Transition to OO? - A perspective from ATLAS

A. (RD) Schaffer

LAL, Orsay, France

1 Where we are today in ATLAS

With the submission of the ATLAS Computing Technical Proposal in December 1996, ATLAS has accepted the object-oriented paradigm for the implementation of its software. In terms of object-oriented experience, those members of ATLAS who participated in RD41/MOOSE have "made the transition" by working on various prototypes. One of the key challenges today is to "make the transition" for the larger ATLAS community and to involve them in the development of the large-scale projects required for ATLAS. Note that the intense effort to design the detector - a Fortran-based software effort - is coming to an end. People are now looking for "practical ways" to make this transition.

In trying to outline how we foresee attacking this problem of transition, I will try to address the questions posed by the convenors:

- Choice of method and CASE tool:
  What are the reasons for the choices made?
  What were the expectations, which came true?
- Definition of the software process:
  To what extent has it been defined explicitly?
  Does it include formal review procedures?
- Education and training:
  Is everybody educated to the same level?
  Are there methodologists and programmers?
- Are training courses or external consultants available?

2 Method and CASE tools

ATLAS has not adopted, yet, a standard "method" - i.e. a step by step procedure for the micro-cycle. Rather, we have chosen a standard method notation: Unified Method Language (UML). Currently we are using OMT class diagrams and message flow diagrams until UML becomes available. And we have defined what should be the "deliverables", for example, for a design cycle within the context of a software process (see later). The method used to arrive at these deliverables is left up to the group working on a particular domain.

As a CASE tool, we has adopted Software through Pictures (StP). The primary motivation for this selection was:

- Supports full developments cycle: i.e. requirements, design diagrams, code
- Can be customized to our needs
- Allows reverse engineering of code to allow code to be developed outside of the tool

In terms of expectations and conclusions, we have found that CASE tools are good for expressing designs, although perhaps not all elements of a method may be supported by a tool. A CASE tool is most likely NOT the best medium for code development. However, it is most likely the best means of maintaining a coherent picture of a complete software development, i.e. requirements, design, and code.

3 Software Process

Its has become clear to us in ATLAS, and to others elsewhere in the software community, that a key to success is to be able to control and manage the software the software effort, which is admittedly difficult in our domain. We feel that an important part lies in defining and following a software process. This should allow:

- the coordination of the efforts of the development team
- improving estimates of time-scales and resource requirements

A software process has been defined for ATLAS. It consists of a cyclical process:

- Cycles of 8 working weeks
- Incremental delivery
- Evolutionary deliverables (requirements, design, code)
- Reviews

There is also the underlying concept of decomposition into domains which reflects the problem as well as the human responsibilities.

The status for ATLAS software is that we have done a first global domain decomposition, have a first set of requirements, and have completed a review of the requirements. Within MOOSE we developed the tools to perform reviews electronically. We have successfully used them for both requirements and code reviews, but not yet for design review.

Our experience and conclusions is that the process is slow to get started and needs to be adapted as we go along. The deliverables and their review hold the process together. We have found that electronic reviews work. And both forward and reverse engineering require tool support.

4 Training

For ATLAS we propose a layered training program according to the level of involvement:

- For all those who are involved in computing:
  - Introduction to software engineering
  - C++ for particle physicists
- For developers of the more general packages:
  - OO Analysis and design
  - Hands-on OO Programming in C++ or Java
- Specialized courses:
  - Advance C++
  - OO Database for developers
  - CASE tool

In response to the question of whether there are methodologists and programmers, my personal opinion is that there will always be a spectrum in people's capabilities. The key point here is to clearly define the organizational structure of a project. For example, in the ATLAS Software Process we define the roles of a chief architect and the domain architects which together form a core team. It is their responsibility to define:

- the decomposition of the problem into domains and
- the domain interfaces which defines the overall architecture

This requires a higher level of competence than for those working on the internal aspects of the individual domains.

Within MOOSE, we have gone through many of these courses. The ones we have found useful are now part of the CERN training program. Some courses have been organised outside CERN, cf. Paul Kunz's course, as well as an OOA&D course which included some BaBar people. External consultants have been used to a certain extent. However, more important is "internal consultation":

- internal training sessions on specific physics-oriented problems
- mentoring
- reviews

5 Summary and conclusions

ATLAS has chosen to follow the object-oriented path and we are currently in the early stages of turning this into practice. This will entail a continuous process of learning and gaining experience as we build the software needed. It is important to consider the concerns of the wider physics community which will need to be trained to different levels to be able to work in this new world. A important element which we feel will hold this together is a well understood software process. An initial definition of this has been made for ATLAS, and we are currently breathing life into it.