Learning styles and CASE tools in Software Engineering

Lynne Fowler, Maurice Allen, Jocelyn Armarego and Judith Mackenzie
School of Engineering
Murdoch University

Software Engineering is a new discipline aimed at the improvement of the production of large, quality software systems. Interest in the adoption of CASE tools has escalated because of the important role they play in supporting the software development process. However, these automated tools are sophisticated and complex. While studies show that CASE tools have a positive impact on quality and productivity, it is also shown that they have been slow to be adopted by industry. This phenomenon is partially explained by the effort in learning to use the tool. Ultimately, knowing the factors that favourably influence the rate of learning will lead to improved approaches to teaching software packages, particularly CASE tools and hence the uptake of these tools within industry. The correlation between learning styles and our methods of teaching CASE tools must be established to investigate where conflicts exist.

This paper discusses an initial study examining the learning styles of engineering students, based on the work of both Kolb and Soloman. This will then be used as a foundation for comparison between student learning styles and use of our CASE tool, Rational Rose. Monitoring online use will be achieved by automated tracing of student navigation within the package.

Our results will be used as a basis to develop an online learning methodology whereby learner characteristics can be used to establish the environment to support the construction of knowledge in our students.

Introduction

Education and learning are ongoing and dynamic. As such, our teaching and learning styles and methodologies must be continually reviewed to respond to developments in technology and to the changing demands from society. In universities, as well as in other educational institutions, online computer resources are seen as essential.

Software Engineering is a relatively new discipline grown from rapidly expanding developments in IT and computing. CASE tools are now an important area in Software Engineering. The complete Software Development Life Cycle needs support as software systems are increasingly more complex, large and often critical. Software costs each year are ever increasing and modest improvements in productivity of software would mean significant savings [Fuggetta et al 1993].

CASE tools are aimed to address the difficulties of developing high quality, complex software on time and to budget. Consequently, CASE was thought to provide the solution and these tools were hailed as the answer to the "software crisis" [Pressman 1997]. However, CASE has not been the panacea promised by earlier hype [Gabel 1994]. Most studies show that CASE tools do positively impact quality and productivity [Iivari 1996] despite having been slow to be adopted by industry [Holt 1997; Dutta and Wassenhove 1999]. Statistics show [Kermerer 1992] after one year of introduction that:

70% of case tools are never used
25% are used by only one group
5% are widely used but not to their full capacity

Cited reasons suggested for CASE tools not having been adopted include:

- CASE tools are complex and difficult
- Case tools require a steep learning curve
- Training on tools is often not available
- Training on tools is often not sought by companies
- Managerial attitudes to the use of tools needs to change
- 50% of industry do not follow a well defined process or methodology
- CASE tools are a sharp step forward for many organisations
- Case tools need a large investment in time, cost and effort
- Organisations often use "in house" techniques and CASE tools are not easily modified to individual requirements
- CASE tools are generally expensive.
- Tools do not perform as well as expected
- Tools do not cover enough of the SDLC

Evidence shows that only 20 to 50% of organisations are routine CASE users [Holt 1997; Dutta and Wassenhove 1999]. Effective use of CASE tools implies the adoption, within organisations, of a methodological approach to their software production. Amazingly, large numbers of organisations are still not using any methodology [Holt 1997]. Studies show [Holt 1997] the following percentages for usage of design methods and methodologies.

21% Object oriented
48% Structured
0% Formal
31% None

International standards for software production have brought great interest in models for standardising software production. Use of models such as Capability Maturity Model, SPICE, TickIt have now attracted much interest. Organisations need to use recognised design methodologies and supporting tools, to achieve these international standards. The choice of CASE tool is therefore critical [Eriksen and Stage 1998].

**Direction of research**

The aim of this research is to use a CASE tool to investigate different aspects of learning, particularly with respect to online learning.

**Background**

The School of Engineering, at Murdoch University, has made the decision to offer most courses online based on currently accepted constructivist models of learning [Phillips 1995; Allen et al 1998]. The Web is seen as a medium that supports student control of the learning process through its capacity to help learners develop unique knowledge representations [Miller and Miller 1999].

In addition to the Web, online packages are made available to students as tools to support their experiential learning. CASE tools fall into this category of resource: they provide a model of a concept space (such as a very simplified version of a real world environment) which students manipulate. Students are able to create constructions within the microworld, the term coined by Papert [1993] with his development of the Logo environment, that will behave in some way consistent with the concepts being modelled. The interactivity inherent in the microworlds allows learners to see immediate results as they create models or try out their theories [Rieber 1994]. This immediate feedback has greater effectiveness in
a constructivist environment as it enables the student to alter the way information will be encoded. Learning is further enhanced where contextual help (a standard feature with these online tools) and explanations are linked to multiple attempts at model construction.

**Pilot study**

The initial concept of this research was based on a small but significant investigative study to examine the learning issues surrounding a CASE tool [Fowler and Allen 1999]. These results have now guided us into investigating learning styles and the development of online statistical results to precisely monitor use of the CASE tool and navigation within the tool.

**Learning issues**

The first phase is to examine learning issues confronting our software engineering students when using CASE tools. The research will then extend into industry where the particular CASE tool is being used. As the emphasis is on learning styles and teaching methods rather than on student assessment, this study will aid the acceptance of CASE tools, therefore bridging the learning gap that these complex tools present.

Whilst there are numerous instruments for assessing learning styles, those advocated by Kolb [1984], Solman and Felder [1999] are well known, and accepted within education theory [Montgomery 1995]. Both instruments provide an efficient way to analyse our students' learning styles. The correlation between learning styles and our methods of teaching CASE tools must be determined. Often, conflicts exist between our styles of learning and teaching methods. Students whose learning styles are compatible with the teaching style of a course instructor tend to retain information better, obtain better grades and maintain a greater interest in the course [Felder 1993].

Engineering teaching professionals are quoted as being intuitive learners [Felder and Silverman 1988]. This was confirmed when each staff member of our department was tested on the Kolb learning style model, revealing that 70% of academic staff were converger/assimilators (how and what people). The evidence suggests that engineering students are more sensors, accomodator/diverger types [Felder and Silverman 1988].

**Women in engineering**

Gender learning issues are also being included in the investigation. The question still to be answered is: Why are more women are not attracted to Engineering in greater numbers?

Software Engineering is suitable for both male and female engineers and should be an attractive area for both to excel. However, student numbers on courses show a very small percentage of females, similar to statistics for numbers of female engineers in the workforce [Deetya 1996]. Two aspects of gender will be investigated; firstly gender and online learning, and secondly gender and the use of CASE tools. Results will indicate if gender is an issue; and whether it has a positive or negative impact on Software Engineering and CASE tool learning and usage. Experience with our students shows that female students have spent less time playing computer games than their male counterparts, and that, when entering University, they will have had less computer experience. In addition, engineering culture is cited as not being female friendly [McLean et al 1997; Hawks and Spade 1998; Copeland and Lewis 1998]. By comparing differences in learning and examining cultural issues within Engineering, we will be able to advise and encourage women into appropriate areas to suit their skills.

**Industrial needs**

Our students also need to be prepared for when they enter the workforce and will no longer have instructors to rely upon. They will only have themselves and their colleagues [Felder 1998]. It is crucial to
help them be independent. Many software packages and tools, are initially self taught or assessed and they are often perceived to be complex. This adds to the difficulty of learning CASE tools.

The extrapolation of our research into industry will be beneficial in the struggle with complex software packages.

**Learning environment**

The tool may channel the learning environment and impose a certain methodology. Therefore, you may have to teach the methodology before addressing learning issues related to the tool.

In addition, our students need a flexible and adaptable approach to training in the fast changing world of information technology to foster life long learning.

**CASE tool selection**

The CASE tool chosen for this study needs to exhibit certain characteristics to produce valid results that will be of value to education as well as industry:

- Support for a large part of the Software Development Life Cycle
- Flexibility
- Intelligence in the tool, not just a replacement of pen and paper
- Support for re-use
- A professional CASE tool useful for industry
- Support for the Object Oriented paradigm that we teach and that industry is using on new projects

*Rational Rose* has been chosen as best fulfilling the above criteria. Support for Object Oriented development through Use-Case Views, Sequence Diagrams, Class Modeling and Component Views are provided, hence supporting the development paradigm we teach. Outline skeleton code can be generated in several languages including Java and C++. We have started using UML (the notation supported by *Rational Rose*) as a vehicle to teaching Requirements Engineering and Software Design from an Object Modelling perspective using *Rational Rose* as our support tool. In addition, our third year students have a good background in both C++ and Java.

**Learning styles**

Learning is a process of acquiring and remembering ideas and concepts. The process not only involves getting information but also full participation by the learner. No longer are the traditional roles of teacher/student: teacher giving, student accepting, considered the only way to learn or even the best way [Kolb 1984]. Learning is also experience based and often related to the personal application required. By examining learning styles we hope to be able to address learning issues involved in CASE tools.

The two learning styles that we are using are by Kolb [1984], Soloman and Felder [1999].

**Learning Styles Inventory**

Kolb views the learning process, as a four stage cycle: concrete experience followed by observation and reflection, which leads to the formation of abstract concepts and generalisations, which leads to hypotheses. The hypothesis can then be tested leading to new experiences and the cycle continuing (Figure 1).
The Learning Style Inventory is a simple test based on experiential learning theory. It looks at four stages of the learning process: concrete experience (CE), reflective observation (RO), abstract conceptualisation (AC), and active experimentation (AE). A series of twelve questions are used and the user has to rank four possible answers for each question.

**e.g. When I learn:**

| 2 | I like to deal with my feelings |
| 4 | I like to watch and listen |
| 3 | I like to think about ideas |
| 1 | I like to be doing things |

Combination scores indicate your emphasis of abstractness over concreteness (AC-CE) and experimentation over reflectiveness (AE-RO). Your preferred learning style is indicated by plotting the combination scores on a grid (Figure 2).

The four learning styles are [Burns 1989]:

- **Accommodator: What if?** people. Often start with what they see and feel then plunge in and seek hidden possibilities. They learn by trial and error and self discovery.

- **Diverger: Why or why not?** These people study life as it is and reflect on it to seek meaning. They learn by being involved and need to listen and share with others.

- **Converger: How?** These people start with an idea and try it out, they like to find out how things work and learn by testing theories.

- **Assimilator: What?** people. These people come up with ideas and then reflect on them. They like to know what the experts think.
A problem experienced on administering this survey was related to scoring of the answers. Very clear direction was required to explain that a score of four indicated the most preferred choice and one the least.

**Index of Learning Styles**

The Index of Learning Styles [Felder and Soloman 1999] is an instrument to assess preferences on four dimensions; active/reflective, sensing/intuitive, visual/verbal, and sequential/global. This instrument consists of forty four simple questions with a choice between two possible answers.

  e.g. I understand something better after I: a)try it out b)think it through

The results are then scored on the four measures [Felder and Soloman 1999]:

- **Active/Reflective:**
  Active learners like to try things out and see how they work and like to work with others. Reflective learners like to think things through first.

- **Sensing/Intuitive:**
  Sensors like to learn facts, use well established methods and practical and careful. Intuitors tend to work fast and be innovative and can often handle abstract and mathematical concepts well.

- **Visual/Verbal:**
  Visual learners like diagrams, pictures, graphs and films. Verbal learners get more out of words heard and written.

- **Sequential/Global:**
  Sequential learners like to work in linear steps that follow logically. Global learners like to jump in, absorb material nearly at random and then get the big picture.
Results of learning styles surveys

Learning Style Inventory - Kolb

The results of students from the Learning Style Inventory are shown in Table 1. They show a similar split for students between active and reflective as the Soloman and Felder survey, Table 2. However, the balance between abstract and concrete learners shows a more equal distribution.

Table 1: Learning Style Inventory Results - Students

<table>
<thead>
<tr>
<th>Accomodator</th>
<th>Diverger</th>
<th>Assimilator</th>
<th>Converger</th>
</tr>
</thead>
<tbody>
<tr>
<td>9%</td>
<td>39%</td>
<td>17%</td>
<td>35%</td>
</tr>
</tbody>
</table>

Abstract

Converger + Assimilator = 52%
Acomodator + Diverger = 48%

Concrete

Active

Accomodator + Converger = 44%

Reflective

Diverger + Assimilator = 56%

Total number of students = 23

Index of Learning Styles - Soloman and Felder

The comparison of student and staff results from the Index of Learning Style survey are shown in Table 2 and Table 3.

Table 2: Student Index of Learning Style Survey Results

<table>
<thead>
<tr>
<th>Processing</th>
<th>Perception</th>
<th>Input</th>
<th>Understanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>57% Sensory</td>
<td>70% Visual</td>
<td>83% Sequential</td>
</tr>
<tr>
<td>Reflective</td>
<td>43% Intuitive</td>
<td>30% Verbal</td>
<td>17% Global</td>
</tr>
</tbody>
</table>

Total number of students = 23

Table 3: Staff Index of Learning Style Survey Results

<table>
<thead>
<tr>
<th>Processing</th>
<th>Perception</th>
<th>Input</th>
<th>Understanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>12.5% Sensory</td>
<td>37.5% Visual</td>
<td>75% Sequential</td>
</tr>
<tr>
<td>Reflective</td>
<td>87.5% Intuitive</td>
<td>62.5% Verbal</td>
<td>25% Global</td>
</tr>
</tbody>
</table>

Total number of staff = 8

The results from Table 2 show that:

- 57% of the students learn best actively, yet our teaching is often passive;
- 70% of the students are sensors, yet we often teach intuitively;
- 83% of the students are visual, yet a lot a our material is presented to them verbally or in written form;
- 39% of the students are global learners, yet our teaching is often sequential and does not focus on
Our results for students are similar to those of Mackenzie, [1998] who surveyed 75 Mechanical Engineering students. Our students are showing a greater tendency to sensory (70% compared with 56% from Mackenzie's studies).

Soloman [1999] has surveyed large volumes of students via her online site and her results show that:

- 80% of all students are active learners
- 55% are sensors and 60% for engineers
- 75% are visual learners
- 60% are sequential learners

Our students showed a more even split between active and reflective learners and a greater tendency to sensory learning. We are aware of the small sample size in this initial study. However, the surveying of students is continuing and an increase in data will enhance this research.

Most engineering academics are intuitors, but the majority of engineering students are sensors [Felder 1988]; our results indicate a similar outcome, Table 3. Consequently, if staff teach according to their learning styles then a mismatch will exist with the average student.

**Application to CASE tools**

**Application to our current teaching practice**

Currently, we introduce *Rational Rose* by means of an online tutorial. A nine step exercise, taking approximately two hours to complete, uses all stages of the package to finally generate skeleton code for a small program. Our results show that a large percentage of our students are active learners and hence using practical hands on sessions to teach the package directly supports their learning style. This hands on approach also supports sensory learners. However, 39% of our students are global learners and we are directing them to work sequentially. We aim to provide a 30 minute quiz which will necessitate students searching through the package and therefore obtaining a global view of the software's capability. This approach should also appeal to the active learner who is keen to "see how it works". The package is particularly visual with its adherence to the UML modelling language and associated graphical icons.

Application of Felder's idea - stopping several times in a teaching session to pause for thought and pose a short question for discussion - would provide support for reflective learners [Felder 1988]. This technique, in the context of the online exercise, would assist in breaking up the long two hour session, making the experience more stimulating and rewarding.

Another suggestion [Felder 1993] is to talk to students about their learning styles and the strengths and weaknesses associated with each style. We propose to incorporate a topic into our first year Foundation Unit to survey and discuss student learning styles.

**Future phases**

The next phase of our research is to develop a means of monitoring patterns of movement within our chosen CASE tool by coded scripts being embedded into *Rational Rose*. The online log will be analysed at the end of each session. Movement within the tool will be related to the students learning style.

**Conclusion**

The application of such results will prove to be positive in the following areas:

- Improved approaches to teaching of CASE tools.
Transferable skills to other software tools and packages.
Better preparation of our students for industry in their approach to learning new software tools.
Training of industrial professionals in the use of CASE tools
Positive encouragement for the uptake of CASE tools by industry.

Staff and students need to be aware of their learning styles and for staff to apply this knowledge to their teaching of CASE tools. This research sets the foundation for future work, to develop an online learning methodology, whereby learner characteristics can be used to establish an environment to support the construction of knowledge in our students.

Further research into the learning issues and CASE tools will be valuable and innovative.

References


---