Software Engineering Institute

An Investigation into the State of the Practice of CASE Tool Integration

Jock Rader
Alan W. Brown
Edwin J. Morris

August 1993

TECHNICAL REPORT
CMU/SEI-93-TR-015

http://www.sei.cmu.edu

Carnegie Mellon

Technical Report
CMU/SEI-93-TR-15
ESC-TR-93-192
August 1993

An Investigation into the State of the Practice of CASE Tool Integration

Jock Rader
Alan W. Brown
Ed Morris

CASE Environments Project

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This technical report was prepared for the

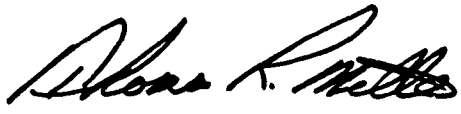
SEI Joint Program Office
ESC/ENS
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER



Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

The Software Engineering Institute is sponsored by the U.S. Department of Defense.
This report was funded by the U.S. Department of Defense.

Table of Contents

An Investigation into the State of the Practice of CASE Tool Integration

Abstract: In the second half of 1992 a team of the SEI CASE Environments Project conducted a study into the state of the practice with respect to the operational use of integrated Computer-Aided Software Engineering (CASE) tools. After many false starts, we interviewed a number of examples of large organizations with integrated CASE tools in operational use on software development projects. Compared to the state of the art described in much of the literature, what was found might be considered modest. Compared to industry norms, it was quite impressive, representing significant commitment, ingenuity, and significant attention to end user needs.

This report details our observations and analyzes the current state of the practice of CASE tool integration as revealed by our study. It also speculates on reasons for the modest state of the practice.

1 Introduction

Software engineering literature contains many articles about Computer-Aided Software Engineering (CASE) tools, CASE tool integration, CASE tool integration frameworks and standards, project support environments (PSEs), and integrated PSEs (IPSEs) (e.g., [Brown 92], [Chikofsky 93], [Long 91]). This literature documents a range of distinct approaches to CASE tool integration, including: relatively independent tools with little integration; direct tool-to-tool linking as commonly practiced by CASE vendors; indirect linking of tools through a common framework such as Hewlett Packard's Broadcast Message Server and Atherton's Software Backplane; and hybrid integrations combining features of the various approaches.

However, informal discussions with practicing software engineers from the defense and engineering communities indicate that few of the concepts, standards, and products that purport to provide CASE tool integration have found their way into operational use. If this is the case, then the resulting wide gap between the state of the art as represented in the vendor literature and the state of the practice as represented by the technology in operational use may be a cause for great concern. There are a number of reasons why the width of this gap may be of significance:

- The vendor-documented state of the art or software engineer's impressions of the state of the practice may not be accurate representations of the type of integration that is possible and commonly achieved with current tools and integration mechanisms.
- The gap between the state of the art and state of the practice may indicate a significant problem in the transition of the new technologies from research to actual use.

- The gap between the state of the art and state of the practice may indicate a conflict between the directions of the research community and the needs and capabilities of the user community.

In all of these cases, the gap between the state of the art and state of the practice may lead to inflated end user expectations. If such inflated expectations are not met, there may be less satisfaction with the modest improvements that are made, and consequently a greater reluctance to continue investing in the technology as it evolves. Future research in this case would serve only to improve the state of the art and have little or no effect on the state of the practice.

1.1 Goals

It is our belief that an understanding of the current state of the practice of CASE tool integration is an important step in identifying the potential for future CASE tool integration research and for establishing priorities with respect to the transition of the best of the state of the art into everyday use.

With this in mind, we were determined to elicit the actual state of the practice of CASE tool integration in representative organizations from the defense and engineering communities. We aggressively sought examples of CASE tool integration with the hope of testing the hypothesis that the use of integrated CASE tools was common among industry organizations.

In addition, we sought to identify the integration approaches followed and the success of those approaches in the observed CASE tool integrations. Specifically, we wished to determine whether the integration efforts investigated were adopting direct tool-to-tool integration approaches, or were opting for newer, and potentially more powerful, framework-based approaches. We hypothesized that the investigated integration efforts would reflect the growing capabilities provided by CASE tools and tool integration standards and frameworks.

Finally, we sought to identify any common or unique experiences of successful users of integrated CASE solutions. In this manner, we hoped to learn about the most important elements for successful transition of the state of the art into the state of the practice.

1.2 Definitions

Our early attempts to identify the state of the practice of CASE tool integration led to a realization that there is little commonly accepted terminology for the CASE area. The lack of a common nomenclature caused a great deal of confusion in determining the extent to which organizations have integrated CASE tools. An accurate understanding of our questions and of the resulting answers required a common set of vocabulary concerning CASE tools and integration. For this study, we adopted the following definitions:

CASE Tool: Relying on the literal meaning of CASE, we define a CASE tool as any automated tool found to be useful in support of a software project. This is a very broad definition and includes editors and compilers as well as structured analysis and desktop publishing tools. It also includes project management tools, both because project management is part of software

engineering and because project management tools and software development tools manipulate many of the same kinds of objects.

Modern CASE Tool: We distinguish a modern CASE tool as a workstation-based CASE tool that supports a graphical user interface and maintains data in some structured, persistent form such as a database. The intent is to capture in this definition more recent CASE tools, including tools for requirements analysis, design, reverse engineering, testing, document preparation, configuration management, and process management. In many cases such tools support a well-defined method (e.g., structured analysis).

Integration: In our investigations it became apparent that both the organizations interviewed and the interviewers used the term ***Integration*** (as it relates to CASE tools) in an imprecise and variable manner. In order to encompass all of the diverse uses of the term, we here use a broad and standard dictionary definition of integration as the combination of two or more components to form a unified whole. Thomas has offered a more refined definition of integration which focuses on the properties of relationships between tools [Thomas 92].

We also note that the broad dictionary definition can be discussed at two different levels: a conceptual understanding of the relationship between the services or functions provided by the components; and a physical means by which to combine the components and thus permit them to operate as a unified whole.

CASE Tool Integration: To distinguish the integration in which we have an interest, we define CASE tool integration as the integration of two or more CASE tools with either of the following statements true:

1. At least one of the tools is a modern CASE tool; or
2. Third-party integration framework services were used in the integration.

Thus, integration of an object-oriented analysis CASE tool integrated with a structured design CASE tool is of interest, but integration of a compiler and a debugger by conventional means (e.g., through shared files) is not (because neither is a modern CASE tool).

However, if a compiler and debugger are integrated by means of framework services we were interested because of the interesting nature of the integration strategy. The most common available frameworks are PCTE implementations (e.g., Emeraude), ATIS implementations (e.g., Atherton, DEC), or message-passing variations (e.g., HP BMS, Sun ToolTalk, DEC FUSE, IBM WorkBench/6000). Frameworks based on PCTE or ATIS have a data orientation, and frameworks based on message passing have a control orientation.

Operational Use: One significant aspect of the study is the concentration on current actual use of integrated CASE tools. We define operational use as use on at least one development or maintenance project with firm product accountability and an actual customer defining the requirements. Specifically, we wished to exclude academic and pilot projects, prototypes, or other experimental uses.

State of the Practice: We then define the state of the practice in CASE tool integration as the operational use of integrated CASE tools. For this study we were searching for the most advanced examples of modern CASE tool integration in operational use. The operational-use condition proved by far the most difficult to satisfy. Again and again in our searches, we were pointed to organizations where a prototype CASE tool integration was in some stage of development but where operational use was, at best, a potential consideration for the future.

1.3 Approach

Because of the nature of the Software Engineering Institute (SEI)[1], the investigation concentrated on the kinds of CASE tools most commonly in use with the US defense contractor community. Such CASE tools are characterized by their operation in a workstation-based environment in support of large groups of software developers, typically for the development of real-time embedded applications. Specifically, in this study suites of tools marketed as integrated solutions for Information Technology (IT) and Management Information System (MIS) applications were not investigated.

Leads were solicited from CASE tool vendors, the major framework providers, colleagues active in the CASE community, vendor user groups, requests on the Internet, and personal contacts. In general, while we found it relatively easy to identify instances of CASE tool integration, we found more difficult to identify instances where modern CASE tools had been integrated and were in *operational* use.

While CASE vendors seemed to be the most logical source of information about organizations which had successfully used their CASE tool in conjunction with other tools, little information about such uses was forthcoming from the vendors approached. While many of the tool vendors were willing to furnish customer leads, they usually did not discriminate (for our use) between operational users and those just getting started or experimenting with the technology.

A more successful approach to identifying operational users was through the software engineering community "grapevine." A number of potential candidates were identified in this manner. All candidates were screened via the telephone to ensure that they were truly operational users and would be of interest to our study. A number of candidate organizations were eliminated at this time because they could not yet be considered operational users. Interviews were arranged with those organizations meeting our requirements in this screening phase.

By following many leads we eventually arranged six interviews with organizations by an SEI team (2-3 people) and a further six interviews by an individual SEI member. The organizations interviewed were either large companies or major operating units of a large company. Typically these organizations had annual sales in excess of $1 billion. A summary of four representative tool integration efforts appears in Appendix A.

---

1. The SEI is a Federally Funded Research and Development Center (FFRDC) whose aim is to advance the state of the practice in Software Engineering of the US. Department of Defense and its contractors.

During each interview we commonly spoke with members of the development team, managers of the CASE tool integration effort, and sometimes with software developers who were making use of the integrated CASE tools. Interviews were primarily unstructured, but SEI personnel directed questions such that a number of predetermined topics were covered. No formal measurement instrument (such as a survey) was utilized.

1.4 Organization of This Report

The remainder of this report is organized as follows:

Section 2 summarizes our actual observations.

Section 3 provides a discussion of the observations.

Section 4 concludes the report with a summary of findings and recommendations for future work on this topic.

2 Summary of Observations

During the screening process there appeared to be an initial reluctance on the part of representatives of organizations to open up about their achievements in CASE tool integration. This reluctance may be based on either or both of the following: fear of divulging institutional secrets that are intended to confer a competitive edge to the organization, or insecurity in the relatively modest successes that had been achieved. Individuals frequently expressed surprise when they were assured that their work was considered to be among the best state of the practice. As interviews progressed, any initial reluctance to share information subsided.

The organizations interviewed typically appear to have established one of three aims for their integration activities:

- Integration focused on document generation, most often involving the integration of a CASE analysis and design tool with a document production system. A primary interest of the majority of the organizations was to develop the capability to automatically generate Mil-Std-2167A documents from the analysis and design tool.
- Integration focused on the code generation cycle, involving the integration of coding tools with system generation utilities and CM systems.
- Wider scale integrations incorporating tools such as requirements tools, CASE analysis and design and other modeling tools, coding tools, and CM tools in support of a more general-purpose process support environment.

While there is wide divergence in the aim and scope of the efforts investigated, a number of common observations can be made concerning the state of the practice of integrated CASE. These observations are grouped into five categories and are detailed in Sections 2.1 through 2.5. There is little doubt that individual examples could be found to contest any of these observations, but we believe that they accurately represent the current state of the practice in integrated CASE as implemented by a range of organizations within our target community.

2.1 Characteristics of Modern CASE Tools

While our efforts did not focus on the use of individual CASE tools, a few observations did reflect on isolated tools. These observations suggest that while great strides have been made by CASE vendors, some problem areas remain. Observations include:

1. The interviewed organizations suggested that early generations of modern CASE tools were unreliable and were subject to database corruption and system crashes. While individual problems with current generations of modern CASE tools were noted, the interviewed organizations suggested that the tools had matured to the point that they were a reliable and robust aid in developing software
2. Modern CASE tools often require tailoring to address the particular needs of the organization. Tailoring was accomplished primarily by using the tailoring mechanisms provided by the tool vendor.

3. A majority of organizations indicated that modern CASE tool support of programming in the large, while improving, is still inadequate in some areas. Specific problem areas identified included inadequate mechanisms for integration with other tools, conflict with support capabilities such as configuration management (CM), and poor performance when used by large numbers of engineers or when used in support of development or maintenance for large systems.

2.2 Characteristics of CASE Tool Integration

The level of tool integration achieved in the interviewed organizations was quite modest compared to the reports commonly found in the CASE vendor literature. This general finding would be somewhat surprising in light of our attempt to locate the best examples of tool integration were it not for our difficulty in locating operational users of integrated CASE tools. The observed state of the practice in general did not support the claims made by some CASE vendors concerning the ease of tool integration. The interviewed organizations commonly found the available interfaces to be inadequate, tight integration extremely difficult to achieve, and the time to achieve the desired gains in productivity and quality more likely to be measured in months than in days.

Among the more specific observations concerning the state of the practice of CASE tool integration are:

1. In all of the organizations which were using CASE analysis and design tools to automatically generate documents, *significant enhancements were required* to the tool-provided documentation capability in order to meet the organization's requirements. This finding is not surprising in light of the need to tailor documents for particular organizations and customers. However, the scope of the effort required to identify and implement the necessary enhancements was surprising. For example, one organization spent approximately 2 staff-years of effort in order to modify the tool-generated System Requirements Specification (SRS) to meet organizational needs.

2. Of the organizations interviewed who had CASE tool integrations in operational use, one was using a message-based integration framework (Sun ToolTalk). Another organization was using a relational database to accumulate data from multiple tools. Two organizations contacted were in the process of developing integrations of tool data centered around data frameworks, but had not achieved operational use.

3. Of those organizations attempting framework-based integration, none had rewritten existing tools to take full advantage of the integration services provided.

4. No operational uses of the PCTE or IRDS standards were identified. One instance of the use of the CDIF standard was identified, along with one proposed use of ATIS (which is not a formally recognized standard). Some "non-users" expressed interest in such standards, but none identified plans to migrate toward such standards over the near term.

5. All organizations expressed frustration with the integration capabilities of existing CASE tools. Reasons given for this frustration include the inability to invoke the full range of functionality from outside the tool, awkward and incomplete access to data, and poorly documented interfaces.

6. A degree of user interface consistency was commonly implemented by extending and customizing tool menus to allow access to some of the capabilities of other tools. We did not identify any organizations attempting to build a common user interface across tools.

7. Among operational users, data sharing between tools was primarily accomplished via vendor-provided access routines. As previously indicated, one instance of the use of a relational database was identified among operational users.

8. Among operational uses as well as in systems under development, data integration was accomplished primarily for coarse-grained objects. For example, a common data-integration scenario involved the extraction of a complete data flow diagram for inclusion in a document.

9. All tool integrations observed allowed only unidirectional data flow from one tool to a second tool. No instances of bidirectional data flow were identified. For example, changes made to a name on a data flow diagram in a documentation tool were not reflected in the "source" diagram in the analysis and design tool.

10. Synchronization between tools was commonly enacted by developing scripts that control tool invocation. Scripts written in both a general-purpose programming language (C) and operating system command language (shell scripts) were identified. In one case, use was being made of a framework capability to provide tool-to-tool communication and control (ToolTalk).

11. Integrations had a limited focus. No instances of a cradle-to-grave integrated environment were observed. A number of organizations focused their integration efforts on document production using analysis and design tools and document generation systems. One organization focused on the coding cycle. One organization focused on supporting a wider range of activities, including requirements analysis, design, coding, and documentation.

12. Plug compatibility between tools was not observed. Only two instances of replacing one tool in a CASE integration with another tool of the same class were identified. Both of these instances involved the same organization, and led to divergent implementations which required separate support and maintenance.

2.3 The Relationship of CASE Tools and the Software Process

A common theme running through all of the CASE tool integration efforts examined is the importance of the interrelationship between tools and engineering processes. A number of observations were made concerning this important relationship. The following observations

address the specific processes supported and the mechanisms necessary for insuring that tools support those processes:

1. A majority of organizations acknowledged the importance of the link between tool integration and process support activities. This link was commonly supported by close cooperation between separate process and tools groups, and by the combining of process and tools functions under a single umbrella. One organization suggested that interrelated and permanent process and tools groups were important.

2. The majority of organizations interviewed were actively involving end users in decisions concerning the type of support to be provided. Reasons given for involving end users were to insure a close match between user needs and integrated tool capabilities and to enhance the probability that the integrated CASE capabilities would be adopted by end users.

3. A number of organizations expressed concern over the integration of CASE tools with the organization's CM practices and structure. To these organizations, overall consistency of CM practice was a primary concern but was particularly hard to achieve since CASE tools frequently support different CM models than the one adopted by the organization.

4. All organizations interviewed were tailoring analysis and design tool integrations to fit their particular needs. No organization was using the integration as it came from the vendor. As a result, all organizations felt that flexibility was an important characteristic of a tool.

5. The use of structured approaches to performing software engineering activities is becoming more common. All organizations interviewed were utilizing the structured methods provided by an analysis and design tool. A number of organizations were utilizing inspection techniques such as Fagan code inspections.

6. The efforts of all organizations suggest that CASE integration is not a quick fix. All organizations interviewed had developed their integrated CASE solutions over a multi-year period. In all organizations, the integrated CASE solution was still evolving.

2.4 Problems of Adoption

Organizations having initial success using and integrating CASE tools appear to be struggling with efforts to place these capabilities into general use. Observations made concerning the problem of CASE adoption are:

1. integrated solution into general use. One organization had modified their tool integrations and their marketing strategy to make it more acceptable to users. Another organization made experience with the integrated solution a de facto requirement for technical leadership on a new project.

2. The majority of interviewed organizations suggested that end user input and feedback are important to the successful adoption of a CASE integration.

3. The majority of interviewed organizations had established a permanent tools group to handle the ongoing support necessary for the CASE tool integration. This was particularly important for those organizations involved in more substantial integrations of larger numbers of CASE tools.

4. All organizations believed that integrated tool sets should be flexible enough to meet the needs of a range of users. Such flexibility is necessary to allow customizing of the tool for use by different individuals and organizations. Extending and enhancing tools is always needed and should be anticipated.

5. All organizations interviewed expressed the need for significant training in the use of CASE tools and tool integrations. To increase control and reduce the cost of ongoing training, some organizations had developed in-house training capabilities. Another approach to reducing training costs and improving tool use was to develop extensive documentation tailored to the organization.

6. Organizations involved in more substantial integrations commonly used a phased approach to implementing and transitioning tools and tool integrations.

7. The majority of organizations had first initiated integrated CASE usage on a small project.

8. The majority of interviewed organizations had some expertise with the individual tools before attempting to implement and transition an integrated capability. It was suggested by one organization that a tool set will be more readily accepted if it includes support for existing tools and practices.

9. Some organizations had facilitated transition by identifying and encouraging champions for the integrated CASE tool capability. This approach was particularly pronounced in the one organization that had made CASE expertise a de facto requirement for technical leadership.

2.5 Working with CASE Vendors

Organizations often found it difficult to work with CASE tool vendors. A common theme discussed by a number of organizations was that CASE vendors are not always responsive to problems and requests. Among other observations concerning CASE vendors are:

1. All interviewed organizations were promoting open systems environments. This was demonstrated particularly in a move to UNIX and a desire to work with multiple vendors.

2. Some interviewed organizations had negotiated better prices and service by combining the tool needs of smaller organizational units into a single, large procurement.

3. A number of organizations expressed concern over the lack of influence that tool users have with tool vendors. Some suggested that the organization has the greatest leverage when purchasing tools, but the degree of influence declines subsequent to purchase.

4. The majority of organizations suggested that tool upgrades are difficult to manage, particularly when tools are integrated with other tools. Because of the need to tailor tools and integrations to user needs, each upgrade potentially requires modifications to the tailoring.

5. Common complaints of interviewed organizations about tool vendors included broken promises, slow response to problems, and perceived emphasis on new sales rather than existing customers.

3 Discussion and Implications

In this section we provide a more detailed discussion of some of our observations. This discussion is based on what we observed and what we were told by interviewed organizations, on what we can infer from our observations, and on the authors' overall knowledge of the CASE field. While we gathered our observations from a small number of organizations, we believe that the observations and discussion reflect the nature of tool use and integration in the larger defense-related software community.

The discussion of our observations is organized according to the same five broad categories as in Section 2. The five categories of observations are:

- characteristics of modern CASE tools
- characteristics of CASE tool integration
- the relationship of CASE tools and software process
- problems of adoption
- working with CASE vendors

3.1 Characteristics of Modern CASE Tools

The interviewed organizations which had used early (mid- to late 1980s) versions of modern CASE tools commonly experienced significant problems with database corruption or system crashes. The head of one integration effort suggested that among the most useful people on the team using these early tools were the computer support personnel who could create and restore backup tapes and restart the tool.

The quality, performance, and capabilities of the current generation of modern CASE tools appears to be much improved over the earlier versions. Crashes are reportedly less frequent and are less disastrous when they occur because the tool database is generally not corrupted. In addition, tools now provide enhanced functionality (particularly for document generation) that appears to better address user needs.

Some organizations do continue to report problems with the current generation of modern CASE tools. These problems often involve what is perceived as inadequate support for large group development. Notably problematic according to some organizations are the relatively poor configuration management capabilities of tools. Organizations also mentioned that some CASE tools do not integrate well with other tools, in spite of claims of openness and support of various integration standards. According to these reports, tools remain primarily "self-centered." They do not appear to provide adequate mechanisms to allow the full range of their functionality to be invoked and directed from another (external) tool.

In spite of some problems, the organizations interviewed were involved in integration efforts because they felt that modern CASE tools (as well as more common tools) had matured to the point of offering useful support for many aspects of software engineering. This does not imply

that these organizations assume that modern CASE tools will automatically provide large benefits to any organization. The interviewed organizations had spent considerable resources tailoring the tools to support their unique needs and processes and in extensive training of personnel to insure proper use of the tools.

No organization interviewed is known to have used a modern CASE tool as it arrived on tape. The interviewed organizations felt that a significant portion of the benefit offered by their modern CASE tool was due to the tailoring activity that provided a closer match of the tool to the organization, and also to the training in process and methods associated with adopting the tool.

3.2 Characteristics of CASE Tool Integration

3.2.1 Integrated Support for Well-Defined Subprocesses

No organization interviewed (including those attempting wide-ranging integrations) has been able to develop an integrated process-centered environment supporting the full software life cycle. Rather, integration is most often achieved within more manageable engineering subprocesses. As previously mentioned, the most common subprocesses for which tools are integrated are the documentation processes. The editing/coding/debugging cycle is another common target for tool integration. Organizations attempting wide-ranging integrations commonly provided strong links within two or more subprocesses, with weaker links across subprocesses.

Characteristics of the subprocesses for which tight integrations have been developed are that they are well-understood and they produce a definite product. For example, the generation of Mil-Std-2167A documents is commonly a goal of tool integrators. The documentation standard itself serves as a well-defined set of requirements for the integration effort, while the resulting set of documents represent a definite and essential product.

The corollary to this position is that for less well-defined (or more variable) portions of the life cycle, integration has proven more difficult. This is reflected both by the choice of a number of organizations to limit integration activities to well-understood subprocesses, and in uncertain success of efforts aimed at less well-defined areas. However, some of the more aggressive integration efforts are attempting to incorporate prototyping capabilities, automatic metrics generation, and project management support.

3.2.2 "Traditional" Integration Technologies

The majority of integration efforts are utilizing integration approaches that have been common for a number of years. Both current and previous efforts are characterized by filters, which extract data from one tool database and make it available to another tool, and scripts, which control sequences of data extraction and tool invocation. However, the mechanisms used by current efforts differ from earlier efforts. Unlike earlier efforts which used primarily user-developed scripts and filters written in conventional programming languages and operating system

scripts, current efforts often use languages and filters specifically provided by a tool vendor for the purpose of integrating tools. In many cases, the tool vendor provides a turnkey integrated capability which can be and often is substantially enhanced by the users.

A primary limitation of script- or filter-based integration efforts described by the interviewed organizations was the inconvenience of access to, and poor documentation of, the program-callable interfaces through which programmatic access to tool data is made possible. The majority of the interviewed organizations felt these interfaces were awkward, incomplete, or poorly documented. Organizations also commented that an unfortunate side effect of using currently-available program-callable interfaces was the highly tool-specific nature of the resulting integrations, in effect locking the organization into using specific tools and making it both costly and inconvenient to migrate to different tools. Although various industry and standards organizations are investigating the standardization of program-callable interfaces, data organizations, and messages for various tool classes, this sort of plug-compatibility is not yet a reality.

The examined integration efforts also share the characteristic of limiting information flow to primarily one direction in that information is *extracted* from one tool and *provided* to a second tool rather than *shared*. As a consequence, modifications to a data item in one tool are not automatically reflected in the related data item in another tool. For increased benefit, *data sharing* requires not only an agreement on the data to be passed, but also a concept of the process or method that underlies the desire to share the data, as well as an understanding of the data relationships within the other tool. One way of providing this type of data sharing is through use of a shared data schema and database. Current CASE tools from different vendors rarely provide this level of integration.

3.2.3 CASE Integration Architecture

Our initial expectation was that we would observe a number of instances of the early use of framework-based integrations. However, a large majority of installations were not using a framework-based approach, and all organizations were using an eclectic or "hybrid" approach in order to perform integrations with the available tools. These hybrid approaches commonly utilized differing levels of tool integration within a single effort depending both on the level of integration necessary and the level that was "doable." The observed CASE integrations often demonstrated characteristics of two or more of the following approaches:

- relatively independent tools
- direct tool-to-tool integration
- framework-based tool integration

Some tools within the studied CASE integration efforts remained relatively independent, with little or no attempt made to integrate them with other tools. Even these tools were afforded some measure of integration as a result of following consistent standards, guidelines, or conventions that are external to the tools. An example of the level of integration possible with independent tools is the consistency offered by multiple tools following a common look-and-feel format such as Motif.

A number of the organizations contacted were currently using this as an initial approach to the use of CASE tools. They were typically in the first stages of CASE tool use on operational projects concentrating on a specific functionality offered by one or more CASE tools. The expectation is that as their knowledge and use of the CASE tool grows they will be more interested in adopting an integrated CASE tool solution. In addition, other organizations with well-developed integrated capabilities continued to use this approach with tools that had not been incorporated into their integrated solution.

A very common integration approach involved utilizing direct tool-to-tool integrations provided by either tool vendors or by the organizations themselves. Vendor-sponsored tool-to-tool integration efforts can be seen in CASE tool workbenches in which a number of tools provided by a single vendor are packaged by that vendor as a set of interconnected tools that perform some logically-connected set of actions. Another form of vendor-provided integration entails situations where two or more CASE tool vendors form a partnership or strategic alliance to integrate and market versions of their tools that can be used more productively together.

In some cases, end user organizations independently constructed the direct tool-to-tool integrations that they required. For complex CASE tools such as design tools, desktop publishing systems, and testing tools, our observations suggest that the amount of effort involved in implementing the CASE tool integrations can be very significant (e.g., between 1 and 5 person-years of effort to integrate two complex CASE tools and to evolve that integration into something reliable and usable). However, the majority of organizations adopted some form of this solution for situations where a suitable alternative was not available.

Framework-based tool integrations were not common, but are beginning to appear. Such integrations employ a substrate providing integration services. Frameworks can take many forms, but two particularly common technologies are structured, persistent repositories for storing and sharing data, and message broadcast mechanisms which notify other tools of events or request specific services of another tool.

Fewer of the interviewed organizations were attempting framework-based solutions to CASE tool integration than we had expected. While such an approach was often cited as a goal for the future, operational use of integrations based on framework approaches were not common. However, the few examples we observed appeared to be enjoying a reasonable degree of success, both taking a database-oriented approach and taking a message broadcast-oriented approach. Interestingly, some organizations choosing a framework-based approach were doing so not because of end user interest, but rather to decrease costs of implementation and maintenance of the integrated tool set. These organizations felt that a framework-based approach to integration was potentially a large improvement over earlier, point-to-point solutions.

A possible reason for the general lack of effort to incorporate frameworks is that tool users are currently struggling to absorb and integrate basic tool functionality. Most have not progressed to the point to even begin addressing framework products, in part because of the relative immaturity of the technology. However, a number of organizations see the long-term future of their integrated CASE solutions as a more consistent, framework-based solution, and look to

products and standards such as HP SoftBench, Sun ToolTalk, and ECMA PCTE as being likely to play a significant role in this future.

The most common approach we observed was a hybrid of the independent and the direct tool-to-tool approaches. Some organizations were also beginning to make use of framework-based approaches. A hybrid approach to CASE tool integration represents a practical effort to integrate existing CASE tools from a wide variety of vendors, which frequently do not provide extensive integration support. While the hope is that a hybrid approach offers the combined advantages of each of the three individual approaches, a potential problem is that it may provide the combined disadvantages, including inconsistent degrees of integration across the environment, extensive maintenance demands, and reliance on unproven technologies.

Given our observations on the current state of the practice, it is likely to be a number of years before framework products and standards are of sufficient maturity so that organizations can readily apply them. Currently, these organizations claim to have an insufficient knowledge of the relationship between the technology and its effect on their organizational and engineering processes to be in a position to take advantage of these products and standards.

3.3 The Relationship of CASE Tools and the Software Process

The organizations interviewed were strongly aware of the need to integrate the organization's process with tool support. In most cases, the strength of the relationship between process and tools was extremely strong; one organization considered process and tools to be inseparable and refused to even consider the value of either process or tools alone.

A number of organizations had distinct process and tools groups which interacted when defining necessary tool support. Other organizations had a single group which was chartered to consider both process and tools issues. In most cases, cooperation between groups addressing process and tool issues was seen as being important to their mutual success. However, in one organization, such cooperation was not common.

The majority of the organizations interviewed professed a strong process orientation within some part of the company. A number had efforts directed toward the SEI Capability Maturity Model (CMM). Other organizations had adopted alternate process models and were introducing comparable process assessment and improvement programs.

However, some organizations interviewed took issue with what they perceive as a strong "process first" focus of some software process advocates. These organizations suggest that while in theory an organization can specify a process and then identify tools that support that process, in practice this approach is faulty. They found that the quality of the tool support available is often a major consideration in developing, encouraging, and enforcing a process. For these organizations, knowledge of available tool support is an important input when defining a software process.

One organization suggested that as tools are integrated the importance of a strong fit to process increases rapidly. While the organization's process can often be modified to incorporate a single tool, an integrated tool set is likely to address a larger portion of the software process. In the case of an integrated tool set, deviation from the existing process may affect a larger number of individuals and subprocesses. The end result may be increasing difficulty of transition of the tool technology within the organization.

3.3.1 Carefully Considered Integration Efforts

All of the organizations interviewed had some integration of modern CASE tools in operational use. Usually this was an analysis and design tool integrated with a documentation tool. Moreover, each had further integrations in various stages of planning, development, and experimentation.

In order to direct their integration activities, each of the organizations interviewed had developed long-term plans for both process improvement and tool support. However, the thrust of integration efforts was very much bottom up, in the sense that initial efforts did not attempt to define an outline of the overall integrated CASE system, but rather attempted to provide strong support for some engineering subprocesses.

For even bottom-up integration efforts to be successful, the interviewed organizations suggested that a clear concept of operations was necessary. The concept of operations should identify how the specific integration activity would serve the current needs of the software developer, as well as the longer-term goals of process and tool support.

The deliberate manner in which these successful integrators of CASE tools pursued their goals was also evident in the time it had taken to achieve their current level of integration. For most of these organizations, integration of CASE tools (and correspondingly process improvement) was a multi-year goal. Three reasons for this very deliberate approach were suggested: first, the learning curve is very steep for new users of CASE tools; second, industry in general has little experience integrating modern CASE tools in order to support a defined process; and third, integration efforts present high risk due to the unstable nature of the CASE tool and framework markets.

3.3.2 Configuration Management Tools and Process Support

The interviewed organizations often had made a significant effort to either build a CM system on top of a source code control system or to enhance a third-party CM system. Matching the CM tools with an organization's overall CM process requires a great deal of effort, even without considering the design objects or the requirements objects manipulated by design and requirements CASE tools.

Due to resource limitations, little time in the study was spent investigating integration involving CM services. However, this seems to be a fertile area for further investigation. Companies have been integrating tools into CM systems for a number of years with varying degrees of

success. Insight into CASE integration issues could well be gained by a more careful examination of the problems and issues plaguing users of CM technology.

3.4 Problems of Adoption

The users of modern CASE integrations were first of all users of modern CASE tools. Each organization previously had years of experience with modern CASE tools prior to producing significant and mature integrations of tools. Their successful integrations were part of their evolving use of modern CASE. The individuals with whom we spoke were highly motivated to see their ideas in operational use. They were creative in selling the capabilities of their tool integrations. It is our impression that they attempted to be responsive to user needs. However, even for such motivated organizations, the transition of the new technology to users was a difficult task.

3.4.1 Selling CASE

A striking characteristic of the groups interviewed was their emphasis on "selling" their integrated CASE wares to internal clients. Most had made multiple attempts to make their integrated capabilities more attractive to users by carefully modifying the integrated support provided in light of input from end users.

However, even though the groups were concerned with user needs and had strong management support, the staff on software projects were often extremely reluctant to adopt the capability. Frequently, the reasons given for this reluctance were based on previous, unsatisfactory experiences with CASE technology. Unsuccessful experiences had often left a general feeling that CASE benefits are greatly exaggerated and difficulties greatly underestimated. Specific complaints included concern about a lack of the training and practice time and inadequate support to address problems as they arose.

The understanding of end user needs and the ability to react to them had therefore become an important component in the successful adoption of integrated CASE tools. Representative approaches used by tool groups to encourage CASE adoption included:

- providing document templates for common document formats in order to encourage staff to use the "WYSIWYG" documentation tools;
- emphasizing the benefits of a single tool as a first step in adopting a larger integrated capability;
- providing new hardware (workstations) as an incentive to adopt the new technology;
- soliciting user feedback and critique in order to identify desired changes and addressing the problems utilizing the user's priority scheme; and
- offering cheaper prices on the recommended tools via group purchasing.

The tools groups were highly aware that user word-of-mouth was critical to the success of the adoption efforts. Successful groups had encouraged word-of-mouth transmission of informa-

tion about their CASE integrations. Among the most successful organizations was one in which word-of-mouth success stories had increased the demand for and relative value of staff members with experience using the integrated CASE technology. In this organization, experience with the integrated CASE technology had become a de facto requirement for senior technical staff.

Of course, such positive word-of-mouth transmission is dependent on the demonstrated success of integrated CASE tool approaches. Such success often came from the use of a bottom-up approach which introduced integrated CASE solutions directly tied to user needs. In particular, organizations introducing integrations of a relatively few tools which reduced the documentation demands on engineers, or which improved the manner in which engineers accomplished editing, compiling, and debugging of programs, were more successful.

In contrast to the successful efforts interviewed, which used a bottom-up approach, we identified a number of organizations that were attempting a "big bang" approach to CASE adoption. These organizations were introducing a large number of tools at one time on a large project. We were not able to find evidence of operational use in cases where a big bang approach was tried, suggesting that this approach may be less likely to succeed.

3.4.2 Making CASE Work

The successful organizations interviewed emphasized the importance of a strong support staff. This staff was necessary in order to perform several distinct functions: for keeping the tools alive and well on the host operating system; for hand-holding of new users; for problem solving with more experienced users; and for tailoring and enhancement in order to support new projects.

One organization estimated that support required the services of a system manager half-time for the first two years that CASE tools were in use. This organization also reported that a post-mortem interview was conducted with users after every major document delivery generated by their CASE integration. Problems were identified and weighted by the number of project staff hours required to work around the problem. Over an 18-month period, they reduced the work-around rates by a factor of eight.

In order to facilitate the interaction between users and support staff, one organization switched people back and forth between the tools support group and operational projects. This provided highly-experienced tool experts to projects and infused user viewpoints into the tools group.

3.4.3 Added Value from In-House Training

All of the organizations stressed the importance of training. The recommended training not only included tool training, but also training in the appropriate methods and in the specifics of the organization's approach to tool use. Although organizations frequently used vendors for initial tool training, most subsequently found it desirable to develop an in-house tool training capability. Several reasons were cited for this, including:

- Organizations required a tighter control of training content than that provided by an outside contractor or tool vendor. With in-house training, organizations were able to select parts of methods and tools to emphasize or de-emphasize. The selection was dependent on the nature of the application or on fitting the method into the project's overall process. Also, as an organization enhances or tailors a set of tools, the need exists to include these changes into the training. It can also help successful adoption if the examples and exercises provided in training closely relate to the organization or project.
- Some external courses are focused and well taught but others can be ill-conceived or poorly executed. Frequently external teachers have little practical experience. Moreover, an external trainer is unlikely to be an expert in the organization's application domain.
- All of the organizations interviewed believed that the timing of training was very important. If the training precedes use by more than a few weeks, much important information is forgotten. Internal training can usually be scheduled more flexibly to ensure that users have the information fresh in their minds.

3.5 Working with CASE Vendors

All of the organizations interviewed stressed the importance of developing a close working relationship with CASE tool vendors, but they indicated considerable frustration with this relationship. While several organizations found small vendors to be more responsive that the more established ones, it is difficult to draw a general conclusion in this area since the expectations of users and the quality of the user-vendor relationship varied widely.

Organizations reported good success in negotiating prices when they were able to deal with a vendor with a unified voice. Even in these cases, however, some organizations voiced discontent with their ability to influence the technical direction of vendors. Moreover, organizations found it difficult to discern vendors' plans for the future. This made it difficult for organizations to plan their own environment evolution.

A number of organizations suggested that after a CASE tool is put into operational use, upgrades (new versions) can be problematic. Even without the bugs sometimes present in new versions, changes in a tool's functionality can cause confusion, user errors, and lost productivity. If the upgraded tool has been integrated with other tools, the potential for problems is increased, and integration code may have to be changed.

One very positive finding was that vendors do appear to take serious problems seriously. In general, the interviewed organizations thought highly of their vendors' technical staff. There seemed to be general agreement that the tool offerings of established vendors are more robust now than they were in the late 1980s.

However, some organizations expressed a concern that on occasion vendors were concentrating more on selling new tools than on improving existing tools to encourage operational use. For example, in order to sell to new customers, tool vendors continually added support for new features and methodologies, rather than improving support for current features and methodologies.

For the organizations interviewed, Unix was the most popular operating system. Two organizations admitted abandoning substantial investments in proprietary host platforms. The major reasons cited for preferring Unix were an improved ratio of performance to cost, better support by CASE vendors, and greater portability.

4 Summary and Conclusions

The need for automated support for many aspects of software development has lead to great interest in the many CASE products that are currently available. However, organizations have varying approaches toward the adoption of this technology, often based on their previous experiences with the technology. The goal of this study has been to analyze the current state of the practice within organizations using CASE technology and to report on their progress in the area of integration of CASE tools.

What we have found is a spectrum of approaches towards CASE tool integration. These approaches appear to reflect two factors:

- the maturity of CASE tool and framework technology, and
- the maturity of the organization in terms of its previous experience with CASE technology.

Based on these factors we can identify five different situations in which an organization may currently find itself.[2]

1. *Isolated CASE tools.* An organization employs particular CASE tools as and when necessary. Little attempt is made to integrate those tools.
2. Clusters of CASE tools. Collections of CASE tools are integrated to support a part of the process. The tools have been integrated using a point-to-point approach.
3. Migration toward framework-based integration technology. The integration of CASE tools using a database or message-passing framework has been carried out. Initially only a small part of the CASE environment may have been integrated in this way.
4. Loosely integrated collections of CASE tool clusters. The existing clusters of CASE tools are integrated using point-to-point or framework-based integration approaches. A different approach to integration may exist between the CASE tools in a cluster and between the clusters.
5. Complete integrated CASE environment. A complete CASE environment where all tools are integrated using a single framework-based solution. The CASE environment is closely matched to the organization's needs.

Our observations indicate that the majority of organizations are currently in Situation 1 or 2 with isolated CASE tools or small clusters of CASE tools directed at a well-defined task. We also observed the first steps toward implementing Situations 3 and 4 with the use of a framework-based integration approaches and loosely integrated collections of CASE tool clusters.

---

2. The five situations should not be interpreted as a model of CASE tool or organizational maturity. The most appropriate degree of integration of tools will depend heavily on the organization's specific tool needs. Thus, the use of isolated CASE tools may represent the most appropriate situation for a specific organization.

Almost all organizations we investigated see their goal over the next few years as Situation 4, with the ability to link clusters of CASE tools using a framework-based solution. However, since this goal is viewed as long range, few organizations are experimenting with (or even express significant interest in) currently-available framework products — they are currently too involved in addressing near-term CASE tool integration concerns.

The more ambitious goal referred to in Situation 5 is viewed by most practitioners as very far off in the future. Interestingly, this approach is the target of much of the current research interest.

The maturity of most individual CASE tools is generally viewed as quite high, with the major CASE tools providing reliable, sophisticated support for well-defined activities. The more recent framework-based products, however, are poorly understood and viewed with some suspicion. A greater depth of knowledge of these products and more extensive operational experience will be required before many organizations will be willing to invest heavily in them.

With regard to the previous CASE experience of an organization, our observations showed that the introduction of integrated CASE solutions requires a CASE pedigree. Those organizations that can be viewed as more successful with integrated CASE had many years of experience with isolated CASE tools, clusters of CASE tools, and were considering framework-based solutions. The least successful organizations had attempted to build framework-based solutions when they had limited experience with isolated CASE tools and almost no experience with the use of clusters of CASE tools.

Hence, our observations indicate that the five situations we identified are not independent. In particular, without a strong pedigree in isolated CASE tools and clusters of those tools, the approaches offered by framework-based products are much less likely to succeed. Adopting an incremental and evolutionary approach toward integration of CASE tools appears to be the best way to proceed.

A second and perhaps more significant finding of this study is that conclusions drawn based on interviews and case studies, while interesting and potentially useful, are not adequate answers to the many questions concerning CASE tool use and integration. The formulation of better answers will depend on the availability of metric data which allows us to address questions such as:

- What methods are best for evaluating and comparing CASE tools?
- How do we determine the appropriateness of a CASE tool to a user's particular situation?
- How do we compare the effectiveness of one CASE tool integration strategy against another?

It is difficult to see how we can advance in the use of sophisticated CASE tools and tool integration without answers to these and related questions.

Finally, we note that the principal value of this report is that it captures the state of the practice of CASE tool integration for a broad range of technical engineering applications for a given point in time (the end of 1992). In many respects operational use of integrated CASE technology is in its early stages and will undoubtedly see many shifts and changes through the remainder of the decade and beyond. In order to accelerate the maturity of operational use of integrated CASE, a greater focus on real end user needs is required by both CASE tool vendors and applied researchers. This will improve the likelihood that integrated CASE technology will be directed at the most acute problems faced by software engineers.

Appendix A Samples of CASE Integration Efforts

This appendix provides a short overview of four of the organizations interviewed in the study, highlighting the major characteristics of their CASE tool integration efforts.

A.1 Organization A

Organization A is a large aerospace firm involved in the production of both commercial and Department of Defense systems. The tool integration efforts of Organization A were an outgrowth of the organization's software process improvement program. This program identified improved documentation of systems (particularly Mil-Std-2167A documentation) as critical for improving the software process.

Initial efforts (circa 1988) at providing automated documentation support were primarily small-scale efforts aimed at generating portions of Mil-Std documentation for specific projects. These initial efforts integrated two COTS analysis and design tools with in-house documentation capabilities. Integrations consisted of scripts which extracted data flow and other information from analysis and design tool databases and inserted this information into documents. Consistency between documents and analysis and design tool databases was maintained by the periodic regeneration of design documents. These integrations operated in a proprietary operating system environment, required significant manual effort, and were difficult to maintain.

In spite of problems, both developer and customer reaction to the generated documentation was favorable. Internal regard for the integration effort was positive enough to encourage other projects to adopt and enhance the system. One early enhancement involved the introduction of a popular COTS documentation tool. Other early enhancements included expanding the range of documents generated and migrating from the proprietary operating system to Unix systems.

As successful projects using the capability were completed, project members experienced with the tools were much in demand for new projects. The integrated capability was applied to larger systems, now including a number in the 100-500K source lines of code (SLOC) range. Experience with the methods and tool set are now considered essential for technical leadership positions.

Approximately three person-years of effort have gone into the system, divided up between a number of individuals working on independent projects. Currently, the tool set can automatically generate a full range of Mil-Std-2167A documents. Alternate versions of the integrated tool set have been produced, with substitutions for both documentation tools and analysis and design tools. The analysis and design/documentation tool integration consists of approximately 20K lines of tool scripts and C code.

Distinct versions of the tool set are used to support software written in Ada and C. Due to differences in the languages and methodologies, the versions are becoming increasingly divergent.

A multilevel CM library structure for documents and source code has been developed to support the integrated tool set. Tools have been developed to handle promotion of documents and source code within the structure and to regenerate documentation at any level.

A number of problems have been identified, including:

- The redelivery of new versions of the integrated tool set when a new version of a tool is released is difficult and has gotten worse as the complexity of the integration has increased.
- The tools used do not allow the level of interaction necessary for traceability.
- Integration with the organization's CM capabilities has been difficult.
- COTS tools appear to be written to be used independently and not under the control of other tools.
- The management of in-line pictures and equations is difficult.
- Some resistance is commonly met during transition of the tool set.

A.2 Organization B

Organization B is a large aerospace firm involved in a variety of commercial and defense-oriented projects. In approximately 1988, managers at Organization B saw a demonstration of CASE tools. In addition, senior-level technical staff had experimented with CASE technology. Following this initial exposure, Organization B invited two well-known CASE vendors to demonstrate their tools. The tool vendors subsequently demonstrated their products and discussed integration of the tool with the a documentation system to produce Mil-Std-2167A compliant documentation. Based on the presentations and the company's other requirements, an analysis and design tool was chosen. In addition, it was determined that the linkage of the analysis and design tool and the documentation system would be exploited.

The integrated analysis and design/documentation capability delivered to Organization B allowed the user to define the structure of the Mil-Std documents using the analysis and design tool interface. While effort needed to accomplish this task was straightforward, the documentation produced was deficient in two ways:

- The data extracted from the analysis and design tool database was insufficient. In particular, additional descriptive text was necessary for diagrams, particularly around data flows. Also, documentation for requirements traceability was poorly addressed.
- The formatting of the resulting document was unacceptable. Font sizes used were inconsistent, and diagrams were poorly laid out on pages.

In order to improve the quality of the documentation produced, enhancements were made to the integration. C language code was developed by Organization B to access the analysis and design tool database, extract additional information, and generate the appropriate documentation tool markup language enhancements. Special information in the analysis and design tool database was identified by the use of hard-coded "markers." Code was also produced to extract and include Ada source files from source libraries. All told, approximately 2500 lines of C code and 250 lines of operating system scripts were generated, requiring 2 person-years of effort.

During the integration effort, a number of impediments to progress were identified. These include:

- The documentation tool's markup language was not sufficiently documented, making it difficult to determine required enhancements.
- Initial versions of the tools were not robust. Tools corrupted data and crashed frequently. Subsequent versions of the tools were more robust.
- The analysis and design tool text editing capability not adequate.
- New tool versions sometimes required reworking of the C interface.

The results of the integration effort appear to be mixed. Among the major findings are:

- A moderate amount of manual effort is still required to generate acceptable documents. This effort includes fixing figure references in the text and tidying up complicated diagrams.
- The unfamiliarity of users with CASE tools and structured methods has been a significant cost factor in the project.
- Training on methods has been essential.
- The integrated tool set has been used on one major project with unclear impact.
- Momentum for enhancements and adoption of the capability has been greatly affected by the availability of a strong CASE champion within the organization.

A.3 Organization C

Organization C is a large aerospace/defense firm which is acting as the prime contractor on a large defense procurement. Personnel from this large contract are assembling a set of integrated tools for use during the procurement. However, the potential target audience for the integrated capability is extremely large and includes the corporation as well as the defense department customer.

Organization C and contract personnel have a history of use of individual CASE tools for analysis and design, document generation, and simulation/modeling. They have over the years developed extensive enhancements to the basic integration capabilities of a popular analysis and design tool/documentation tool combination. Some of those enhancements were given

back to the CASE tool vendors and subsequently were incorporated in the tools and the integration capabilities.

In order to provide a more capable environment for the current defense procurement and for the company in general, Organization C is enhancing the existing tool integrations and incorporating new integrations into the environment. The scope of the resulting integrated tool set is anticipated to be much wider than for the other integration efforts we interviewed. At least 7 commercial tools, including tools for system requirements analysis, software requirements analysis and design, documentation, database support, requirements management, configuration management, and simulation/modeling will be incorporated. Separate but closely cooperating process and tools groups have been important in determining the scope and direction of the integration effort.

The approach taken by Organization C in assembling an integrated tool set is somewhat unique. Because of the large size of the contract and the organization, it has been moderately successful in encouraging CASE tool vendors to provide integrated capabilities conforming to requirements written by Organization C. When Organization C requires an integrated capability, it establishes the requirements and encourages COTS tool vendors to offer a solution. Organization C does not fund the vendor's integration activity, but rather suggests that a large number of future sales are dependent on the integration.

To the chagrin of some CASE vendors that considered themselves to be well established at Organization C, the provision of a specific tool or integration capability has proven to be highly competed by many vendors. A key to such competition appears to be the unified voice with which Organization C is speaking. However, in spite of the healthy competition and strong general position of Organization C, vendors have not always followed through on commitments in a timely manner.

The types of integration solutions provided by COTS vendor are variable, but frequently entail the extraction of tool data from one tool and the insertion of that data into a second tool. For example, data is extracted from a number of tools and incorporated into design documents and reports.

The integrated tool set is not yet completed. However, incremental capabilities are in use on a number of projects. One of those projects employs over 300 software engineers.

Organization C initially utilized vendor-provided training for their integrated tool set. However, most training is now done in-house in order to exert control over the content and timing of the training.

A.4 Organization D

Organization D is a large real-time engineering applications firm with a strong in-house software tool integration and environment activity. This activity pursues three major goals: to en-

courage use of best practices and procedures, to integrate and support best-of-class CASE tools, and to provide a conduit for training and support from vendors.

The initial tool integration efforts of Organization D involved building point-to-point integrations of analysis and design, code manipulation, and CM tools. This initial integration effort was offered to internal customers as a monolithic (all or none) system. This approach met with very limited success.

Subsequent integration efforts have used a commercially-available message-passing framework as part of a more general solution. The resulting integrated tool set is offered to internal users as a set of unbundled products and services with phased availability. This approach appears much to be more successful. As the benefits of the integrated capability have become more apparent, specific projects have provided additional funding for the integration efforts.

Approximately 10-20 person-years have already been spent on the effort, including time spent learning and integrating tools and transitioning the technology to individual projects. Support is also provided to train and encourage modern software engineering practices. In addition, each new tool introduced is a significant investment, requiring another full-time person.

Organization D has found that the integration of a project's process and tools is essential for success. In recognition of this need, introduction of the integrated tool set into a project is highly customized. An on-going dialogue between the producers and consumers of the integrated capability is necessary to match needs to individual tools, relate tools to process, and determine how best to introduce the capability.

Other major lessons include:

- Frameworks reduce the effort necessary to integrate tools. However, users are generally not willing to pay for framework products.
- Multi-user support is a major problem. Many tools do not support groups of users well.For integration efforts to be successful, users must be able to tailor and enhance their integrated tool sets. However, customer tailoring causes problems with maintenance when problems arise, since it is sometimes difficult to determine the source of a bug.
- Licensing of integration technology and CASE tools to customers (even internal customers) can be a problem. The use of encryption technology and license servers can help in controlling access to integrated tool sets.

References

[Brown 92] Brown, A.W.; Earl, A.N.; & McDermid, J.A. *Software Engineering Environments: Automating Software Engineering*, McGraw-Hill (1992).

[Chikofsky 93] Chikofsky, E. *Computer-Aided Software Engineering (2nd Edition)*, IEEE Computer Society Press (1993).

[Long 91] Long, F. ed. *Software Engineering Environments (Volume 3)*, Ellis Horwood (1991).

[Thomas 92] Thomas, I.; & Nejmah, B. "Definitions of Tool Integration for Environments", *IEEE Software* V9, #3, (March 1992) 29-35.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for Public Release
Distribution Unlimited

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
CMU/SEI-93-TR-15

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ESC-TR-93-192

6a. NAME OF PERFORMING ORGANIZATION
Software Engineering Institute

6b. OFFICE SYMBOL (if applicable)
SEI

7a. NAME OF MONITORING ORGANIZATION
SEI Joint Program Office

6c. ADDRESS (city, state, and zip code)
Carnegie Mellon University
Pittsburgh PA 15213

7b. ADDRESS (city, state, and zip code)
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
SEI Joint Program Office

8b. OFFICE SYMBOL (if applicable)
ESC/ENS

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F1962890C0003

8c. ADDRESS (city, state, and zip code))
Carnegie Mellon University
Pittsburgh PA 15213

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO	WORK UNIT NO.
63756E	N/A	N/A	N/A

11. TITLE (Include Security Classification)
An Investigation into the State of the Practice of CASE Tool Integration

12. PERSONAL AUTHOR(S)
Jock Rader, Alan W. Brown, and Ed Morris

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month, day)	15. PAGE COUNT
Final	FROM TO	August 1993	39 pp.

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)
CASE tools
computer-aided software engineering tools
software development
software environments

19. ABSTRACT (continue on reverse if necessary and identify by block number)

Abstract: In the second half of 1992, a team of the SEI CASE Environments Project conducted a study into the state of the practice with respect to the operational use of integrated Computer-Aided Software Engineering (CASE) tools. After many false starts, we interviewed a number of examples of large organizations with integrated CASE tools in operational use on software development projects. Compared to the state of the art described in much of the literature, what was found might be considered modest. Compared to industry norms, it was quite impressive, representing significant commitment, ingenuity, and significant attention to end user needs.

This report details our observations and analyzes the current state of the practice of CASE tool integration as revealed by our study. It also speculates on reasons for the modest state of the practice.

(continued over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ■ SAME AS RPT ☐ DTIC USERS ■

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified, Unlimited Distribution

22a. NAME OF RESPONSIBLE INDIVIDUAL
Thomas R. Miller, Lt Col, USAF

22b. TELEPHONE NUMBER (include area code)
(412) 268-7631

22c. OFFICE SYMBOL
ESC/ENS (SEI)