

Integrated Tools For Experimental Physics

An integrated environment for computer-based applications aims to ultimately support all the phases of a production process without requiring the conversion of information between phases. It consists of many different tools and provides mechanisms for tool integration. Each tool encapsulates not only an engineer's productivity but, more profoundly, embraces a methodology: a good tool uses a metaphor to implement the methodology in a compelling way [1].

Four broad categories of integrated environments can be envisaged [2]: *language centred* to comprehensively support programming in specific languages; *structure oriented* to manipulate structural objects (e.g. data sets, software modules, files) directly and independently of specific languages; *method-based* tools to support a particular set of processes or methods; *tool-kits* (collections of tools which are usually language independent).

The support offered by the first three environments shows many similarities. The main differences stem from the basis of the (methodological) approach for providing support, e.g. many of the CASE tools developed over the past five years are included in the method-based environment. Tool-kits, however, consist of collections of tools that are preferably developed and supported by more than one organization. They as a result help to extend, tailor and port an implemented software program because you can add or modify tools, and because you can change the tool configuration as technology evolves. Moreover, one is not dependent on a single vendor for service if tools come from different vendors.

Implementation

The effective operation on objects created at different times with different tools has to face the same problems inherent in all environments, namely consistent user interfaces, data interchange, data integration and portability.

The X Window protocol is an example of a consistent user interface and a wide varie-

ty of computers and workstations apply this windowing standard. Open Software Foundation's Motif is a specific solution based on X Window. However, other products (e.g. Sunview, IBM Presentation Manager) focusing on consistent user interfaces across tools enforce a single proprietary solution.

The common format for data interchange between tools is the most basic form of *exchange* (e.g. importing or exporting files between data dictionaries). However, to avoid loss of information, the meaning of objects must be combined with the defining semantics (e.g. the description of a dataset).

Data integration requires that similar objects created by different tools have equal semantic contents. Simplification, applicability and portability result as advantages in the tool-kit environment, where several tools operate on objects created by different tools. Disadvantages include having to work with the lowest common definition of data objects, although this can be overcome using a common repository that maintains a core semantic content for objects as well as tool-specific views. With its common semantic dictionary, the repository lets tools work together in the same fashion as a dataway for hardware modules.

The handling of different objects in different environments is required for *portability*. It is being enabled more and more by concentrating effort on common standards which are independent of hardware and operating systems (e.g. the Portable Common Tool Environment created by a consortium of European computer manufacturers).

Today's trend in tool integration is definitely towards greater portability and "interoperability" to make implementation and maintenance less critical, by using a more formal and methodological approach to establish a stronger coupling between the analysis of requirements and the final implementation. Surely the same should apply to tools for process control environments in physics?

Community Response

A workshop to define and specify a tool-kit constituting an integral part of the Los Alamos ground test accelerator (GTA), a prototype of a fully automated, space-based accelerator, revealed in 1988 that:

- When getting to details, the community did not use common semantics and hence had no common understanding of the objects under discussion;
- One could learn from the wide variety of tools offered by the software industry (ranging from programming and debugging support to highly integrated workbenches for project management);
- The computer-aided engineering of software tools for accelerator operation should be done to make them portable.

It was proposed to form working groups to develop tool-kit specifications but, somewhat amazingly, no significant progress is visible. However, the controls community has undoubtedly recognized the impact and the challenge of tool integration on all phases of a control project's life cycle. It has understood that tool integration involves the complex task of letting different tools used in different phases operate together to produce the total product. It has been wise enough to carefully plan tool implementation in small but well-defined pilot projects by first studying the underlying methodologies, and by developing standards for using the tools and for assessing and controlling the quality of a tool's output. It has accepted that introducing an integrated tool strategy involves a major financial commitment at the onset so it is crucial to have the active support of top management.

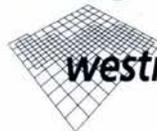
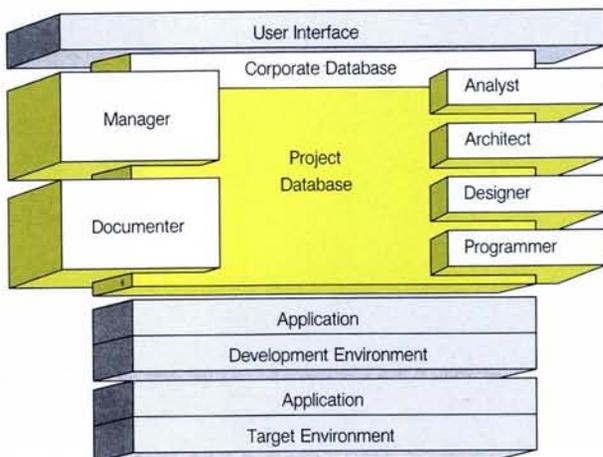
REFERENCES

- [1] Oman P.W. *et al.*, *IEEE Computer* (May 1990) 15; *IEEE Software* (The Tools Fair, May 1990) 20.
- [2] Dart S.A. *et al.*, *Computer* (11/1987) 18.

W. Busse

Hahn-Meitner-Institut, Berlin

I-CASE: increased quality and productivity



westmount - integrated CASE Tools

- Westmount Technology's CASE Tools are based on relational database systems and industry standards like UNIX, VMS, SQL, X Window, OSF/Motif.
- TSEE/RTEE implements well-known engineering methods and supports the complete life-cycle.
- Version and configuration management for networked development environments.

Professional Tools for Software Engineers:



Computer Software Beratung Täfernstrasse 22a
5405 Baden-Dättwil
Tel. 056 / 83 25 35
Fax 056 / 83 39 05