "use" is a run time concept. Paul Bassett then presented frame technology as a way of doing adaptive reuse. The reusable frames contain commands and variables, which define the execution and construction behaviour of the frame. The frame commands guide the assemblage of the frame into source modules as in a manufacturing process. He finally presented remarkable results from projects using frame technology. This included a time-to-market reduction of 70% and a project cost reduction of 80%.

The panel session addressed a number of issues that were raised during the workshop. It was chaired by John Favaro (Intecs Sistemi) and the panelists included Colin Tully (CTA), Grady Campbell (PHS), Alexander Ran (Nokia Research Center) and Paul Bassett (CAP-Netron Inc.).

3. Reuse across a Large Corporation

Experience session 3 reported on the experience gathered in the reuse adoption experiments of ROADS project (Reuse Oriented Approach for Domain based Software). These pilot experiments were performed during 1996 and 1997 in four different Business Units of Thomson-CSF, with the collaboration of the European Software Institute (ESI) and Prosperity Heights Software (PHS). The project is partially funded by the EC as a PIE (Process Improvement Experience) under the ESSI programme.

Each of the experiences addressed a distinct domain and was motivated by different business objectives:

- The first experiment is carried out in SDS (Systèmes de Dètection de Surface) in the domain of Air Traffic Control (ATC) and has the main objective of improving time-to-market.
- The second experiment is performed at DSM (Division Systèmes de Missiles) in the domain of control and command of short range air defence systems. The most important aspect in this domain is reliability and thus the business driver here is to improve the reliability of systems.
- The third experiment is done at Thomson Training & Simulation (TT&S) Unit in the domain of training simulators. The business goal here is to obtain significant reduction of costs.
- The fourth experiment is carried out at SYSECA in the domain of Traffic Management (planning of traffic). The objective is to improve the flexibility and robustness of applications.

The methodological approach followed was common to all of the experiments and co-ordinated at the corporate level. The approach consisted in an adaptation of the reuse adoption process described in [SPC93a]. (See Figure 1).

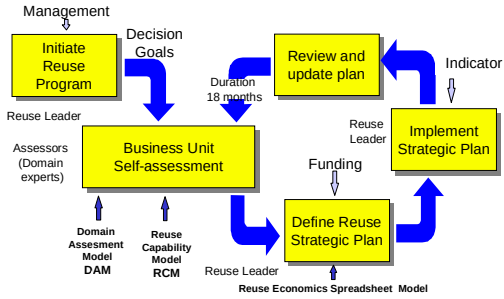

Figure 1
Reuse Adoption Process (Adapted from [SPC93a])

The initiation of the reuse programme at Thomson-CSF has its roots in process improvement. Thomson-CSF started a corporate process improvement programme in 1992 using the CMM model [SEI95] as a basis. As part of this programme, several Thomson Business Units have undertaken improvement actions to reach levels 2 and 3 of the CMM. Since the CMM does not explicitly address reuse issues, this action was complemented with the creation of a specific SIG (Special Interest Group) on reuse in 1994 and a reuse leader was appointed at the corporate level. The ROADS project was carried out in this organisational context.

The first step in the ROADS experiments consisted in the assessment of the current situation. This included a domain assessment and a reuse capability assessment in each of the business units, preceded by a training action to present the overall approach to the staff involved in the project.

The planning and implementation of the reuse actions was performed in an incremental manner. Each of the increments consisted in developing an action plan and going through the domain engineering activities as defined in the Reuse-Driven Software Processes (RSP) [SPC93b]. The documents produced during the increment were reviewed at the end of the increment and the feedback was used to plan the subsequent increment. The duration of each increment was quite short (typically around 3 months) to ensure a fast feedback loop.

3.1 Assessment Experiences

Two types of assessment were conducted at the beginning of the ROADS experiments: a reuse capability assessment to characterise the state of practice as far as reuse is concerned and a domain assessment, to measure the reuse potential of the domain. The main objective of the assessments was to guide the planning of reuse adoption by helping to identify the priorities for each Business Unit.

The assessment teams included persons belonging to the unit being assessed (i.e., self-assessments) plus a facilitator with the role of introducing the assessment model and manage the meeting. The assessment team size ranged from three to eight individuals. The typical duration of each of the assessments was one day.

The reuse capability assessments used the Reuse Capability Model (RCM) [SPC93a]. This model consists of a set of goals grouped in four reuse capability levels: Opportunistic, Integrated, Leveraged and Anticipating. The assessment process rates the extent to which the organisation meets each of these goals.

The domain assessments examine the domains from a business perspective to provide an indication of the potential for profitability in applying reuse. The assessment model used, called Domain Assessment Model (DAM) [SPC93a] consists of five factors, namely market potential for products, existing domain assets, commonalities and variabilities between systems in the domain, domain stability and maturity and domain standards. Each of the factors is rated in a 1 to 5 scale and the results are plotted in a Kiviat diagram to appreciate the relative strength of each of them.

3.2 Improvement Planning and Implementation

At writing time, five increments have already been performed in each of the ROADS experiments. The incremental nature of the adoption process makes it possible to start obtaining results very early in the reuse adoption process. This is fundamental to keep the process on the right track and to demonstrate (to management and to the practitioners working in application projects) the benefits of the approach by providing tangible results and benefits.

The typical plan for an increment includes the following items:

- *Domain definition*, including a glossary and domain communalities and variabilities.
- *Decision model*, a formalisation of the variabilities of the domain including the range of variability.
- *Product family engineering*, development of configurable and adaptable work-products of all kinds (including requirements documents, domain architectures design, code test, project plans, contracts, etc.)
- *Process engineering*, discussion on the changes to be introduced in the current application development process
- *Domain strategy and planing* of the subsequent increment.

Generally speaking, all the experiments completed successfully the domain definition and most of the decision model (at least for some significant sub-domain). Regarding product family engineering, each of the experiments concentrated the efforts in those work-products that could maximise the return on investment. This depends, among other factors, on the nature of the domain and on the stage of development of application projects. Finally, process engineering was the activity that most stretched people's ability, since it required to identify the changes in the current practice to incorporate reuse.

To illustrate the kind of problems faced in the project, we provide a couple of examples regarding two of the four experiments: one corresponds to the

experiment in the domain of Air Traffic Control (ATC) and the other to the one in the domain of training simulators.

Air Traffic Control domain

The Air Traffic Control group has a line that develops small systems for control centres in airports. These systems have been delivered world-wide to more than 15 Civil Aviation Authorities of various countries, including Denmark, Mexico, Bulgaria, various ex-SSSR republics, etc.).

Since 1992 the international competition has become stronger. This motivated an investment in architecture and the establishment of an incremental and modular development approach. In other words, a "baseline" product is incrementally enriched with new functionality, as required by clients. The additional functionality represents a small part of the code since most of it is reused from previous applications (see Figure 2).

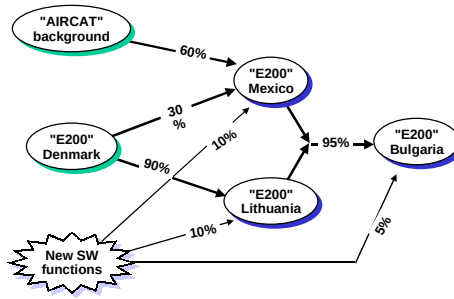

Figure 2
Evolution of the system and reuse rates

This advanced situation regarding reuse in this Business Unit was reflected in the results of the reuse capability assessment, which showed that several of the Leveraged level goals were meet to a great extend. Not surprisingly, the first benefits obtained from the ROADS project in this domain were not in the reuse in coding phase, but on other phases of the development process. For example:

- the decision model was started to be used in the Bid-NoBid phase (to decide whether a contract is within the boundaries of the domain),
- the baseline product was better defined and some documents, such software specification and
- software development plan, were standardised to allow for automatic generation.

Training Simulators domain

We have concentrated the attention on the driver trainer domain. The reuse capability assessment showed that the organisation achieves most of the

Opportunistic level goals and some of the Integrated level ones. This organisation has been recently assessed at level 3 of the CMM and the domain assessment showed that there is good reuse potential. However, the reuse practices are unplanned and based on code cut&paste.

Figure 3 shows that the effort associated with the different application projects in the domain tends to decrease as the number of projects increases. However, a new functional definition of the domain creates a break, making the subsequent project (the first after the break) much more expensive.

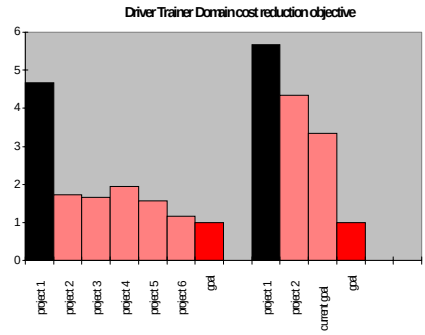

Figure 3
Cost of project application in domain trainer domain

In this context, the ROADS project served to document the existing application projects (according to commonalities and variabilities). This had al least two tangible benefits. On one side, it provided the basis for better anticipating the impact of functional breaks. On the other side, it served as a tool to create awareness about the current existing functional capabilities in the domain. This latter issue is especially useful among commercial/business staff to reduce costs when negotiating new applications with customers.

4. Conclusions and Lessons Learned

There are a few lessons that can be derived from the experience accumulated so far in the project. These lessons are general enough to be useful for other similar experiences.

A first observation is that the participation of domain experts in all the activities is essential for the success of the experiments. However, domain experts tend to be overloaded and it may be difficult to involve them in the key activities.

Regarding assessment experience it is important to point out that there is not necessarily a direct relationship between process maturity (e.g., in terms of CMM levels) and reuse capability. The latter depends on other factors such as the experience the organisation has in a domain, the level of standardisation in the domain, etc. We can conclude that

- The assessments served as a means for reaching a common understanding within each of the units to determine the strengths of the organisation and the priorities for improvement.
- It was unclear which improvement actions would address the unaccomplished goals, making it difficult to plan improvements and assess progress.

The experiments showed that the architecture plays a central role in determining the existence of a product-line in a given domain. For example, without a fixed agreed architecture it would be impossible to do incremental developments in the air traffic control domain. However, the same variations and flexibility that is necessary for a domain architecture are required for all other work-products. By allowing this variability all the phases of system development can benefit from reuse.

5. References

[SEI95] Software Engineering Institute - Carnegie Mellon University, *The Capability Maturity Model - Guidelines for Improving the Software Process*, SEI Series in Software Engineering, Addison-Wesley, 1995.

[SPC93a] Software Productivity Consortium, *Reuse Adoption Guidebook*, SPC-92051-CMC, November 1993.

[SPC93b] Software Productivity Consortium, *Reuse-driven Software Processes*, November 1993.