Workshop on
Application Integration Architectures

Workshop Report

Workshop Organizers: Bob Hodges and Craig Thompson

Dallas Parkway Hilton, Dallas TX

February 8-12, 1993

Acknowledgments

The organizers of the Application Integration Architectures Workshop acknowledge and appreciate the high quality and open contributions of all attendees (listed in Appendix A).

Most especially we appreciate the efforts of breakout session scribes and spokespersons, and section authors and editors: E. Bailey, B. Balzer, B. Harrison, G. Hollowell, M. Imber, M. Law, F. Manola, B. Stucke, and J. Winkler.

We also wish to thank the National Institute of Standards and Technology (NIST) for publishing this workshop report and making it available to a larger audience.

Executive Summary

In the last several years, many organizations, including both standards bodies and industrial consortia, have developed a variety of software architectures for application and enterprise integration. These application integration architectures cover a broad spectrum of information technology (IT) including distributed systems, frameworks based on a backplane of services, integrated software engineering environments, repositories, and data management. For some time, there has been a need to insure that emerging de facto and de jure standards for integration architectures will work together.

During the week of February 8-12, 1993, forty individuals from twenty-five standards development organizations met in Dallas, Texas, to participate in the first Workshop on Application Integration Architectures. The objective of the workshop was to provide a one-stop-shopping forum where individuals active in standardization could meet together to address technical and management problems associated with the development of a shared vision of a common industry-wide integration architecture.

The workshop, coordinated and hosted by Texas Instruments, provided a neutral venue where diverse perspectives could be considered. While individuals attending were not "official" representatives of specific groups, their participation nevertheless served to provide a "big picture" of the standardization landscape of application integration architectures. Workshop attendees included individuals active in industrial consortia, military and government standards groups, and national and international standards bodies (see list below). Some of the groups focus on generic software architectures while others focus on architectures for specific domains such as software, electrical and mechanical design, and manufacturing. While many of the groups develop component standards or families of standards, others profile collections of standards to find or develop those collections that work together in an integrated manner. The workshop covered perspectives from both the standards producer and the standards customer. The operating rules, schedules, and scope of the groups vary, but their common interest is to set standards for the integration of integration technologies.

WORKSHOP CONTENT: The workshop began with the vision that standards for managing, sharing and using information assets "enable the integrated enterprise." Workshop participants identified goals leading toward this vision. The goals are to: (1) provide coordinated IT standards with minimal redundancy and a common vocabulary, (2) minimize the time and cost to establish new IT standards, (3) minimize the time and cost of producing standards-based IT products, and (4) minimize the time and cost of integrating those standards-based products.

Substantial time was spent in plenary sessions in which representatives of each group described their group's scope, membership, activities, liaisons, and schedule. This part of the workshop served to insure that each group was aware of complementary groups that might provide needed solutions to related problems. The remainder of the workshop took on a town hall flavor with breakout sessions addressing both technical and management topics.

The technical sessions focused on data and object models and on plug-and-play, compositional architectures. While there is a requirement to deal with legacy data models, many groups report moving toward object models for a wide range of enterprise needs in integrating information and applications. There is no single standard for object models, so several hybrid models are being formulated to add modeling power to subsume other models (e.g., the entity-relationship model). A major integration problem identified is the need to share enterprise information for different purposes in different object models. One group, the X3H7 Object Information Management Technical Committee, is acting as a focal point for comparing different standards' object modeling needs. They are working with other groups to develop strategies for evolving IT standards toward compatible, common perspectives on object-based concepts that would support improved interoperability of future IT standards-based products.

Workshop on Application Integration Architectures

In considering different architectures, the workshop began by noting that some existing IT standards seem to be monolithic compositions of more primitive standards and might better be separated. Most of the time was spent on plug-and-play architectures providing common runtime services. These integration architectures promise to make next generation applications easier to develop, since common services will be reused through a shared basis for specifying and requesting services. Problems identified with this approach are: (1) to get the most benefit, these architectures must be "open" to allow addition, improvement, or replacement of services, (2) careful integration is required to achieve openness, and (3) different standards groups and different IT products bundle overlapping collections of services. A first step toward integration is for groups to develop a profile of services and to compare these profiles.

Management sessions focused on ways to improve the effectiveness of computer standards development processes. Workshop participants identified the need for standards groups and industrial consortia to cooperate more effectively. Some of the roadblocks are: (1) the number of meetings involved, (2) openness of membership, (3) schedule, and (4) lack of understanding regarding different groups' missions and modes of operation. Believing that better cooperation could yield complementary, non-overlapping standards that will enable integration, the participants developed a model for interaction among accredited standards committees and industry consortia. The model suggests that consortia should become active members of the relevant standards development organizations, and that they actively promote "interim" standards they are developing in order to accelerate development of evolving "formal" standards. The model, it was agreed, does not require any new formal standards coordination organization, instead relying on improving the effectiveness of existing "liaison" mechanisms. As a first step, the group developed a snapshot of current standards work and began work on a roadmap to discover where communications paths among the standards and consortia needed reinforcement or realignment. The workshop provided the overview needed to encourage individuals to work together in forging better relationships between related groups.

WORKSHOP RESULTS: Participants recognized the need for continuing work that was started at this workshop. Several participants volunteered to contribute time or resources to support:

- A central catalog of groups, listing scope, work items, schedule, liaisons, and, where relevant, brief descriptions of each group's data/object models and services provided. With some analysis, a roadmap showing which group is producing what by when will be available to allow groups to better coordinate their efforts.
- A central calendar to allow groups to plan overlapping meetings.
- A second workshop, or "congress," tentatively planned for December 1993 to provide continuity and a second chance to build a shared "big picture."

AFFILIATIONS OF WORKSHOP PARTICIPANTS: Workshop participants included individuals working in the following groups or organizations (in alphabetical order):

1. Ada Joint Program Office (AJPO) Portable Common Interface Set (PCIS)
2. Advanced Research Projects Agency (ARPA, formerly DARPA)
3. CALS Industry Steering Group Information Integration Working Group (IIWG)
4. CASE Communique
5. CASE Data Interchange Format (CDIF)
6. DoD Ada Joint Program Office (AJPO)
7. DoD Corporate Information Management (CIM) Initiative
8. International Conference on Enterprise Integration Modeling Technology (ICEIMT)
9. ISO TC184/SC4 Standard for the Exchange of Product Model Data (STEP)
10. ISO/IEC JTC1/SC21/WG3 (Reference Model of Data Management)
11. ISO/IEC JTC1/SC7/WG11 (Description of Data for Software Engineering)
12. National Institute of Standards and Technology (NIST)
13. Network Management Forum (NMF)
14. North American PCTE Initiative (NAPI)
15. Object Management Group (OMG)
16. Rapid Response Manufacturing Consortium
17. SEMATECH
18. Navy Next Generation Computing Resources (NGCR) Project Support Environment Standards Working Group (PSESWG)
19. Unix International
20. USAF Integration Toolkit and Methods (ITKM) Program
21. X/Open
22. X3H2 (Database), ISO/IEC JTC1/SC21/WG3 (SQL DBL Rapporteur Group)
23. X3H4 (Information Resource Dictionary System), ISO/IEC JTC1/SC21/WG3 (IRDS Rapporteur Group)
24. X3H6 (CASE Tool Integration Models)
25. X3H7 (Object Information Management)

Table of Contents

Executive Summary ii
A. Introduction 1
B. Opening Plenary Session -- Workshop Objectives, Approach and Outcomes 2
C. Problem Definition 5
D. Objectives of Workshop Participants 7
E. Group Presentation Summaries 11
F. Technical and Management Breakout Sessions 12
F.1. Architectures Session I 12
F.2. Architectures Session II 13
F.3. Data/Object Model Features Session I 13
F.4. Data/Object Model Features Session II 15
F.5. Data/Object Model Features Session III 17
F.6. Data/Object Model Features Session IV 18
F.7. Domain-Specific Standards Session 21
F.8. Interchange Standards Session 22
F.9. Data and Control Integration Birds-of-a-Feather 25
F.10. Object Model Soup Birds-of-a-Feather Session 26
F.11. Standards Roadmap Session I 29
F.12. Standards Roadmap Session II 33
F.13. Standards Roadmap Session III 35
F.14. Management Session I 40
F.15. Management Session II 42
F.16. Management Session III 43
F.17. Management Session IV 44
G. Conclusions and Recommendations 46
Appendix A. Attendance Roster 51
Appendix B. Call for Participation 56
Appendix C. Workshop Program 59
Appendix D. Workshop Document Register 65
Appendix E. Group Information Templates 67
Appendix F. Workshop Evaluations 88
Appendix G. Participant Position Papers 93
Appendix H. Service Specification Template 121
Appendix I. Project Summary Repository 123
Appendix J. Consortia Standards Process Model 125

List of Figures

Figure 1 - Software Infrastructure Standards Landscape 4
Figure 2 - Object Model Features Matrix 20
Figure 3 - Domain-Specific Import/Export 24
Figure 4 - Generic Import/Export 24
Figure 5 - Interchange Architecture 25
Figure 6 - Message Interface and Data Interface Channels 28
Figure 7 - A Generic Framework Architecture 31
Figure 8 - Services Architecture Hypothesis 32

A. Introduction

The First Application Integration Architectures Workshop was held in Dallas, Texas, on 8 - 12 February 1993. Forty individuals from twenty-five accredited standards development organizations (SDOs) and industrial consortia attended.[1] The purpose of the 5-day workshop was to provide a forum where individuals active in one or more standards efforts or industrial consortia in the information technologies software area could meet to discuss how their efforts relate and work to formulate a roadmap to insure convergence of de jure or de facto standards in software information technology.

The first two days were spent reviewing each of the twenty-five different SDO/C efforts. The value was to provide participants with a much clearer idea of "the big picture" of how each group contributes to this goal. The final three days were spent in discussion sessions in which we identified technical and business/cultural/management roadblocks that can prevent convergence and interoperation of standards and worked on a roadmap to reach our goal. The structure of this part of the workshop was dynamic to insure our time resulted in productive outcomes.

The purpose of this report is to provide a faithful summary of results of the First Application Integration Architectures Workshop. It is not our purpose to be prescriptive except as far as some workshop sessions identify technical or management issues and recommend courses of action for others to follow. We recognize that different readers of this report will take away different conclusions and we hope the workshop and report provides a basis for acceleration of improvements in both the content and process of developing IT standards.

The report is structured to mirror the workshop structure: Section B describes the mission, approach, and outcomes of the workshop as outlined in the Opening Plenary Session. Section C provides a refinement by participants of the problem the workshop was addressing. Section D reviews in more detail why individuals attended the workshop and what their expectations were. Section E provides a listing of the plenary presentations. Section F describes a series of breakout sessions on object/data models, application integration software architectures, a roadmap for standards convergence, and management issues. Finally, Section G summarizes the conclusions and recommendations of the workshop.

Appendices provide the Attendance Roster, the workshop Call for Participation, the Workshop Program, the Workshop Document Register, The Group Information Templates, Workshop Evaluations, Workshop Position Papers, Service Specification Template, and information on accessing the Project Summary Repository.

---

[1] It is recognized that specifications important to industry come from both SDOs and consortia; in this report, these are collectively referred to as SDO/Cs.

B. Opening Plenary Session -- Workshop Objectives, Approach and Outcomes

Workshop organizers Craig Thompson and Bob Hodges welcomed participants to the First Application Integration Architectures Workshop. The workshop drew forty individuals active in one or more twenty-five accredited standards development organizations (SDOs) and industrial consortia in the information technologies (IT) software area. Because of this diversity, one workshop participant termed this "the mother of all workshops."

As this session began, we noted that workshop participants were acting as individuals, not as official spokespersons of groups, but that they were invited to the workshop because they play some active role in one or more relevant organizations.

We also noted that the workshop was intended to provide a "neutral forum" to allow individuals from different kinds of standards SDO/Cs to meet together. To keep the forum neutral, by design, no company nor SDO/C sponsored this workshop.[2]

The Opening Plenary Session suggested a strawman Program of Work for the workshop that would be refined during the rest of the workshop:

- scope
- problem the workshop addresses
- objectives
- approach
- metrics for success
- workshop products

SCOPE: The workshop targeted individuals active in SDO/Cs (including national and international accredited standards groups, industrial consortia, and government efforts) that are developers or customers of either component standards or profiles of cooperating standards in the software IT area. These IT software standards include data/object models, network, database, frameworks, repositories, change management systems, CASE, engineering, manufacturing, and enterprise information systems. Forty representatives from twenty-five SDO/Cs attended the workshop.

PROBLEM THE WORKSHOP ADDRESSES: The information technologies area is large, fast changing, and competitive. Standards are needed to allow industry and government customers of information technologies to build information systems with long-lifetimes, that is, to protect their investment. Several years ago, some organizations believed they could develop proprietary hardware platforms, then proprietary operating systems, then proprietary database management systems, repositories, and frameworks, then tools to fit into integration frameworks; now many organizations spend all their software resources on "glue" to integrate different software components and standards together. This is phenomenally expensive.

---

[2] Originally, the organizers requested OMG and X/Open to "sponsor" the workshop in order to attract participants; both organizations agreed; but after further consideration of the need for an unquestionably neutral forum, all parties agreed there should be no sponsor.

As organizations attempt to scale IT systems to become enterprise integration solutions, standards are increasingly needed. The sheer number of SDO/C organizations is daunting. Different groups see different "parts of the elephant"; they have different scope, time horizons, and reasons for existence. Consequently, there is no real guarantee (other than market forces) that systems of interoperable standards, that is, standards that will operate together, will be formed.

OBJECTIVES: The common objective of SDO/Cs is to develop useful component standards or suites of standards for use by some community or communities. Of importance are techniques for improving coordination between SDO/Cs, accelerating a consensus that leads to standards, improving standards, and reducing the time and cost of standards development.

APPROACH: The challenge of the workshop was not only to better state the problem to be solved and workshop mission statement but to suggest as much of a plan to ensuring mission success as possible. The workshop itself provided a way to view the IT standardization process from an overarching perspective. This kind of forum is valuable to insure overall convergence. Also, it provided an opportunity to explore and possibly initiate case-by-case collaboration that is needed whenever groups need to insure their standards will work together.

Our approach to ensure the workshop goals were met was to invite key people active in SDO/Cs and provide a neutral forum. Then, we challenged them to identify deliverables that would help insure SDO/Cs will converge and interoperate.

The workshop structure mirrored two top-down approaches to encouraging convergence of IT standards. First, the workshop was structured to provide a forum for groups to understand the standards landscape and to brief each other on their scope, objectives, status, plans, and liaisons. This occurred in the first two days of the workshop when each SDO/C reviewed its program or work in plenary session. This provided a rarely seen top-down view snapshotting the current state of the different IT SDO/Cs. Figure 1 provides one initial view of the technical landscape and how (some) IT SDO/Cs relate to it. Figure 8 on page 32 provided a refinement completed during the workshop. The workshop also provided a rare opportunity for SDO/Cs to gain critical, face-to-face review by a "jury of peers" when describing their SDO/C's workplan.

Second, the workshop was structured to identify technical or management roadblocks to progress toward interoperable standards and algorithms for cooperation and coordination that could lead toward this goal. This occurred during the breakout discussion sessions in the final three days of the workshop. An example of a technical problem is the "yet another object model" in which different groups are developing similar but different object models, which will cause data integration problems. An example of a management roadblock is the "creeping scope" problem, which occurs when groups expand their standards to cover overlapping technical areas. Better coordination can pool expertise, reduce duplication, and increase interoperation.

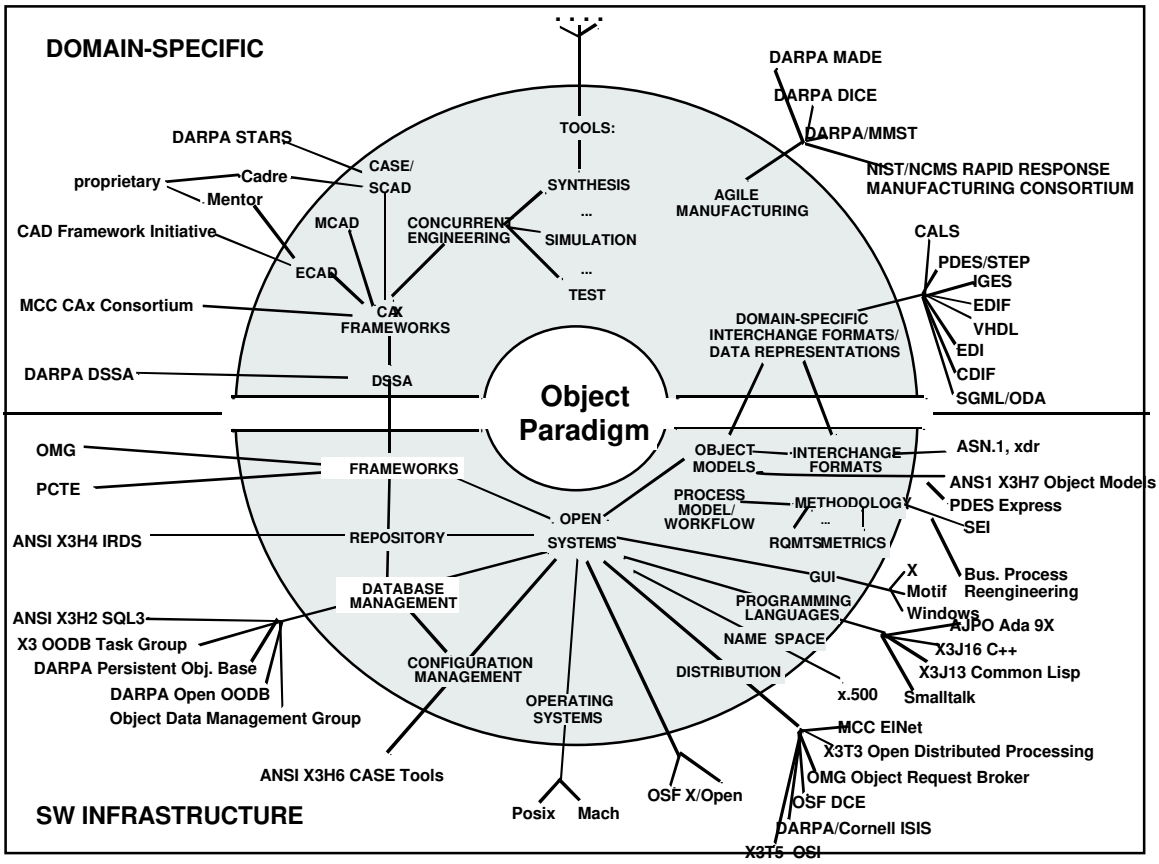


Figure 1 - Software Infrastructure Standards Landscape

METRICS FOR SUCCESS: We identified the following strawman list of "metrics of success" for the workshop:

- participation from SDO/Cs that cover the IT standards landscape
- work items and action items completed
- changes in approach or relationships (toward cooperation).

WORKSHOP PRODUCTS: We identified the following strawman list of products that might make sense as instruments of convergence.

- Updated Standards Landscape Map with additional groups positioned
- Template (descriptor) describing SDO/Cs (missions, activities, outcomes)
- Business Roadmap with Dependency and Convergence Possibilities
- Matrix of Domain Coverage
- Matrix of Data/Object Model Features
- Matrix of (Object) Services
- Template for an (Object) Service

We recognized that not all Workshop Products would be completed during the workshop but that the workshop needed to at least define the structure of the deliverables and a process for their completion.

C. Problem Definition

The purpose of this plenary session was to refine the workshop's Problem Statement by better stating the common problem we are trying to solve and the vision we hope to achieve. We began with the question, What do "customers" of standards development organizations want and which of their problems are we solving? We agreed that these customers need information (quantity, quality, added value). Today information is distributed, stored in many data representations, and governed by many tools and information management systems. Today's solution approaches to managing information are diverse and interoperate poorly. Industry spends a tremendous amount of money on "glue" that links heterogeneous software systems together.

Workshop participants agreed that their unifying theme is to achieve "the integrated enterprise." An enterprise is any set of resources, including people, organizations, and knowledge, that share or contributes to achieving a common goal, usually related to an economic endeavor. Modern actual or virtual enterprises often include many separate groups or companies that work as partners, suppliers or subcontractors to meet the needs of their customer, the key member of the enterprise. An integrated enterprise unifies these many resources into a smoothly functioning unit. Enterprise integration is the efficient coordination of the enterprise resources toward the shared goal with a minimum of duplicated or wasted effort.

All of the workshop's groups shared a common focus on the information that supports the enterprise processes. To integrate those business processes, the information that supports the processes must be integrated. Standards for managing, sharing and using the information assets of the enterprise are the enablers that allow an enterprise to integrate. Such standards and the information technology they support cannot achieve integration for an enterprise, but they enable it by allowing business functions to interact and coordinate more efficiently.

Specific benefits of this vision are:

- to enable sharing data and services across enterprise boundaries;
- to maximize the "plug & play" of products based on standards, allowing flexible combinations of products that work effectively together;
- to preserve investment in enterprise information systems based on durable IT standards;
- to minimize the time and cost of producing standards-based IT products;
- to minimize the time and cost needed to develop and establish new IT standards.
- to provide coordinated IT standards with minimal redundancy and a common vocabulary, i.e., standards designed to support one another; and

The generic strategy to achieve this vision, we agreed, will involve understanding the requirements, defining a software architecture(s), identifying components that populate the architecture, identifying interfaces between components, and providing standards for these interfaces. We noted that any solution approach must provide migration paths, must assume change is an invariant, must not assume a global standard, must not assume a single data or object model, and must be market driven. We also noted that, while many groups are moving toward object-oriented solutions, OO approaches are not panaceas, and any overarching approach must also handle existing (legacy) data and systems. Within these constraints, we want to induce a converging family of complementary, compositional standards.

Breakout sessions (described in Section F) were focused on ways to accelerate progress toward these goals. Technical sessions on software integration architectures, data/object models and services architectures all provided possible avenues for analyzing the content of standards to provide better ways to partition or align them. Management sessions addressed the processes used in developing standards and the ways that they might be improved to lower cycle time and total cost of arriving at usable standards that could support the development of integrated products. Roadmap sessions attempted to provide a map of the existing standards efforts to better understand how each SDO/C contributes to the enterprise integration vision.

D. Objectives of Workshop Participants

Near the beginning of the workshop, participants were asked to identify their objectives in attending the workshop and what they felt could be accomplished. This provided a gauge of whether our objectives were common and also helped us tailor the remaining workshop days to accomplish these shared goals. This section summarizes individual responses (from most attendees).

D.1. Bruce Speyer (ICEIMT--International Conference on Enterprise Integration Modeling Technology led by MCC) has been trying to build consensus around both technical AND business perspectives on Enterprise Integration for the past several years. He notes BOTH are required. He is encouraged by what he sees. He mentioned system integration for enterprises via EINET. Also that standards are starting to realize their role in integration. He noted that we do not need to eliminate all overlap. We do need to figure out where pieces fit and agree on integrating protocols. This is not an all or nothing situation. The hard part is agreement on meaning and semantics; we need to identify boundaries and agree on where agreement on meaning is possible.

D.2. Derek Kaufman (X/Open) wanted to talk to the repository folks and sort out confusing overlaps and conflicts. He believed we need to add a Problem Statement with focus or roles of groups shown on a Roadmap. (See sections C and F.11-13.)

D.3. Vic Goddard (X/Open) attended to listen and learn. He suggested that we need to take the mosaic of SDO/Cs and re-view these efforts as a composition of component standards that sum to cover enterprise needs.

D.4. Jack Bissell (UI--Unix International) restated the overall vision "by the year 2000, the enterprise is integrated". This includes integration of the application space, e.g., the vision is more expansive than CASE. He pointed at OLE, hotlinks, and other run-time integration strategies. He sees OO as glue for legacy integration. He recommends the OSF Distributed Management Environment (DME) model for solving this type of problem, i.e., an involving process of identifying and refining the problem statement and solution. Their technology is like a chaotic quilt that is constantly ripped apart and put back together.

D.5. Geoff Speare (OMG--Object Management Group) attended to learn what other groups are doing, to understand how to avoid overlap, and to look for opportunities to leverage others' work.

D.6. Barbara Cuthill (NAPI--North American PCTE Initiative) wanted to understand how object models and service architectures aid or hinder quality analysis. She wanted to understand the equation SUM(PCTE + OMG CORBA/OSA + NIST/ECMA Reference Model + IRDS)/4, e.g., how to converge these efforts.

D.7. Colin Ashford (NMF--Network Management Forum) was interested in the integration of Object Models for, especially, network and distributed information.

D.8. Jim Willits (NMF--Network Management Forum) was interested in spreading the NMF OMNIPoint 2 architectural vision; in architectural integration and coexistence of multiple object models; and in object services architectures, not just for networking and management interfaces, but also for data management, GUI, and CASE.

D.9. Bruce Murrill (NMF--Network Management Forum) was interested in user requirements. The NMF consortium, as a user of standards, wants to understand what is going on, including scope and timing of other efforts.

D.10. Edie Bailey (CASE Communique) hoped to educate workshop attendees about CASE Communique and also learn from others. She is a champion of the standards consumer. She believes that industry push is needed for standards to work together and is committed to convergence. Industry push provides speed. Standards keep everyone honest. She suggested a breakout on data/control integration and the separation. She votes "yes" on integration and "no" on object models (that is, she believes we need to support legacy data representations and that OMs are not a panacea).

D.11. Mike Imber (CDIF--CASE Data Interchange Format) hoped to educate others about CDIF and learn about complementary work, hoping to find holes and get inputs. He reports success in strong liaisons and reuse. In CDIF, a CASE information model and import/export are in progress. He hopes for a migration path and to get something useful in the short term, then incrementally improve.

D.12. Glenn Hollowell (SEMATECH) stated that SEMATECH is a consumer of software standards. He wishes to encourage interoperability of standards and has high hopes but low expectations. He would like to spawn an effort to define an interoperability model against which any standard system or implementation could be measured as a beginning of a standard suite for interoperability. Concepts need to be abstracted to a level independent of environment, language, or system. He is an advocate of OO, though notes that one can foul up an OO design just as badly as any other.

D.13. Misako Sterbenz (RRM--Rapid Response Manufacturing Consortium) states that Ford has found that integration is hard when you purchase all computer technology (COTS = commercial off the shelf). The union of what is there meets the requirements, but can't be integrated cheaply. She asks if there is hope in a reasonable timeframe?. She lists first order integration requirements as: migration, integrity, performance, and cost.

D.14. Neil Christopher (RRM--Rapid Response Manufacturing Consortium) notes we are all facing competitive pressures on company and national bases. Applications need broad access to process and product data and interoperability. He hopes the workshop will be a forum for validating and enhancing standards and consortia. He expects a toolset for integration, not one monolithic solution. He wants a roadmap for integration. He would like to collapse barriers between design and manufacturing. He reports that the MCC ICEIMT glossary is useful.

D.15. George Maney (RRM--Rapid Response Manufacturing Consortium) left the integration business because it became hugely expensive. He believes system development is disappearing. He can provide real-world data and experience for Bob Balzer's vision (see below).

D.16. Tom Ternus (CIM Integration Center at IBM) is, like the RRM project, a customer of standards. He needs to integrate legacy applications. He wants to apply OO concepts to integrate these. He must manage multiple models and transformations. He needs a common API (application program interface) for applications to be able to interact. He wants to hide networking so applications are not concerned with location details.

D.17. David Beech (X3H2 SQL) is interested in a two-way exchange toward something that brings the DBMS world together to be a better fit with other standards. He hopes for harmonization of DBMS, IRDS, and CASE areas, extended to distribution and areas of integration. He understands the tension between descriptive new standards and marketplace driven standards. He asks if we are expecting to influence direction or are picking the winner from the marketplace, and he suggests we should be mainly concerned with influencing, not picking the winner. He would like to restate our overall objective to be "to make system integration unnecessary". He is an advocate of OO (flexibility through encapsulation, inheritance, and polymorphism to extend in ways that are not intrusive, possibilities for reuse that did not exist before).

D.18. Jerry Winkler (X3H4 IRDS) stated that X3H4 is working on providing a harmonization strategy between groups, that is, a reference model for X3 that states the overall problem that is being solved, what each group contributes, how we can create and integrate the solution, and how we manage the evolution of that solution. OO is viewed as one useful technique.

D.19. Jack Liu (X3H4 IRDS and X3H6 CASE Integration Services) would like to converge standards that are solving the same problem. He wants to "stay home more" (commenting on the travel required in interacting in related standards groups, a very real problem). He believes customers of standards are confused by the sheer number of groups.

D.20. Bob Hodges (X3H4 IRDS) believes that re-viewing an IRDS as a repository that requires a collection of object services may provide an architecture that others in the community can agree on, avoiding problems of the past with a monolithic IRDS architecture.

D.21. Roger Burkhart (X3H4 IRDS) is involved in several committees, still looking for the holy grail. He believes it is wise to separate generic infrastructure from domain specific solutions. He counsels that we need to develop what that generalized model will look like. This will involve research effort, including work on logical foundations.

D.22. Bill Harrison (X3H6 CASE Integration Services) stated that his primary interest is integration architectures that support fine grained access to data with high-performance. There will need to be standards. He wants an architecture that crosses the performance spectrum. He feels we need OM richer than or different from classical OMs. If we accept today's OM now we will be in trouble later. He feels object services are being divided too finely. He wants to cut 5-15 years out of research-->standards cycle to get technology into the hands of customers before it becomes obsolete.

D.23. Frank Manola (X3H7 Object Information Management), wearing his research hat, is interested in distributed object management for heterogeneous object systems and in extended transaction facilities. He participates in X3H7 because we need to learn how different groups can handle interoperating objects. One work item in X3H7 is to complete a features matrix that will allow comparison of different OMs. Finally, as a customer of standards, his company shares all of the software integration problems others are mentioning. He is part of a team to identify a next generation corporate computing architecture. He wants to make software running the business more flexible to react as the business changes. This includes CASE, spatial data, legacy code, and business rules defined declaratively and enacted semi-automatically.

D.24. Craig Thompson (X3H7 Object Information Management, OMG, X3 OODB Task Group, ODMG) believes object services architectures hold promise. They appear to be configurable, improvable, scalable, provide a way to sort out overlapping activities among SDO/Cs, and provide a route to interoperable, compositional standards. He suggests that "reference implementations" that drive standards work are a good way to insure that standards will evolve that will interoperate (e.g., develop a common set of services that underpin IRDS, PCTE, OMG, SQL, OODBs, etc.).

D.25. Fred Hathorn (DoD CIM--Corporate Information Management Initiative) is committed to a standards based architecture for the DoD Integrated CASE procurement. He wants a Roadmap of standards with their roles and to eliminate duplication and find gaps. He wants to know how to converge PCTE, IRDS, POSIX, CDIF, ... He is not sure what vendors will deliver because their standards overlap. He councils groups to be willing to give up some of their turf.

D.26. Alan Brown (SEI/Navy PSESWG) sees the scope of the workshop as including frameworks and project support environments. He wants to compare service based collections of functionality with other taxonomies. He believes there is a relationship of ISEE/CASE to other service efforts. He mentioned it is important to manage our expectations.

D.27. David Carney (SEI/Navy PSESWG) reminded us that several years ago STARS and others identified the same framework problems. Integration is a hard problem. Any methodology that can help should be considered.

D.28. Larry Johnson (CALS IIWG--Information Integration Working Group) stated that single company total solutions are rare and so open systems are needed. His group's old mission was an enterprise gateway, its new mission is to develop a profile for enterprise integration. There are business AND cultural issues. We need an integration framework.

D.29. John Solomond (AJPO--Ada Joint Program Office) is program manager of Portable Common Interface Set (PCIS). He wants to understand what object management services are needed beyond PCTE including software management services for a software environment.

D.30. Gio Wiederhold (ARPA--Defense Advanced Research Projects Agency, now ARPA) was interested in insuring that ARPA is investing in the right efforts. ARPA provides inputs to plans under the new administration. He asks if new consortia are needed? Technically, he sees a need for object views, otherwise they won't scale, also for declarative representations. He noted that scaling and integration don't happen automatically.

D.31. Bob Balzer (USC/ISI, ARPA DSSA--Domain Specific Systems Architectures) attended as a consumer, more interested in identifying integration mechanisms than in developing standards. He is involved in the ARPA Domain Specific Systems Architectures program, which is targeting not CASE, but application integration. He is not as much interested in a vision where people integrate systems as in how programs will integrate programs. Automated development must churn out applications by pulling together pieces in an automated way. He wants to see more declarative information to allow reasoning about systems by programs. He wants to see us building software from end-user specifications, and thinks this will happen first in limited domains. He is serious about the objective of "putting system integrators out of business." We do not need a single monolithic standard. A variety is ok if they are well specified. The hardest social problem is egocentric viewpoints, e.g., having each group accept that their piece is not the centerpiece.

D.32. Tom Rhoades (NIST--National Institute of Standards and Technology) attended to understand the scope of existing efforts. His goal is Integrated Software Engineering Environments (ISEE), their components (services) and gaps. He hoped to begin the convergence process (which he sees as ambitious) via common coordination between workshop attendees. Like ARPA, he is looking for promising approaches for NIST.

D.33. K.C. Morris (NIST--National Institute of Standards and Technology) described the PDES/STEP work on semantic integration and stated that their API is hard to scope because of the diversity of standards.

D.34. Elizabeth Fong (NIST--National Institute of Standards and Technology) is interested in interoperation issues in system architectures. She notes it is and will remain a heterogeneous world: mappings, standards, maybe OO is a solution approach.

D.35. Margaret Law (NIST--National Institute of Standards and Technology) notes that the U.S. government is the largest enterprise and customer for enterprise integration. She believes standards are a promising approach to the problem. She wants to better understand dependencies between standards and avoid overlap. She wants to insure that standards interoperate and that there is a migration path for legacy systems and data. Maggie has previously participated on many different IT standards groups.

E. Group Presentation Summaries

Major portions of the first two days of the workshop were devoted to introductory presentations on each of the groups represented at the workshop. The purpose for this use of workshop time was to insure that attendees developed a clear idea of the charter, progress, plans, roadmap, and liaisons of each group. A small amount of time was also devoted to informal discussion of participants' knowledge of the groups with no attending representative. Each presentation was allocated twenty minutes, including any questions and discussion. Copies of presentation materials were made available to the workshop attendees following the presentations (see the document register in Appendix D).

Feedback from workshop participants indicated that the presentations were a valuable part of the workshop program. Many suggested, however, that in future workshops this type of introductory information should be conveyed through written material to maximize the time spent in working sessions. One of the workshop follow-up actions is to establish a database of information about groups in a common "resume" format that could be used to establish this baseline of awareness (see group template in Appendix E).

The following list identifies the activities that were presented and the speakers.

1. International Conference on Enterprise Integration Modeling Technology - Bruce Speyer
2. X/Open - Peter Janecek
3. Unix International, ATLAS, DADSIG - Jack Bissell
4. Network Management Forum - Bruce Murrill
5. OMG - Glenn Hollowell
6. PCTE - no presenter
7. CDIF - Mike Imber
8. ISO SC7/WG11 Information Model for Software Engineering - Mike Imber
9. ISO SC21 WG3 - RMDM - Liz Fong
10. X3H2 SQL3 Project - David Beech
11. X3H4 IRDS - Jerry Winkler
12. X3H6 CTIM - Bill Harrison
13. X3T3 - ODP no presenter
14. X3T5 - OSI no presenter
15. X3/SPARC/DBSSG OODB Task Group - Craig Thompson
16. X3H7 Object Information Management - Frank Manola
17. DoD CIM/NAPI - Fred Hathorn
18. CALS IIWG - Larry Johnson
19. DARPA Programs - Gio Wiederhold
20. DARPA DSSA Domain Specific Program - Bob Balzer
21. SEI and Navy PSESWG - Alan Brown
22. NIST/ECMA Reference Model for Software Engineering Environment Frameworks - Margaret Law
23. Integration Toolkit and Methods Program - Brian Stucke
24. ODMG - Craig Thompson - brief status report
25. CASE Communique - Edie Bailey
26. Standard for the Exchange of Product Model Data (STEP) - K. C. Morris
27. Rapid Response Manufacturing - Neil Christopher
28. SEMATECH - Glenn Hollowell

F. Technical and Management Breakout Sessions

F.1. Architectures Session I

Participants: Bob Balzer (presenter), Bob Hodges (scribe), David Carney, Barbara Cuthill, Elizabeth Fong, William Harrison, George Maney, Geoff Speare, Jim Willits.

This session focused on identifying types of software architectures. The participants in this session established a definition of architecture and explored the dimensions of different types of architectures as they contribute to integration. Through the classification of different examples of integration architectures, we began to lay the groundwork for using architectures as a basis for partitioning and integrating standards contributions.

Architecture, in its general meaning, is defined as a "coherent structure that positions components and their interrelationships (how they fit and work together)." There is a distinction between a generic architecture that establishes rules for component selection and interaction and a specific architecture that is an instance of the generic architecture. A generic architecture defines an open-ended class of possible systems that comply with the rules of the generic architecture. Each specific architecture may also add rules related to the particular components included. When discussing architectures, some confusion may be related to the mixing of generic and specific architecture contexts.

A taxonomy of architecture types can be based on the "integration style" that is employed. The following partial listing are the integration styles identified during the session discussion:

- Invocation-Based Integration Styles
  - procedure request
    - requester directly specifies what function is to be invoked
    - no decision required by the service provider
    - synchronous responses
  - message/method
    - requester specifies a service in terms of an interface
    - service provider determines the method to invoke based on object model concepts of inheritance and polymorphism
    - responses may be synchronous or asynchronous
  - event-based
    - notifications of events may be received by zero, one or more service providers
    - decisions on invocation of functions determined entirely by notification recipients
    - no responses are expected or provided
  - schedule-based
    - functions are invoked through time-based triggers
    - no direct communication between integrated components
    - may be combined with data integration (e.g., batch processing by scheduled jobs against shared database)
    - no interprocess communication
- Data Model-Based Integration Styles
    - components are interrelated through access to common data
    - no direct communication between processing components
  - single shared repository
    - components integrate through shared schema and data
    - logically unified schema defining meaning

federated repositories
components integrate by sharing data managed in distinct repositories
repositories may have different schemas defining meaning
blackboard
components exchange data through a shared temporary data store
components supply interpretation of meaning of exchanged items
interchange format
components exchange data through a neutral interchange format designed for storage and transport
tools must share understanding of meaning
schema may or may not be explicitly included in interchange format

Although these classifications may serve to characterize the primary style of different integration architectures it left open questions about the role of architectures in relation to the role of object models in supporting application integration. This topic was considered in Architecture Session II.

F.2. Architectures Session II

This session focused on the relationship of the architecture to the underlying object model. Detailed notes about this session are included under the Data/Object Model Session II report.

F.3. Data/Object Model Features Session I

Participants: Frank Manola (presenter/editor), Glenn Hollowell (scribe), Colin Ashford, David Beech, Alan Brown, Roger Burkhart, Glenn Hollowell, Margaret Law, Jack Liu, K. C. Morris, Michael Richardson, Geoff Speare, Misako Sterbenz, and Gio Wiederhold.

After a period of discussion about what was actually technically achievable during the two days of breakout sessions, it was decided to start by taking up the topics suggested by the Workshop organizers. Those topics were:

- Why "Yet Another Object Model"? Are different object models needed for different purposes?
- Classification of features of object models
- Core model proposal (what features should be in a common object model)
- Deliverable: Object Model Features Matrix

Each of these subjects was discussed during the Data/Object Model breakout sessions, although to different levels of detail.

The group initially considered the question of whether different object models were needed for different purposes. It was generally agreed that different object models really were needed for different purposes, but with caveats on this general conclusion. Specifically, there was general agreement with the idea that there was probably a core set of object model features that most people would agree on, and that some mechanism for structuring object models into the "core", plus additional features required for specific purposes, would be useful (although some questions were raised about the need to integrate object models from different domains). There was also general agreement that some differences among object models were *not* essential, and could potentially be eliminated. Examples of "structuring" mechanisms mentioned were the "Core + Components" approach taken by OMG, a "layering" approach, and a "RISC-like" or "metamodel" approach for the core, such as that described in Frank Manola's position paper [Manola, 1993], in terms of which various extensions could be described (references are listed at the end of these minutes). A complication for object model rationalization efforts is that new object models are being defined all the time, and that any "rationalized model" or "core" must be one that has practical

mappings to major/important object models used in implementations today. It was also suggested that there would not be much use in defining yet another "core" model if it was just to be a "least common denominator" (LCD) model (i.e., one in which, unlike the "Core + Components" or "RISC" ideas, it would not be possible to also describe more complex facilities found in "real" object models).

There was considerable discussion of the OMG "Core + Components" approach. One question raised was, "Why not layering?" of object model features rather than the Core + Components approach. There was an extended discussion about the problems of compliance in the layering approach. Points were raised about "what do you do when there are features in a lower layer and upper layer that are appropriate to your application or product, but features in the middle layer that make no sense to support for that particular type of system?" Hollowell reported that the OMG Object Model Task Force first took a three layer approach to developing OMG's Object Model standard. The issues of compliance, non-compliance, and partial compliance with the layering approach was a primary cause of much delay in the development and consensus process. It was only after the "Core + Component(s) = Profile" approach came about that work progressed beyond the issues about very diverse domains having to comply to a comprehensive common object model.

There was significant misunderstanding about how the OMG's object model approach worked and there was a period of explanation of the concept contributed to by several people in the breakout session who understood the concepts well. In summary, the OMG object model approach is:

1) The Core is a consensus of the least common denominator of atomic features that the OMG membership feels must be supported by any system that calls itself "object technology." Those features are identity, typing, operations, and subtyping/inheritance.

2) Components are additional atomic features that may be needed in some application domains, but not others. Examples of such features are exception handling, attributes, and relationships.

3) Profiles are composites of the Core, plus one or more Components, that make up a useful object model for a specific domain. These domains can be technology specific (DBMS, GUI, programming language, etc.) or application-specific (manufacturing, finance, etc.).

4) Only Profiles make up useful object models, and compliance will be measured against a Profile. Compliance to just the Core or individual Components that don't make up a Profile is irrelevant.

In the ensuring discussion, some felt that the OMG "core model" was more than they would consider an LCD model. The point was also made that a model in which "components" were required to be atomic, and there was only one way of expressing a particular feature (e.g., exceptions), as in the OMG approach, would be very different from a model in which "components" could express, e.g., something to be subtracted from another component, or in which there might be alternative components for the same feature. It was observed that the OMG approach had not really been tested yet, and it was not clear how robust it was. However, there was general support for trying some approach like it. It was noted that the SQL3 object model was being divided into something like a "Core + Components".

The group then turned to a discussion of object model features. The group had in hand two documents that could contribute to this discussion, the X3H7 object model features matrix, and Colin Ashford's report comparing the OMG and ISO/CCITT object models [Ashford, 1993] (both had been distributed at the workshop). It was also noted that a comparison of object models had been done in conjunction with the DARPA sponsored Persistent Object Base program, and that a report had been produced describing those results.

The group decided to use [Ashford, 1993] as its basis for comparing object model features, since its comparison of models was at more of a summary level. This made it more suitable for the brief time available at the workshop, since it did not require going through the detail contained in the X3H7

document (and since there was little documentation available at the workshop for detailed comparison of object models). It was noted that Ashford's comparison had been made against an interim version of the OMG object model, and that some of the entries needed to be updated to be consistent with the version of the object model approved by the OMG membership. Accordingly, it was decided to approach the comparison by first updating the OMG entries to take into consideration the current OMG model, and then adding columns of entries for the SQL3 object model, as well as other object models known to members of the group. Generally, Ashford's comparison proved to be very useful to the group as a foundation for this process. The results of this process, which continued through several successive sessions of the Data/Object Model track, are shown in the table on page 20. (The entries for models other than OSI/CCITT, OMG, and SQL3 should be considered as "tentative" until verified by experts on those models).

A discussion occurred on exactly which aspects of the comparison tables should be considered as really intrinsic to the object model. For example, it was suggested that things such as whether "multiple replies" or "events" were supported were at a different level from more fundamental aspects of an object model. The question was also raised as to whether specifications of allowed messaging activity belonged in an object model or not. A comment was made that those working on the object model in the ISO and X3T3 work on Open Distributed Processing (ODP) were attempting to decide if events were a separate capability, or just "reversed operations". The discussion of exactly what parts of an object model specification ought to be considered fundamental to the model, and what parts should be considered elsewhere, e.g., as object services or parts of the architecture, was continued in later sessions.

F.4. Data/Object Model Features Session II

During this session, members of the Architecture II breakout session joined the Data/Object Model session to discuss invocation mechanisms (for objects and services), and the general relationship between an object model and its architecture (e.g., what role the object model played in describing services, and the boundary or distinction between the object model and object services provided within a given architecture). No attendance list was circulated, but essentially all of the people from the morning Object Model I Breakout Session returned and approximate total attendance of the combined groups was 20 people.

During the preceding plenary session, the Architecture group had noted that the differences between method and event invocation were based on differences in dispatching (mapping messages to the code that will handle them). In method invocation, the object model disambiguates the overloading of operation names ("ad hoc polymorphism"), while in event handling the disambiguation comes from event handlers registering themselves with a dispatching element (and multiple event handlers can register themselves for the same event). Implementing changes in the desired responses to messages has to be handled differently in the two approaches. Each mechanism can be used to implement the other, but the mechanism affects the organization of elements in the architecture in a particular way. Some of the issues involved seemed to be best described as part of the object model, while others seemed to be best described as part of the architecture. The problem was in deciding which was which. There was general agreement that a clean separation was desirable, for example, in order to be able to replace one object model with another in the same architecture. It was recognized that some of the aspects of an architecture came from the object model, while other aspects were applied on top of the object model.

The session discussions began with a strong statement of preference by one of the architecture breakout group for an event-based environment over conventional object technology's messaging techniques. The discussion that followed centered on defining the differences and similarities of the event-based and message/response paradigms. The concepts required for the event-based approach were described as a system that provides a registration mechanism allowing notification of events that are of interest to a specific function, along with a request/reply mechanism.

An assertion was made and supported by several people that an event-based approach and object technology are not mutually exclusive. Colin Ashford described some details of the concepts used in the OSI/CCITT model and its associated architecture to demonstrate that the OSI/NM model incorporates the notions of both event notification and message/response through its "managed objects concept". In that architecture, notification-handler "agents" collect notifications (events) emitted by objects. The agents in turn notify "managers". The managers respond to events by invoking operations on the objects (via their agents). There is a need to define how an individual object's behavior interacts with the coordinated activity of the system as a whole. Manager components depend on objects "knowing" that they have to emit certain types of notifications.

Discussions then turned to issues concerning boundaries between an object model and object services and what role the object model plays in services. The discussion ranged through categories of object services that included atomic services, meta-level descriptions of services and system level managers for coordinating object services. There was a position expressed that there must be a unified view of managed and unmanaged objects. It was asserted that all requirements for both (such as operations invocations, event notification, attributes, and behavior) can be generalized into a single set of semantic concepts regardless of whether objects are managed or autonomous. Further, that OSI/NM can be accommodated in a generalized model that serves many other applications.

A specific discussion ensued about whether or not operations and events could be "folded together" in some way. Some felt events could be folded into operations, essentially by adding operations to define notification (i.e., that a given method should receive certain invocation messages) and by distinguishing between messages sent to a distinguished recipient and "broadcast" messages. Others felt that operations should be folded into events (e.g., by considering an "operation" as a macro of a start event and the notification of interested parties when the operation had completed). Some felt that many services behaved a lot like operations, but that some do not, and that events might be more general.

Discussions then turned to "what is an object model versus what is architecture?" It was suggested and supported by several people that the OSI/NM model is really an architecture, not a model, because it describes the interaction between objects and rules of how they are controlled. It also was strongly contended that OSI/NM is both an object model and architecture because an object model is nothing more than a description of the structure and behavior of objects in whatever role they play--thus, there is no conflict between object models and architecture.

The session wrapped up with a series of somewhat disjointed statements by various people about their beliefs about object technology and architecture. They included:

- The OSI/NM model is nothing more than an extended OMG CORBA model.
- There should be a minimalist model to describe architecture.
- There should be very small atomic pieces that can be used in building an object model. Even OMG's Core object Model has too much specificity in some areas and not enough in others.

Several people suggested that it would be very difficult to have a "minimalist" object model; that certain aspects of what might be considered "implementation" were sometimes required as first class model concepts for specific applications. For example, object models used for analysis or design did not want a lot of things (like encapsulation) that object models used in programming languages would absolutely require, because they were irrelevant at the analysis and design level. This sort of thing was viewed by some as a clue that it might be impossible to define an "LCD core" that was simultaneously a complete object model in its own right. Others noted that they did not want an overly primitive "core" embodied in tools or facilities used in system implementation, since they did not want to have to build systems in terms of a base that was too primitive. Instead, they wanted their model to have the specific facilities they needed defined as first class parts of the model.

F.5. Data/Object Model Features Session III

Participants: Frank Manola (presenter/editor), Glenn Hollowell (scribe), Colin Ashford, David Beech, Roger Burkhart, Misako Sterbenz.

In this session, Colin Ashford went over some of the details of the OSI/CCITT model for managed objects, as an example of a model that differed somewhat from the sorts of object models used in object-oriented software development. This model was then compared with the OMG core model and the object model being developed for SQL3. Objects in the CCITT model are described by specifying separate templates for attributes, operations, and notifications. Templates may be related by inheritance. Templates are then grouped into "packages". An object type definition specifies the mandatory packages to be included in objects of the type, plus optional packages. Inclusion of an optional package in a particular object is determined by a predicate. Thus a given object of a type includes all the mandatory packages, and may also include optional packages. Operations, notifications, etc. are in some sense treated as "subobjects", which are composed to form the regular objects of the model. A notification specification is a definition of a notification that an object may emit, and is specified in terms of a signature, together with the conditions under which the notification will be emitted.

The model does not support subtyping in the sense of substitutability. Instead it supports what is called "allomorphism", a limited form of subtyping in which, under some circumstances, an object of a subclass can be used in place of an object of a superclass. This specialized form of subtyping is used because the model does not enforce the regularity of object structure required by the usual notions of substitutability. The model supports "variability". For example, a notification can be specified with a parameter specified as ANY DEFINED BY, and pass an object to be examined at run time--essentially, an object of a new, one-of-a-kind type. In such cases, the recipient is responsible for understanding the object passed to it.

This sort of structural flexibility (and less-rigid typing) is required because one managing system may have to manage many managed systems (e.g., purchased from different manufacturers). The objects typically represent pieces of hardware (like telephone switches), and thus resemble configurations of components in a CAD or CASE design. The ability to treat objects as configurations of mandatory and optional collections of components allows the model to treat objects that are very different structurally as being of the same type, while allowing it to deal with the differences as well. A more conventional object model would require a large number of subclasses to deal with the same problem (since every little variation might require a separate subclass definition).

The discussions following the OSI/NM presentation revolved around three issues:

- The need to resolve "openness requirements" for the OSI/NM object model and the need for interoperability with other applications.
- The need for mixed binding (early whenever possible, late when necessary).
- The definition of scope and how it is determined in various object models.

It was noted that a more flexible aggregation mechanism would have helped the CCITT model, and that current object models were too rigid in disallowing type changes (as objects change). This is probably due to the desire for static type checking in programming languages. The relationship of these facilities to "schema evolution" in SQL3 (and database systems generally) was mentioned. It was noted that at least the CCITT model provided an upper bound on what might be in a given object, and that there was work in the type theory literature on related facilities.

With this as a background, the group went on to complete the features matrix it had been compiling throughout the sessions, including tentative entries for some of the models.

The group then heard a presentation by David Beech on the SQL3 type system. He noted the role that SQL3 would play in some types of interoperability, since many applications currently communicate via shared databases. SQL3 is borrowing some aspects of its type system from C++, e.g., the use of explicit "ref" types. However, it is not borrowing all aspects (e.g., "friends"). SQL3 objects can either contain other objects or contain references to other objects. Referenced objects cannot be destroyed, leaving dangling pointers. The effect of the generalizations in SQL3's type system is to divorce it from a tight dependence on the relational data model. The relational model is now referred to as a "tightly disciplined subset" of the SQL3 type system. SQL3 currently lacks a universal supertype (like type OBJECT; EXPRESS has ANY for this purpose). Having a universal supertype is useful in being able to form, e.g., collections containing objects of any type. The most general aggregate in SQL3 is a multiset; basically a table. Set and list are specializations of multiset.

The subject of whether there should be separate type and implementation hierarchies (distinguishing subtyping from inheritance) is currently being debated. Beech said he preferred not to separate the two, and to distinguish multiple implementations using subtyping alone (he cited an argument by Bertrand Meyer, the developer of Eiffel, in support of this), but others disagreed. Another issue over which there is controversy is whether it should be possible to define abstract data types that do not have object identifiers (and thus which act as values rather than objects). This facility is currently in the specifications, but Beech opposes it.

An important issue in the SQL3 definition will be making sure it is closed; i.e., making sure it is possible to determine the result type of any query, and that the type of a query result is a type defined in the language/model. Closure is an important characteristic of the relational model, and this should be preserved in the object extensions, since closure makes it possible to use the results of one query as the basis of another (as in nested queries), and makes it possible to use the query language in defining views.

F.6. Data/Object Model Features Session IV

Participants: Frank Manola (presenter/editor), Glenn Hollowell (scribe), Colin Ashford, David Beech

In this session, the group discussed two topics, the distinction between object models and object services, and mechanisms for "reconciling" differences between object models.

In the discussion of object models versus object services, it was generally agreed that this was sometimes a hard distinction to make, as had been noted in several of the previous sessions. It was observed that object model definitions sometimes described things that might, at least by some, be considered parts of the architecture, in order to fully describe the assumptions that were being made about how the model would be used. Examples of things that might be in object model specifications, but might also be considered as aspects of the environment or architecture instead, are built-in types, run-time type objects, or pre-defined objects like the ODP Trader or a Transaction Manager. Such things help define or constrain the run-time environment in which the object model is expected to function, but might also be considered to help describe aspects of the objects themselves, which presumably is what an object model is supposed to do. For example, the knowledge that an object in an ODP architecture must be prepared to access the Trader to obtain a reference to another object helps describe the objects in that architecture, and thus might be considered to be part of the object model.

Another example of a facility that can be considered either as part of the model or a service in the architecture is events. Events can be defined as part of the object model as either a built-in distinct type of communication or as a special case of messaging. Alternatively, an "event service" can be defined as a specific component (object) of the architecture. Objects would send "notification messages" to the event manager object indicating that specific things have happened, and other objects would send "registration messages" to the event manager object indicating that they wish the event service to forward certain types of notifications to them under specified circumstances. When events are defined as a service, the object

model does not necessarily reflect the existence of events at all, and one could build another system not using events with the same object model. OMG is looking at both these approaches for incorporating events.

The distinction between object models and services is complicated by the fact that object models intentionally blur the difference between built-in and user-defined aspects of the system. The OMG "Core + Components" approach was discussed again, and it was again suggested that the approach might be more flexible if it allowed "negative components", as opposed to just atomic, non-overlapping components. However, then it might not serve its intended purpose in facilitating interoperability.

Another complication is that interfaces to services may be hidden or visible at a given point. For example, a persistence service may have an interface that is visible to an application programmer, as in a conventional DBMS API, or hidden from an application programmer, as in a programming language supporting "orthogonal persistence", in which case the interface to the persistence service may be visible only to the compiler (or compiler writer).

The group concluded that it was desirable to make a clean separation between object model and services, but that, in defining a model, it was necessary to be able to record assumptions about the environment, or dependencies between the model and the environment, someplace. This could be done by considering the model as part of a "configuration" of components, and using a separate specification of the configuration itself to record relationships between the model and other components of the configuration. In many cases, such as that of the OMG specifications, the definition of what is called the "architecture" serves as this configuration specification. The group felt that this provided a useful way of clarifying the roles of the architecture specification and that of the object model within it.

In discussing reconciliation mechanisms, the group went over proposed mechanisms for reconciling differences between the CCITT and OMG object models described in [Ashford, 1993]. The three mechanisms identified were:

- align the models; for example, OMG might provide a component providing the features necessary to support the features required in the network management application.
- provide run-time mediation between implementations of the models; for example, software might be developed to handle the notifications, run-time object structure determination, and multiple replies supported by the CCITT model.
- provide notational mapping tools; for example, provide compilers that would translate between specifications of object types in the two models.

The specific reconciliation techniques suggested for various aspects of the OMG and CCITT models are described in Section 3.3 of [Ashford, 1993]. The group also went over the approach suggested in that report for an integrated architecture using aspects of both the CORBA and CCITT managed object approaches.

References (both distributed at the workshop)

[Ashford, 1993] Colin Ashford, ed., "Comparison of the OMG and ISO/CCITT Object Models", report of the Joint Forum/OMG Taskforce on Object Modelling, Draft Final, Feb. 1993.

[Manola, 1993] Frank Manola, "The Need for Object Model Interoperability", position paper submitted to the Workshop on Application Integration Architectures, Dallas, Texas, Feb. 1993.

Concept	OSI/CCITT	OMG	SQL3	EXPRESS	PCTE	ATIS
Interoperability	syntactic & semantic levels	both levels	both levels	?	?	?
Reusable Components	library/ catalog of mgmt. info	type library	schemas have dictionary info.	schemas	schemas	yes
Encapsulation	yes	yes	yes	no	no	yes
Object Operations	yes	yes	yes	no	no	yes
Behavior	yes	yes	yes	no	no	yes
Attributes and Attribute Operations	yes	not in core object model	yes	yes	yes	yes
Object Types	quasi-type hierarchy supported by multiple inheritance	type hierarchy plus multiple inheritance	yes	yes	yes	yes
Direct Selection	by disting-uished name and association	by object identifier and assoc.	by id and assoc.	by id and assoc.	name and path	oid
Non-objects	yes	not yet	yes	?	?	?
Intended Use	distributed network manage-ment	OO software develop-ment	OODBMS and assoc. software	data modeling	tool integration	tool integration
Multiple Replies	yes	not discussed	yes/no	n/a	?	no
Events	yes	object service	no	n/a	?	?
Variation	yes	no	no	yes	?	no
Associative Selection	filters	filters	yes	yes	?	no?
Associative Selection Scope	subtrees of tree of managed objects	arbitrary scope	arbitrary	arbitrary	?	n/a
Specification	piecemeal	core + components	monolithic	monolithic	?	?
Specification Tools	GDMO	IDL	SQL3	EXPRESS	?	API
Specification Formality	syntax	syntax	syntax	syntax	?	?
Object Reference	by name	by oid	by oid and name	entity	?	oid

Figure 2 - Object Model Features Matrix

Object Model Features Comparison

Table 1 is the result of a high-level comparison of object model features done during the Data/Object Model breakout sessions. It is based on work reported in [Ashford, 1993], updated to take into consideration the current OMG model, and adding columns of entries for the SQL3 object model, and other object models known to members of the group. The entries for models other than OSI/CCITT, OMG, and SQL3 should be considered as "tentative" until verified by experts on those models.

F.7. Domain-Specific Standards Session

Participants: Mike Imber (presenter and scribe), Roger Burkhart, Fred Hathorn, Mike Richardson, Brian Stucke, Craig Thompson.

This session discussed the topic of domain-specific standards and how they fit into the area of generic standards. It was agreed that there were two different types of standards that could be considered to be 'domain-specific'; those which provided "enabling" facilities or support too many areas of government and industry and those that were targeted at a very specific segment of industry. Examples of the former "enabling" category include CASE and CAD/CAM, in that they are domain-specific in terms of their focus and coverage, but provide support for a wide spectrum of systems across all of government and industry. Examples of the latter include industry-specific standards and consortia such as EDIF and POSC.

Examples of "Enabling" and "Industry-specific" standards and consortia

"Enabling"

- CASE — CDIF, CIA, CASE Communique, PDES/STEP, NGCR PSESWG
- CAD/CAM — CFI

Industry-specific

- Semiconductor — EDIF, SEMATECH
- Petrochemical — POSC
- Product Manufacturing — PDES/STEP

It was agreed that what all the different domain-specific groups (of either type) are trying to do is to provide a representation of the concepts that are required to express information in their domain. They need a "shared" (i.e. agreed) definition of the semantics of the domain-specific information to be able to integrate tools and repositories. Ideally these should be produced as abstract information models, which can then be delivered using multiple concrete delivery mechanisms, such as 'content modules' or schemas for repositories and definitions for interchange formats.

Also, any domain-specific operations will need to be defined; these should take into account that many of the operations required on domain-specific objects may be generic operations - these should not be defined on a domain-by-domain basis, but generic operations should be used. It was agreed that we should encourage those working in domain-specific areas to understand what general services were available, and to assume their provision. Where they had not been provided, groups developing generic services should be encouraged to add them, rather that have them defined by many domain-specific groups.

It was identified that some domain-specific groups may have requirements that have not been met by the groups defining generic standards, but nevertheless, these requirements covered more than one domain;

examples include real-time requirements and the need for materialized views. In these situations, they should be encouraged to feed the requirements to the relevant group responsible for generic standards, rather than defining them themselves.

The subject of the sharing of definitions where domains overlap is a difficult one, and groups need to be aware that the domains they are working on may have overlaps with others, and to ensure that relevant liaisons are established so as to avoid conflicting definitions being generated by different groups. The following guidelines were developed for consideration by groups responsible for developing standards that are domain specific, requiring the modeling of the objects of interest in their domain.

- Existing generic modeling techniques such as CDIF and EXPRESS should be used where possible; new notations should not be developed. Experience and tools have been developed within the communities using these notations, which will save significant effort and speed the delivery of a new domain-specific area.
- The existing techniques should be examined to ensure that they have adequate representational power for the degree of expressivity required for the domain; examples include the definitions of cardinalities on relationships and constraints.
- Where a domain-specific group identifies special needs not met by any of the existing modeling notations, they should inform the most appropriate group of their requirements, and request that they be supported in the generic modeling notation.

F.8. Interchange Standards Session

Participants: Mike Imber (scribe and presenter), Roger Burkhart, Fred Hathorn, Mike Richardson, Brian Stucke, Craig Thompson

This session discussed architectural approaches to interfacing and import/export and also classified some existing interchange efforts in terms of generality and domain-specificity. The section below lists the interchange formats identified and the classification given to them.

ASN.1 [Abstract Syntax Notation.1 - ISO 8824]

- This is generic; in fact it is a 'meta-syntax', that is a syntax for defining syntaxes - it is part of the OSI family of standards. Any language defined using ASN.1 and its companion encoding BER.1 [ISO 8825] can be transported by OSI.

CDIF [CASE Data Interchange Format]

- This has both domain-specific and generic aspects.
- The transfer format, architecture and modeling notation are generic and can be used to model information in almost any domain.
- The CDIF Technical Committee have started discussions with the EXPRESS community to explore the possibility of bringing the two notations closer together over time.
- The standard information definitions developed are domain-specific, relating to CASE.

EXPRESS

- This has both a domain-specific and generic aspect.
- The EXPRESS language, and the corresponding Physical File Format are a generic modeling notation and interchange mechanism for that notation.

- Its primary use is by the PDES/STEP community for modeling product data, although its use is now spreading beyond that community.
- It is an ISO standard, and is currently under review.

EDIF [Electronic Design Interchange Format]

- This is a domain-specific effort, with the syntax and the modeling combined into a single definition, although they are moving towards the use of EXPRESS as a modeling notation.

IDEF Import/Export

- Work has recently started within the IDEF Users Group to define an Import/Export format for IDEF0 and IDEF1X. A domain-specific proposal called IDL (IDEF Definition Language) has been defined, but the group are apparently considering either ANSI IRDS Import/Export or CDIF for a longer-term solution.

XDR [Sun]

- This is generic, and provides a syntax, but no separate modeling facilities. It does not support pointers.

NDR [??]

- This is generic and provides a syntax, but no separate modeling facilities.

Import/Export Architecture

The concept of an Import/Export architecture was then discussed. It is important to understand the difference in approach between a domain-specific Import/Export facility, where the semantics of the information in the interchange has been standardized so that common definitions can be used between disparate tools, and the generic case of repository import/export, where there is no such agreed semantics.

In the former case, the exporter must implement a mapping function to convert the semantics from the representation understood internally by the exporting tool or repository to the form defined in the standard for the domain in question, and the importer must provide a similar mapping function to map the information from the standard form to that understood internally. The domain specific Import/Export approach is illustrated in Figure 3.

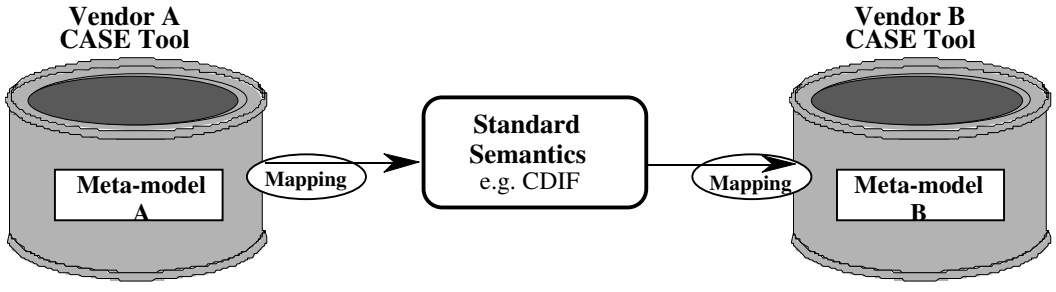


Figure 3 - Domain-Specific Import/Export

The approach taken is that an exporter should export all information that it can map to the standard semantics, and even use the extensibility features where required, and the importer should map all information that it can understand. The level of functionality of the importer will be driven by market forces; direct mappings and simple transformations are easy to implement - more complex mappings may wait until the market demands them once the technology is proven.

Even with sophisticated mapping functions, it must be understood that even the use of standards to define the semantics will never guarantee that 100% of the information will be transferred, or that a 'round trip' can be carried out without information loss due to differences in semantic richness between the two tools, but the approach offers the most effective way of removing the necessity for 'pairwise' development of custom interfaces.

In the generic import/export case, where there is no standard semantic definitions to provide a neutral set of definitions for the information being transferred, then the definitions must be provided by the exporter, and the importer must decide how to treat them; it may import the definitions and all the information, or it may have enough information to map it to internal definitions, but this is a completely different problem from that outlined above where a standard set of definitions are used. The generic Import/Export approach is illustrated in Figure 4.

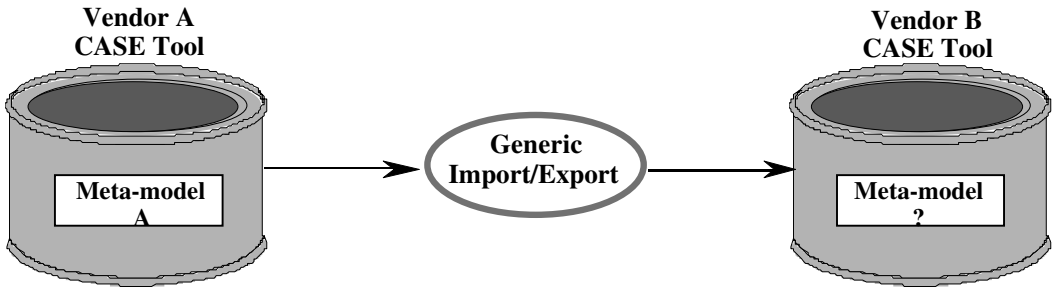


Figure 4 - Generic Import/Export

Another important aspect of interchange architecture to increase the reuse of concepts and standards is the separation of the definition of the syntax of the interchange language from the definitions of the information that is being transferred. To achieve this an appropriate architecture is required, similar to that developed by the CDIF Technical Committee, where both the syntax and the information content are defined in terms of a 'meta-metamodel' without reference to each other. This is illustrated in Figure 5. This approach allows the modeling notation adopted to be used to model information an any domain, and the interchange format to be used without change.

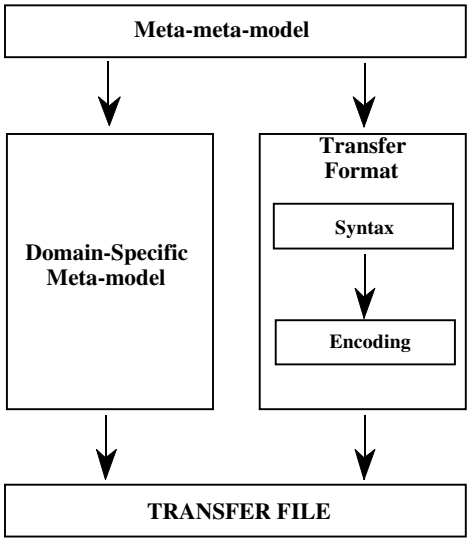


Figure 5 - Interchange Architecture

Within the definition of the transfer format itself, there may also be a split between the definition of the syntax from that of the encoding. The syntax defines the grammar of the interchange language down to the terminal tokens, and the encoding defines how these terminals are represented in the physical file. For example, different encodings could be 'Clear-text' intended to be human-readable, 'Character' intended to be easily transmittable over communication lines and 'Binary' which is optimized for use within a single processor, but is restricted due to the differences in integer representations across machines and other such aspects.

It was agreed that the objective of groups working in the area of interchange format definition should be to separate the requirements for domain-specific information modeling from the generic requirements of an interchange language. They should examine the existing generic facilities provided by standards such as CDIF and EXPRESS to see whether they provide sufficient modeling capabilities, and, if so, adopt the modeling notation as the means for expressing the domain specific semantics. If the modeling facilities are deemed inadequate, they should be encouraged to indicate the deficiencies to the groups concerned, in order to improve the capabilities for the whole community, rather than to invent a new interface language themselves.

F.9. Data and Control Integration Birds-of-a-Feather

Participants: Edie Bailey (presenter and scribe), Elizabeth Fong, Maggie Law, Bill Harrison, Bob Hodges, Frank Manola.

The topic was, Can you have control integration without data integration?

We began by asking, What does "data integration" mean? The following definition emerged:

> **Two agents are data integrated to the extent that they share a common understanding of the meaning of the data they both access.**

Key aspects of this definition are that:

- Data integration is defined with respect to the ability of two agents (re. tools or applications) to share data.
- Both agents must be able to act upon the data item based upon a common understanding of the meaning (i.e. understanding purpose, composition, contents, and/or format) of the data item.
- The extent of data integration is important in that agents that can thoroughly understand and modify the contents of a single data item are better integrated with respect to data than agents that can merely take action based upon a data item's schema or known data type.

We then asked, what does "control integration" mean?

The following definitions emerged, but there wasn't consensus on the wording:

Version I: **Two agents are control integrated to the extent that one agent's actions can invoke or trigger actions in the other agent.**

Version II: **Two agents are control integrated to the extent that they share a common understanding of the messages (formal interactions that can invoke or trigger actions) they exchange.**

Key aspects of this definition are that:

- The control integration definition should somehow mirror the data integration definition (two agents, common understanding).
- It must encompass both the Request and Notification aspects of interaction between agents.
- Some disagreement remained as to whether we are talking about control integration when informal events (e.g., a program aborts, a file is deleted, etc.) trigger some action by an agent. It is considered control integration when a formal Notification of an event (e.g., a message) triggers an action by some agent listening for that Notification.

Given the definitions of data and control integration, we ended by asking, Can you have control integration without data integration? and vice versa? It was generally agreed that you CAN have control integration without data integration. Examples included Requests and Notifications whose operations do not require data (e.g., Stopping an agent, etc.), or where only opaque handles to the data are passed between agents. However, the better integrated agents are with respect to data, the better they are able to communicate at a control integration level.

F.10. Object Model Soup Birds-of-a-Feather Session

Participants: Craig Thompson (presenter), Bill Harrison (scribe), Frank Manola, K.C. Morris, Roger Burkhart.

Many SDO/Cs are adopting object models or extending existing data models to support object models. ANSI X3H7 was chartered in part to help this process. It is doing two things:

— providing a matrix of object model features that may lead to better, perhaps even standard definitions of object model features like inheritance. This may help in converging new groups toward using common object models and may help existing groups to better understand migration paths for extending an existing object model toward the same coverage of another object model (e.g., relations SQL3 OM → or OMG IDL → C++ or C++ + EXPRESS-like constraints → better C++ for PDES/STEP). This approach may lead us toward a better way to compose OMs

from components (e.g., following the OMG notion that object model X's profile = Core + Set of Components).

— determining a taxonomy of approaches to handle situations where objects represented in different object models can interoperate. This latter task recognizes that there are already large installed bases for different object models and, unless they can be made to converge quickly (unlikely), there will be many programming-in-the-large situations where applications will need to cross object model boundaries.

This birds-of-a-feather session hoped to identify some scenarios where multiple OMs cause problems and some solution approaches.

We began with an object lifecycle scenario. In a conceptual object's lifecycle, it might be designed using an OA&D specification tool, created as an EXPRESS object, mapped to C++ so behavior can be specified via an PDES/STEP EXPRESS SDAI-C++ transformation, stored in a relational DBMS which uses the proposed SQL3 data model, moved to a remote location using OMG CORBA IDL, operated on from an application written in SmallTalk, displayed via the C++-based Interviews GUI toolkit; in short, the object may be moved through many different object services and represented in many different object models. While some of these mappings may be automated, many will be manual transformations requiring people to write code to do the transformations, understand versions of the object represented in many OMs, etc.

What can we do to make data and data model transfer between systems and services more seamless? Can we characterize the problem? Can we insulate object services from particular choice of object models?

We agreed that these are all data transformation problems and can be dealt with (though not always seamlessly) by known technologies. Problems occur at least three levels: conceptual schema mismatch, operational mismatch, and physical schema mismatch. Of these, perhaps it will be easiest to automate mappings among physical schemas.

Some approaches to handle the data model mapping problem are

- pairwise transformations (n**2 problem)
- clearinghouse representations (2n problem).

But, in the latter case, what clearinghouse should we use: SQL3, C++, OMG IDL, CDIF, PCTE, EXPRESS, etc. One "danger" is to assume that, to be programming language neutral, a group should invent "yet another object model" (YAOM). It is becoming clear to the community that this approach typically results in invention of a new object model that no community current uses and, in effect, invents a new language with both an object model (OML/DML) component and often even an object manipulation component. For instance, SQL3 has both a new object model and a new computationally complete object manipulation language (ODL/DDL).

Automated transformation may make the resulting model too awkward to program against. The problem of writing methods when the models don't match is worsened when importing software presuming a simpler model.

One promising compositional approach is to insulate the data model from the service provided, i.e., Service[Data Model]. This compositional approach can be used to decouple a service like change management or queries from a particular data model, at least at the specification level. An example would be to view the ATIS specification as Change Management[AtisDataModel] or to view SQL3 as SQL[theProposedSql3ObjectModel]. If this can be accomplished at the specification level, it can make it possible to define SQL[X] where X is the OM of ATIS, C++, IDL, EXPRESS, etc. This could be applied similarly for other services. This opens the door to a very seamless transition between services, for instance, allowing strong typing to be retained at the service control boundaries in the scenario we started

with. It seemed likely to some of us that most services are insensitive to the particular semantics of a given object model and CAN be decoupled. Demonstrating this in industrial strength implementations remains a challenge.

We noted that ATIS is a bundle of (object model, dispatch model, CM model) and might be written as ATIS = SUM(…) where SUM is some composition primitive. Using this idea, Change Management = SUM(Versioning + Configurations + Dependencies). We then discussed whether and how CM[X] = SUM(V[X] + C[X] + D[X]) can be specified/used either independently of X and or with different OMs bound to X.

We noted that if SQL[3] rather than SQL3, the compositional approach might help us specify simpler standards faster. This opens the questions of what constraints are applied between SQL and X in SQL[X]. It is at least clear that SQL1 = SQL[RelationalModel]. If SQL3 = SQL[new OM] then the question arises which object model to choose (e.g., C++, invent one, ...) and whether SQL can be made "particular object model independent." If this could be demonstrated, it would no longer be necessary to cross an object model boundary whenever accessing a database, which will be a drawback of SQL3. There is experimental evidence that the particular choice of object model CAN be decoupled from the SELECT-FROM-WHERE specification for querying sets. This same argument can be made for other services tightly coupled to particular object models.

This may relate to the X3H6 issue in which a tool (a.k.a. service) has both a messages interchange interface channel (to the other tools (services) and a data interchange channel (to repositories storing the state). The two channels have different characteristics (think of the messages channel as the client/server method call and the data channel as the method-access-to-instance-variable channel), but if repository adapters can exist (provide one repository's DML against another's DDL), it should be more possible to reuse service architectures against different repository architectures.

Figure 6 is not meant to imply that all state in a repository, or data store, is uniformly accessible to any service at any time. That is, we need to avoid letting services in through the "back door" to change data other services encapsulate. For example, if an indexing service depends on the state of some data items, updating the items directly will have side effects on the index it is referenced by.

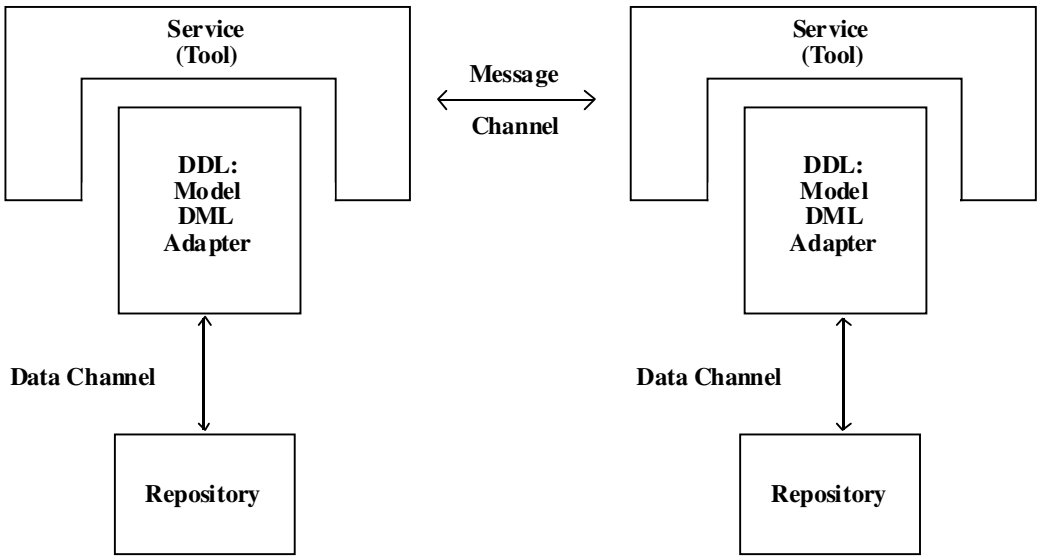


Figure 6 - Message Interface and Data Interface Channels

Toward the end of the session, we discussed other interesting service/OM compositional examples:

- It was pointed out that in CLOS generic functions may be versioned, methods may be versioned, and classes may be versioned. This raises the question of applying CM to meta data as well as data.
- There was a question whether EXPRESS is one of these meta data model languages?

  A reply was that it would be good to inherit the versioning mix-ins into an EXPRESS package. It was pointed out that EXPRESS has no execution semantics, at least not yet. Some people want to add methods to EXPRESS.

In summary, this session suggested that object services may be composable, that they may be parameterized by object model and that services may be "particular model independent." If these properties can be much better understood, it may mean that software systems will be simpler to develop and standards for them will be less monolithic and more compositional.

F.11. Standards Roadmap Session I

Participants: Craig Thompson (presenter), Maggie Law (scribe), Bruce Murrill.

One major product of this workshop and its follow-up activities should be a Roadmap for Integration of Information Technology Standards. Such a roadmap could become a significant tool for improving the interoperability of standards and standards-based implementations. This integration of information technology (IT) standards activities is a necessary step toward enabling enterprise integration.

Purpose of a Standards Roadmap

For this roadmap, the definition of "standard" should include approved specifications both from accredited standards activities, national and international, and from consortia developing industry-driven standards. As individuals and standards groups participate in developing the roadmap, the potential alignments of standards activities and products to achieve integration can be clarified. The types of roles that various standards activities play in achieving IT standards integration can then be addressed in the roadmap.

Another major benefit of development of a standards development roadmap would be to encourage discussion among participants leading to agreements between different standards groups and consortia. Such agreements between standards groups might include:

- Agreement to establish liaison among related standards activities.
- Agreement by one group to use the products of other standards group(s).
- Agreement to use terminology consistently across a number of standards groups.
- Agreement to a common data/object model, perhaps with model extensions as required for different types of standards.
- Agreement to use a specification document developed by an industry consortium as a base document for standardization in an accredited standards committee.

The roadmap can be developed in iterative cycles, with input and feedback from standards groups and other industry participants. As the roadmap is developed, it should be distributed to IT standards developers, implementers, and users, who can use it for standards comparison across industry. With the roadmap as a basis for standards activity coordination and standards integration, standards groups will share a common reference point to facilitate the agreements needed to carry out the integration tasks.

Examples of roadmap development tasks could include identifying and defining the following:

- Object Services that can be used in comparing the functionality of different standards, and as a guide in coordinating future versions of standards.
- Object Model features that can be used in comparing different object models from different standards, and as a guide in integrating future versions of the standards.
- Approaches for assessing the degree to which current standards support the Object Services and the Object Model features.
- Procedures for conformance testing to ensure that future standards provide particular Object Services or Object Model features that reflect the definitions in the roadmap for IT Standards Integration.
- Procedures for interoperability testing to ensure that future standards (and their implementations) provide Object Services or Object Model features that can be used by other standards (and their implementations).

A Framework for Classifying Technical Content of Standards

We identified two immediate tasks that needed be started at the workshop to initiate the development of a standards roadmap:

- develop a "big picture" view that fits as many as possible SDO/C efforts into a common framework.
- develop a data collection template that could be used to refine this global view by allowing collection of detailed information on different SDO/C efforts, their mission, scope, schedule, status, plans, and liaisons.

We began with a prototype for the first task. Our first assumption was that we needed some sort of classification scheme for grouping different SDO/C activities. We determined that any chart should list technical areas being standardized and separately SDO/Cs working in that area. This would allow capturing many-to-many relationships to help us detect possible overlap between efforts.

We decided to first develop a category scheme just for the technical areas, later to insert SDO/Cs working in those areas. We hypothesized that a framework architecture (hereafter referred to as an Object Services Architecture, or OSA) might provide a good picture allowing comparison of different group's activities and their relatedness.[3]

Figure 7 shows a generic OSA. Some sort of backplane connects different object services. In OMG, the CORBA backplane provides the composition of distribution, dispatch, and object modeling. Other SDO/C groups like X3H6 and Case Communique take dispatch or message passing as the primitive and might make distribution a separate service.

[3] We assume we can use the term "object" here since, for our purposes, object models can be viewed as subsuming relational, ER and some other data models. But think "data model" if you view object model as too restrictive or otherwise inappropriate.

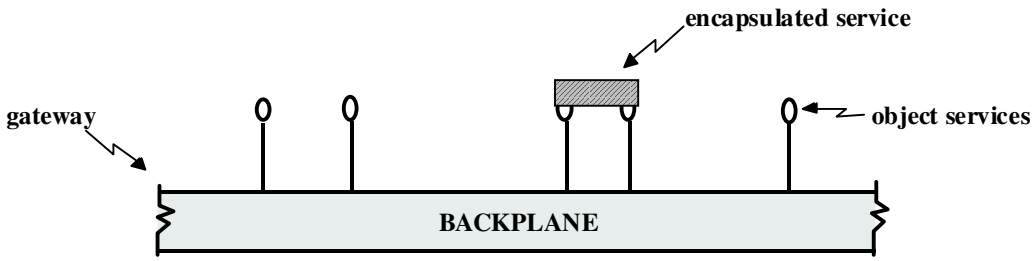


Figure 7 - A Generic Framework Architecture

OSAs provide a good candidate for understanding different groups' work since they can be viewed as essentially providing a list of services or functions that some SDO/C standard provides. Thus we can view Network Management Forum as providing a collection of network and communication services. Some groups hide several services in one service. Thus OMG CORBA can be viewed as hiding several more primitive NMF-like services. Some SDO/C standards incorporate several services into one standard monolithically. These monolithic standards provide the benefit of an integrated collection of services but often do not provide a rapid migration path for improving a composed standard when individual component standards improve.

Some broad categories of services are shown in Figure 8 (from right to left). These might include:

- communication and network services
- distribution services
- data management services (e.g., name services, security, link service)
- query service
- change management services
- generic interchange formats service
- presentation services (e.g., GUI, UIMS)
- common facilities (e.g., help, editors, licensing, and trading services)

Figure 8 - Services Architecture Hypothesis

For any given domain (e.g., Corporate Information Systems, CASE/software engineering, electrical or mechanical Computer Aided Design, Computer Integrated Manufacturing, Automotive, Petrochemical), there are typically domain-generic services (sometimes called tools) that describe common services (e.g., chip floorplanning) that may be widely useful across an industry but are specialized to a domain.

Any OSA requires some data or object model(s) for representing or passing data. Figure 8 contains an orthogonal row above the OSA row. It shows that different SDO/Cs rely on different data/object models to represent data. Models vary from flat (e.g., relations) to structural (e.g., IDL, various ER models) to OO models (e.g., SQL3, EXPRESS, C++, SmallTalk, CLOS). This list is far from exhaustive.

We can distinguish various meta modeling approaches that different groups are focused on. Also, generic models like those mentioned in the previous paragraph might be used in many domains. But many domain-focused SDO/Cs are developing "content modules" that standardize the content and behavior (entities, properties, operations) of domain-specific classes (e.g., signals, tires, tolerances).

Given the OSA/OM technical roadmap, we can at last map SDO/Cs to object services and object models. This is shown in the top three rows of Figure 8. We distinguished three kinds of SDO/C in our simple category scheme: those developing standards, those developing profiles of standards that will interoperate, and those representing user communities. Some organizations are doing all three. Overall, Figure 8 shows a first approximation of how SDO/Cs relate to services and object models.

Our conclusions from the above exercise are:

- an OSA-based classification scheme can be used at least to make sense of the SDO/C landscape and can be used to detect potential overlap between groups. However, as a diagram, it does not capture the temporal dimension of when certain standards will mature. Also, at a global level, the map has poor resolution and more fine-grained analysis will be needed to insure we can use the same device at a finer resolution to focus on, say, how ODMG (a consortium of OODBs) and X3H2 SQL relate to each other and how both relate to OMG's OSA architecture.[4] Similarly, an IRDS repository might also be viewed as depending on a composition of mandatory services, like name service, query service, change management service, etc.[5]
- we speculate that an object services architecture provides a strong candidate for an enterprise integration framework if it can really "cover" the existing SDO/C activity landscape. Further work will, of course, be needed to understand that it is THE right way to go since there are many unanswered questions (will OSA-based schemes scale, handle federation, have good performance).

F.12. Standards Roadmap Session II

Participants: Maggie Law (presenter), Bob Hodges (scribe), Barbara Cuthill, Bill Harrison, and Larry Johnson.

This session continued work on the standards roadmap by outlining the content of a data collection template that would help with collecting usable information on a wide range of standards development organizations and consortia. The group began defining the template with a goal of generating a "fill-in-the-blank" form that would take only a few minutes to fill in and still provide enough characterizing information to support analysis. This format would allow explanations in free-form text, but would also

---

[4] Contact Craig Thompson if you are interested in this analysis.

[5] Contact Bob Hodges if you are interested in this analysis.

provide a baseline of information that would not require parsing the information from the text. The approach included:

- defining or borrowing a classification structures for 1) the areas of Information Technology (IT) covered and 2) the vertical integration domains;
- mapping groups to the standard classification schema;
- capturing group relationships and dependencies.

Once the information was collected, analysis could include the following tasks:

- build a dependency network of groups and their deliverables;
- map group deliverables/standards against the classification structures;
- map group deliverables/standards against a timeline;
- identify group deliverable dependency conflicts;
- identify overlaps and gaps in terms of IT areas;
- identify overlaps and gaps in terms of subject areas or integration domains.

The following topics were the result of a brainstorming session to generate a "wish list" of the information about SDO/Cs that would feed the analysis tasks and lead toward the realization of the roadmap:

Types of IT standards:

- ANSI accredited IT standards, (i.e., X3, IEEE, EIA)
- International IT standards, ISO groups
- Association and society IT standards (e.g., ECMA)
- Industry-driven IT standards from consortia (e.g., OMG, X/Open)
- Government led IT initiatives (e.g., STARS, DSSA)
- Domain-specific IT standards (e.g., PDES, STEP)

Role Definition:

- Who you are? How can you be contacted?
- What role does your group play in IT standardization? In what way is it unique?
- Are the standards you produce accredited or industry-driven? To what degree does the IT industry now support/implement your standards?
- What role do the standards that your group provides play in support of an enterprise?
- With what other groups would it be helpful for your group to interact or cooperate? For what benefits?
- Who is your customer base?
- From which other IT standardization efforts do you receive support, information, or products (i.e., standards)?
- To which other IT standards efforts do you provide support, information, or products?
- Across the enterprise, who uses the IT standards that you produce?
- What is your mode of operation?
- How does your group now operate to produce IT standards?

- What would be useful in improving your group's standards in terms of:
  - integration and interoperability with other IT standards,
  - speed of developing standards,
  - quality of standards specifications and products based on standards,
  - user's selection of services, and ease of use of products based on standards, and
  - ease of maintenance of standards and products based on standards.

Types of Standards Products

- Base standards profiles
- Implementation profiles
- Conformance profiles or testing
- Interoperability testing
- Technology transition testing

F.13. Standards Roadmap Session III

The final Roadmap Session continued developing the template format and preparing for collection of information from the groups in attendance at the workshop. The template that resulted from this session is reproduced below.

The following information was developed to help clarify items included on the template. As another means of refining the template we asked for volunteers from the workshop participants to complete the template. The completed templates for twelve of the groups who attended the workshop are included in Appendix E.

Dependencies Liaison:

Relationships indicate the kind of dependency on the other group described. Examples of this include: Using a Deliverable to be supplied by another group, influencing the decisions made by another group, supporting the work of another group, continuing cross-communication from another group, tracking the decisions of another group for reuse.

Areas of Technology:

The focus areas are the areas on which the group is focusing its efforts in creating its output. Dependency areas are other areas from which the technology is expected to be used or whose technology direction is of interest in defining the output.

Data Models:	strictly this is intended to indicate an interest in meta-models and meta-meta-models rather than in models of specific domains. Data models provide support for the description of data to be shared among tools.
Message Models:	provide support for the description of messages interchanged among tools.

Object Models:	Object models combine data and message models in a style that associates data and message declaration with types or classes of objects and generally describe how messages sent for an object are delivered to implementations called methods and how the methods access their associated data.
Tool Models (Application Integration Architectures):	define the way in which tools (or applications) are described, how they interchange messages, and how they access shared information.
Process Models:	define the way in which the development steps are and sequences of development steps are defined, tracked, and coordinated.
Interchange Formats:	define concrete representations of data to be interchanged.
Repositories:	define a functional interface for accessing shared information, generally associated with data models.
Methodology:	defines the way in which the development steps are and sequences of development steps are intended to lead to product results.
Requirements:	are that part methodology concerned with eliciting the customers' needs that must be provided for in a product.
Metrics:	are that part of methodology concerned with measures that can be taken during the process of product development to assure quality and satisfaction of requirements.

Group Information Template

Group Name (Full/Acronym):

Contact **Name:** _____

Address: _____

Phone/FAX: _____

e-mail: _____

Mission:

Characteristics:

[ ] Standards Development

[ ] Accredited Standards Organization

[ ] Government Organization

[ ] Other:_____

[ ] Vendor Consortium

[ ] User Consortium

[ ] Non-government Organization

Integration Domain:

	Current Focus	**Areas of Applicability**
CAD/CAM	[ ]	[ ]
CASE	[ ]	[ ]
Enterprise Integration	[ ]	[ ]
Office Automation	[ ]	[ ]
Management	[ ]	[ ]
Other:_____	[ ]	[ ]
Other:_____	[ ]	[ ]

Group Deliverables:

Title	Description (e.g., Standard, Spec., Test Suite Implementation, etc.)	Target Date
1.		
2.		
3.		
4.		
5.		

Dependencies Liaison:

Group	Product/Technology	Relationship (e.g., Using Deliverable, Influencing, Supporting, Cross-Communicating)
1.		
2.		
3.		
4.		
5.		
6.		
7.		
8.		
9.		

Areas of Technology

	Focus Area	**Dependency**
Data Models	[ ]	[ ]
Message Models	[ ]	[ ]
Object Models	[ ]	[ ]
Tool Models (Application Integration Architectures)	[ ]	[ ]
Process Models	[ ]	[ ]
Interchange Formats	[ ]	[ ]
Repositories	[ ]	[ ]
Database Management	[ ]	[ ]
Operating Systems	[ ]	[ ]
Distribution	[ ]	[ ]
Name Spaces	[ ]	[ ]
Programming Language	[ ]	[ ]
User Interfaces	[ ]	[ ]
Graphical User Interfaces	[ ]	[ ]
Methodology	[ ]	[ ]
Requirements	[ ]	[ ]
Metrics	[ ]	[ ]

	Yes	No
Are you basing work on Object-Oriented Technology?	[ ]	[ ]

Which object model do you use: _____

	Yes	No
Does your work apply in the absence of Object-Oriented Technology?	[ ]	[ ]

F.14. Management Session I

Development of Vision, Mission, and Goals

The four management breakout sessions focused on how to leverage ongoing standards efforts and maximize the effectiveness of resources used to develop and use standards. The primarily focus of these sessions was on the management of standards activities. Each session had a distinct area of concern which the group addressed. The sessions were organized so that the end product would be a strategy to help coordinate and guide the development of "standards." The topics of these sessions were as follows:

1. Develop Statements of Vision, Mission and Goals
2. Identify Relationships Critical to Standards Development
3. Develop a Plan to Support Standards Activities
4. Decide Possible Directions for Future Coordination

The first session focused on trying to define the mission, goals, and objectives which the group could use as guidelines for developing strategies to improve the standards development process.[6]

Several assumptions were made while developing the goals. The management group was acutely aware that any effort to improve the standards development process would have to work within the current structure and operating regulations of the Standards Development Organizations (SDOs). The standards development and approval processes operate according to applicable legal and legislative restrictions. The standards organizations have resources such as legal staffs which are cognizant of considerations like anti-trust laws and indemnification requirements. Another assumption was that neither a consortium nor any particular agency can have special influence over any standard organization to radically change their standards development process, so any suggested improvements would have to be leveraged through formal and informal liaisons to standards groups.

The vision, goals, mission, and objective statements are described below. We discussed how to distinguish goals from objectives (e.g., objectives cannot be easily defined in quantifiable measures), lacked time to complete this discussion, and did not reach consensus on the list below but did complete the following strawman, which will be open to refinement in future meetings or work groups.

VISION — To enable Enterprise Integration

MISSION — Maximize the "plug & play" of the technical infrastructure products based on standards.

GOALS —

- Provide coordinated standards, reducing redundancy, coordinating terminology.
- Minimize the time and cost of producing standards.
- Minimize the time to market and cost of standards based products.
- Minimize the time and cost of integration of standards based products.

---

[6] During the first session, we decided to spin off a parallel activity to focus on building a roadmap for standards activities (see F.11. Standards Roadmap Session I)

OBJECTIVES/DELIVERABLES —

- Guidance for industry and users.
- Roadmaps for Standards coordination and evolution.
- A common glossary for communication between standards efforts.
- Models of standard harmonization
- Commitments among groups for coordination
- Coordinated schedule of standards activities
- Metrics for integration (e.g. Software Engineering Institute)

The group felt there were several ways to improve the standards development process (e.g., encourage earlier and frequent reviews of standards-in-progress, seek to include academic participation in the process, collect requirements from users via surveys or user meetings). Some suggestions do not involve legal changes but rather cultural changes that might be put in place in the form of guidelines.

A main area for general improvement is the current standards coordination and development process. Some changes are just in adopting newer technology for coordinating efforts more efficiently, for instance, via the use of e-mail for document preparation and review and the use of teleconferencing when full face-to-face technical reviews are not practical. This may help smaller organization compete, making standards development less dependent on company size or wealth

Another area for improvement is the use of interim standards. Many standards organizations do not publish any interim standard, and draft copies of a proposed standard are only released to members of the standard's working group. The standards effort does not receive outside public review until it is ready for balloting. If interim standards were released for review, some potential problem areas in a candidate standard could be resolved before reaching the critical balloting stage. This suggestion does not seem to imply a change of rules but rather a change of culture in which a group actively seeks community consensus for its ideas via a continuous review process.

Another area for improvement in the standards process is to better coordinate critical players in the process, some of whom may not be involved in SDOs. Obstacles to harmonization (working together to cause convergence and interoperation of standards) were identified as follows:

- Funding (time and travel costs) to participate in multiple overlapping activities to form cross-group consensus.
- Proprietary ownership of specifications, e.g., copyright, which can prevent a group from putting its specification in the public domain. Reasons may be to gain or hold competitive advantage or to retain proprietary control an emerging standard.
- Speed of consortia or closed groups versus slowness of the open, public standards process. Temporal mismatch between closed and open groups.
- Turf battles and NIH; terminology or scope mismatches; also lack of awareness of competing, overlapping, related standards, possibly because work-in-progress is not made widely available; also, lack of understanding of standardization procedures.
- Differing priorities.

It was decided that the Management II breakout session would focus on how to better coordinate critical players in the standardization process.

F.15. Management Session II

Identifying Relationships Critical to Standards Developments

This session focused on identifying the major players within the standards development process. The question we asked was, how can industry and government maximize their investment in standards efforts. Companies are "downsizing" or "rightsizing" their organizations and scaling back activities which are not in direct support of returning company profits. Standards work is a volunteer-driven process and requires heavy investment from companies. A company must not only supply the manpower and salary, but also travel cost and company resources and equipment. A real improvement is needed in the standards process for companies to more directly understand how to leverage their investment in standards.

One problem is, the standards landscape is complicated. Many groups exist and work on (sometimes) overlapping projects. There is no real guarantee that standards that are produced will be able to interoperate. An immediate benefit of this workshop is the initial development of a roadmap of ongoing standards efforts. See Roadmap Breakout Sessions. This high level view begins to provide industry and government with guidance on which standards are addressing technical areas vital to their futures. Another benefit of the roadmap is the ability to determine where possible redundancy or duplication exist in standards efforts. The roadmap will help pinpoint where technology gaps are within the current standards effort. In either of these situations liaisons, formal or informal, must be established to help correct these deficiencies and increase the coordination among the standards developing organizations.

Several existing mechanisms were identified that allow SDOs, consortia, and other groups to be coordinated. These include:

- Groups can send information about their principal work items to related groups in order to "advertise" their efforts.
- Consortia members can become members of SDOs.
- Individuals in one SDO can act as a coordinating liaison to another SDO.
- SDOs and consortia can choose to meet at the same place and time and overlap their meetings.
- Memoranda of Agreement (MOA) or informal agreement.
- One group can "reference" another standard instead of reinventing it.

A major opportunity for improvement in the standards development process involves leveraging the results of technology consortia in accelerating standards development. Many consortia make large investments in computer technology. One benefit consortia could provide to the standards organizations is a source of candidate interim standards. The technology base developed by consortia is derived from funded projects, that is, they are based in experience. The consortia could also leverage their relationships with other consortia to help coordinate with other groups before standards reach the SDOs. The liaisons the consortia can establish between each other are not as restrictive as the relationships the standards groups can establish.

Another possibility is that consortia can seek funding to implement standards test suites. This facilitates the transition of new technology into industry more quickly. Models for standards validation (field testing) within industry/government exist for example at NIST.

The following model for coordinating behavior between the consortia and standards organizations was developed.

1. SDOs serve a purpose; they provide an open forum for developing a consensus standard (national and international).
2. Industry consortia serve a purpose; they develop a mini consensus more rapidly.
3. SDOs should take advantage of the products of the consortia, whenever possible, as they should reflect a generally agreed to approach to satisfying customer needs.[7]

Attendees recommended that:

1. The industry consortia should consider the standards that they develop to be "interim" industry standards, where the final standard would be reflected in the product of the SDOs. The "interim" industry standard is an idea available in IEEE and EIA standards; EIA and IEEE are both ANSI accredited SDOs.
2. The "interim" industry standard should be submitted to the appropriate SDO(s) as early as possible (e.g., even before it becomes an "interim" standard) as a proposed base document or as a change proposal to evolving work to begin the process of building open consensus (while getting feedback from the SDOs).
3. The consortia should represent the advocacy position for the contribution by participating in the SDO process as a member of the SDO, and should, in good faith, consider comments on the contribution as contributions themselves. Not invented here (NIH) must be absent from the perspective of both groups if progress is to be made, and neither group should expect that they have **THE** answer.

F.16. Management Session III

Developing a Plan to Support Standards Activities

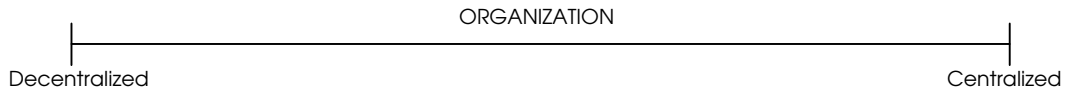
The third management session focused on (1) identifying ways to enable further work to coordinate standards and (2) discussing to what degree this coordination should be formalized. There was consensus that, if nothing else occurred within the bounds of this or future, similar workshops, these events are successful because they provide a forum for communication between organizations doing similar technology. Beyond this, members of the management session identified the following list of items which could help expedite and coordinate the standards processes.

- Enhanced working group arrangements(e.g. E-Mail, Telecons).
- Single source of up-to-date information on all aspects of standards group work plans, roadmaps, etc., of relevant groups.
- Development of a common vocabulary for related efforts.
- Integrated calendar of all standards working groups.
- Continued forums (e.g., future standards coordination workshops) for technical discussions and interaction.

---

[7] See Appendix J. Consortia Standards Process Model

The problem with these items is that they take committed resources to establish and maintain. The management group also looked at the full spectrum of control a standards coordination effort might accomplish, from a fully formalized central controlling body to an informal dispersed set of activities.



The characteristics of a fully centralized organization are:

- Establish as a legal entity
- Elected roles and responsibilities
- Membership requirements
- Establishment of working relationships through formal liaisons

The characteristics of a decentralized organization are:

- Little or no legal considerations
- Liaisons are informal or established ad hoc
- No membership requirement for participation
- May lack continuity

The idea of a formal organization with by-laws and a full charter was determined to be too aggressive at this point. On the other hand, for some of these resources to become available, they would have to funded and probably could not be maintained by a strictly volunteer effort. Much of the discussion during this session related to the concern that the group wanted to facilitate ongoing coordination but did not want to set up another formal structure.

The management group developed a recommendation to present to the general session to establish a core team of representatives to help guide the standard coordination effort forward. The management group decided to use Management Breakout IV to develop a set of recommendations to help market the coordination activities.

F.17. Management Session IV

Deciding Possible Directions for Future Coordination Efforts

This session focused on developing recommendations for continued SDO/Consortia coordination activities, including future workshops similar to this one. The following actions were recommended to disseminate the results of this meeting and to gain support for possible future coordination efforts:

- Publish a press release and executive summary of workshop results o Publish the workshop proceedings
- Distribute the SDO/Consortia Coordination Model
- Release results of the standards roadmap activity
- Decide plans for next steps (e.g., future meetings) and an approach for funding future efforts

In the plenary summary of this session, workshop participants agreed that the documentation listed above will provide a good starting point to help coordinate future SDO/Consortia efforts. The possibility of setting up a funded organization to supply all the coordination services discussed in Management Session III was judged to be too aggressive. Representatives from several organizations volunteered to provide a subset of these services as a starting point. The National Institute of Standards and Technology (NIST) and the Object Management Group volunteered to host the next workshop, probably in early December 1993. NIST also volunteered to host the Roadmap results on their bulletin board service.

G. Conclusions and Recommendations

The workshop concluded with a plenary session in which we attempted to capture "lessons learned." In addition, we requested written comments from attendees on the value and results of the workshop (see Appendix F). This section summarizes many of our conclusions and recommendations. There is the usual caveat: no workshop participant is likely to agree with all of the following.

Trends

Across most of the standards landscape, there are several trends that may affect the future and coupling of different SDO/C activities:

There is a trend toward extending existing standards to use **OBJECT MODELS**. The current tendency of SDO/C groups is to develop their own object model ("yet another object model"). There are arguments why some object models need features like delegation or asynchronous dispatch. There is confusion on where an object model ends and an architecture or object service begins. So far, there appears to be too little work on developing techniques for insuring interoperation of object models.

There is a trend toward **DISTRIBUTED SOLUTIONS**. Some groups are delivering just distributed services. This goes beyond client-server and peer-to-peer views of distribution to assuming these underpinnings and considering decentralized and cooperative higher level distribution services. We are still understanding how to distribute functionality in distributed systems (e.g., should they include asynchronous operations, concurrency control primitives, higher-level transactional RPCs, replication, security, be closely coupled to object models as in OMG IDL, etc.) and how higher level systems (e.g., change management, query, licensing, etc.) should depend on them.

There is a growing trend toward **SERVICE ARCHITECTURES**, like that of OMG. The Roadmap I breakout session provided an overview of all SDO/Cs attending the workshop and argued that many of these fit this framework. There is hope that we can learn to compose object services to configure systems like repositories, databases, CASE tools, etc. out of primitive services, but the composition primitives are not fully understood. At this time, many individual SDO/C activities do not view themselves this way. More work will be needed before industry can be sure this trend will become dominant.

There is a trend toward **OPEN SYSTEMS**. Frameworks and standards that are monolithic can lock customers to specific proprietary solutions. Customers need to migrate information between these. More and more, expensive custom federation and glue software is being written to bridge these gulfs. At the same time, some standards, like the X/Open XA and Remote Data Access target specific forms of federation to insure that transactions or queries can span different transaction monitors and relational databases. Proprietary frameworks are still the most sophisticated but are idiosyncratic; open frameworks are less developed but appear more generic.

There is a trend toward development of **CONTENT MODULES** (domain-specific, vertical industry models that yield sharable "common entities" for ECAD, MCAD, software, graphics, etc.). This trend may allow generic operations (queries, change management, etc.) to be useful across broad domains and can, at the same time, allow domain/industry-specific reuse of common services and tools. For a while, some domain-specific groups worked to develop generic frameworks; there seems to be a trend away from this now, so that domain-specific SDO/Cs depend on generic SDO/Cs for this support.

There is a trend, or at least a growing interest, in **ENTERPRISE INTEGRATION**. Groups are considering how to scale software architectures to real and virtual enterprises. One driver toward integrated enterprises is the shift toward the use of electronic commerce (e.g., EDI) for cross enterprise interactions. The CALS Initiative, which requires the ability to share information (e.g., a design) between subcontractors, is another example of this trend.

There is a trend toward **COMPONENTIZATION or SEGMENTATION** of at least some IT standards. SDO/C Groups appear more willing to reference others' work. This seems related to the trend toward open systems and services architectures.

There is a trend toward more **INDUSTRIAL CONSORTIA**. As the pace of technology quickens and the expense of developing durable industry-wide enterprise-wide solutions grows, companies are banding together. This means that there is an increased need for consortia that can often more quickly develop good "interim" solutions than an open SDO process might. But the openness is ultimately required to insure the broadest market. It also means, at least, that the consortia landscape is getting harder to understand and is increasingly seen as a battle field for achieving competitive advantage.

We did not discuss other relevant trends: the trend toward multimedia, the trend toward downsizing, the trend toward internationalization, the huge and escalating costs of maintaining legacy software, the trend toward dual-use, defense conversion, and economic competitiveness, etc.

More R&D is Needed

A careful reading of this workshop report indicates several areas where we lack the technical understanding needed to reach our goal of achieving the integrated enterprise. Some of these areas are listed below. Some groups are already working in these areas:

At present, we do not know enough about specification formalisms and fail to make enough use of declarative formalisms. We have not yet identified a very complete list of integration mechanisms. We do not even have very clear definitions of overworked terms like repository, framework, interoperation, open, migration, etc. We do not have a list of scenarios of what it will mean to have an integrated enterprise nor have we carefully decomposed such scenarios to technical and cultural subproblems.

We do not understand enough about "algebras of composition of megamodules". We have too few reference implementations of (object) services architectures. We have too little experience with mix-and-match, with scaling to enterprise solutions, and with loose federation, tight integration, and mediation as solutions to heterogeneity.

We do not yet have effective solution approaches to allow interoperation among data/object models. Though we are making progress in factoring both control integration (via Services Architectures) and data integration (via object models and content modules), we can not yet demonstrate durability of standards and reuse. We still do not understand the algebra of change, that is, of graceful migration from one system to another or to a later variant. For instance, we do not yet have a way to describe a migration path for PCTE and OMG to converge; we can not quite describe how a repository and a DBMS relate. We do not yet know if compositional standards will reflect compositional architectures.

We do not have a widely used pipeline to connect academic work to industry need and to connect industrial solutions to standards, along with social processes to promote cooperation and accelerate consensus leading to standards. Much academic progress never results in change to industrial solutions; standards vary in quality and are often ignored because industry fails to realize they exist. Reference implementations, provided as testbeds, may provide a way for academic groups to experiment with testbed

implementations to improve them. Too often we stand of the toes of past generations instead of the shoulders and fail to learn enough from earlier experiences.

We have few business models to accelerate the evolution of revolutionary increases in capability in IT Enterprise Integration.

What the Workshop Accomplished

We noted at the beginning of the workshop that the cost to participate in time and money was no more than a typical SDO/C meeting and that a simple metric of success of the workshop was whether participants found it worthwhile.

Based on the workshop evaluations, workshop participants DID value the workshop. By targeting knowledgeable, activist contributors to SDO/Cs, this workshop increased the bandwidth of communication across groups since all participants were well-informed in their respective areas. The workshop provided:

- an opportunity to build a shared vision of enterprise integration
- a top-down opportunity to paint "a big picture" of how different groups contribute to this goal
- a kind of SDO/C marketplace in which various sorts of SDO/Cs (accredited, consortia, government, national, international, producer, profiler, consumer) could meet
    - to better understand the overall objectives (vision and big picture)
    - to advertise their own SDO/C's contributions
    - to understand, review, and critique other groups' directions, and
    - to locate promising liaison partners.

The value of the five days was partly measured in the progress of participants in teaching and learning about their various activities. The value of the workshop report is to provide a vehicle for disseminating some of this information. But the lasting value of the workshop probably cannot be measured: it will be in what legacy of actions it causes to occur in the future that progress the vision of achieving the integrated enterprise.

Workshop Deliverables

During the workshop, we identified the following results we felt would be valuable outcomes of the workshop or of post-workshop efforts.

- **Landscape of Standards**. This is intended to provide an overview of how SDO/Cs related to each other and to the overall objective. This was started with Figure on page 4 and refined with Figure on page 32. Much refinement is needed before this roadmap is useful. Some of the following results provide the needed better resolution.
- **Roadmap of SDO/C Activities**. We completed an SDO/C Description Template (section F.13) and some groups began to fill in their status (Appendix E). We identified a need for an electronic bulletin board to post SDO/C calendar and status information and NIST is providing one.[8]
- **Matrix of Data/Object Model Features**. This effort intended to help us understand the range of needs different groups have in developing object models with different semantics. X3H7 is

[8] See Appendix I. Project Summary Repository

pursuing this problem and also the problem of how to deal with data integration issues arising from multiple existing object models.

- **Matrix of Services**. This is intended to provide a decomposition of control and management systems into a collection of subsystems that can be individually described and composed. The Service Description Template in Appendix H provides a strawman descriptor for individual object services analogous to the SDO/C Template for SDO/C groups. Several groups are describing their activities via Services Architectures. X3H4 is providing a comparative matrix.

In addition, we identified both technical and business/cultural/management roadblocks that are preventing us from realizing our vision as quickly as we would like. In some cases, we identified promising solution approaches, some of which are mentioned here.

- Object models may be decoupleable from object services. If so, we could solve a number of data and control integration problems caused by close coupling of different object models to specific services.
- If object services can be composed to form more complex information systems, then compositional standards can be developed that reflect this, simplifying the IT standards landscape.
- There is a need to continuously improve our process of standardization. SDOs and consortia need to talk more; groups that meet together for a day of overlap provide an excellent review process for each other. Groups need to think in terms of passing their requirements to each other rather than letting scope creep and overlap occur. Information on the status of some SDOs (annual reports) is not widely disseminated; no such information exists for some other groups. Similarly, standards themselves require too much effort to track, understand, review, evaluate. An annual SDO/C workshop will be helpful in directing our overall progress. The ACM Transactions on Standards can make this information more widely available.
- Specific funding for limited tasks may be useful to remove specific roadblocks. These include developing a roadmap for PCTE and OMG convergence, developing solution approaches to the interoperation of object models, developing reference implementations of some standards or demonstrations of scalability, converging specific efforts all working in a common area (like change management service, CASE messages, generic interchange formats, etc.), and developing standards migration strategies.

Our Challenge--Follow-up Action Plan

As mentioned earlier, workshop participants recognized that the workshop was not an end in itself. Litmus tests of the value of this forum will be seen in:

- whether technical and management solutions to problems identified in the workshop are resolved by cooperating SDO/Cs, e.g., see list of roadblocks above.
- whether participants or interested others actively contribute to follow-on activities (e.g., a Second Application Integration Architectures Workshop; making information on SDO/C activities more accessible, perhaps by gathering and sharing SDO/C descriptors by e-mail; more analysis of areas of overlap).
- whether the results of the workshop and its spirit of cooperation can be propagated beyond the group of workshop participants.
- whether individual SDO/Cs take actions to change or improve their technical direction to insure the vision of enterprise integration.

The challenges before this community are:

- to continue to build a shared vision of an integrated enterprise and to better understand whether and what form of compositional IT software architecture(s) can achieve this end.
- to provide effective ways to understand technical and business issues that need to be resolved to realize the vision of the integrated enterprise and cooperate to share solutions.

We discussed how to go about this. Two ends of the spectrum are the centralized approach and the decentralized. In the centralized approach, we might create some central authority responsible for standards convergence. Most attendees felt this just had the effect of creating yet another SDO/C and that there is no natural "top" in this picture. That is, no one central authority like the government, a standards group, or a consortia can naturally play this role. At the other extreme, we considered leaving all further action aimed at enterprise integration up to activist individuals or to SDO/C groups themselves.

Workshop attendees felt this is closer to the right way to proceed---a series of small steps rather than one giant step. In effect, enlightened self-interest and market forces continue to be the driver for groups to take the initiative to band together to solve enterprise integration problems.

Several individuals are interested in continuing the activities initiated at the workshop. Participants felt more can be done by e-mail by activist individuals willing to sponsor data collection and analysis activities (e.g., coordinating SDO/C calendars, work plans and schedules, sharing specifications).

The workshop organizers encouraged attendees to take the initiative to continue the work started at the workshop. There is still a need for someone to encourage different groups to complete the SDO/C Templates. Once the data is collected, there is the task of analyzing it and keeping it up to date.

There is a continuous need for groups to meet together to critique each others' work and to try to identify overlap in their activities and remove it. Participants of the workshop felt that there is a need for continuity in developing and improving our vision. They suggested that we should annually convene via a top-down workshop (or congress or symposium) which can continue to provide neutral ground for helping to steer all groups toward a common enterprise integration solution. A group of workshop participants volunteered to help organize a Second Application Integration Architectures Workshop in early December 1993.

Appendix A. Attendance Roster

Colin Ashford Bell-Northern Research MS: CAR242 P.O. Box 3511, Station C Ottawa, Ontario, Canada K1Y 4H7	+1 613 765 4929 Fax: +1 613 765 4920 ashford@bnr.ca	Network Management Forum, ISO/IEC/JTC1/SC21/WG4
Edie Bailey Hewlett-Packard Company 3404 E. Harmony Road MS #7 Fort Collins, CO 80525-9599	303-229-6160 bailey@fc.sde.hp.com	CASE Communique, (1992 chairperson)
Bob Balzer USC/ISI 4676 Admiralty Way Marina Del Ray, CA 90292	310-822-1511 Fax: 310-823-6714 balzer@isi.edu	DARPA Programs
David Beech Oracle Corporation 500 Oracle Parkway Redwood Shores, CA 94065	415-506-6420 Fax: 415-506-7203 dbeech@oracle.com	X3H2 SQL, Object Management Group
Jack Bissell UNIX International 20 Waterview Blvd. Parsippany, NJ 07054	201-263-8400 x233 bissell@ui.org	UNIX International -- Distributed Application Technology Project Manager
Alan Brown Software Engineering Inst. Carnegie Mellon University Pittsburgh, PA 15213	412-268-6194 Fax: 412-268-5967 awb@sei.cmu.edu	Navy Next Generation Com puting Resources (NGCR) Project Support Environment Standards Working Group (PSESWG)
Roger Burkhart Deere & Company John Deere Road Moline, IL 61265	309-765-4365 Fax: 309-765-5128 roger@ci.deere.com	X3H4 Information Resource Dictionary System
David Carney Software Engineer Inst. Pittsburgh, PA 15213	412-268-6525 djc@sei.cmu.edu	Navy Next Generation Com puting Resources (NGCR) Project Support Environment Standards Working Group (PSESWG)
Neil Christopher 6500 Chase Oaks Blvd. MS 8408 Plano, TX 75023	214-575-3360 neil@ease.dseg.ti.com	NCMS - Rapid Response Manufacturing, Architecture Committee (Chairperson)

Dr. Barbara B. Cuthill NIST, Div 872 Bldg. 225, B266 Gaithersburg, MD 20899	301-975-3273 bcuthill@swe.ncsl.nist.gov	NIST Software Engineering Group, North American PCTE Alliance
Elizabeth Fong NIST Technology Building, A266 Gaithersburg, MD 20899	301-975-3250 Fax: 301-590-0932 fong@ecf.ncsl.nist.gov	ISO/IEC/JTC1/SC21/WG3, X3H7 (International Representative)
Vic Goddard X/Open Co. Ltd Apex Plaza Forbury Road Reading, RG11AX England	+44 734 508311 v.goddard@xopen.co.uk	X/Open -- Distributed Systems (Development Manager)
William Harrison IBM T.J. Watson Research HI-B28 P.O. Box 704 Yorktown Heights, NY 10598	914-784-7631 Fax: 914-784-7455 harrisn@watson.ibm.com	X3H6 Case Tool Integration Models -- Ad-hoc Subcommittee on Infrastructure (Chairperson)
Frederick C. Hathorn Directorate of Defense Info. 1225 Jefferson Davis Hwy Suite 910 Arlington, VA 22202	703-746-7924 Fax: 703-746-7396/7 fhathorn@ddi.c3i.osd.mil	DoD Corporate Information Management Initiative -- I-CASE Technical Manager,
Bob Hodges P.O. Box 869305 ms 8482 Plano, TX 75023	214-575-3442 Fax: 214-575-5130 4937603@mcimail.com	X3H4 IRDS System Architecture and Integration Task Group (Chairperson)
Glenn Hollowell 2706 Montopolis Drive Austin, TX 68741-6499	512-356-7166 Fax: 512-357-3575 glenn_hollowell@sematech.org	SEMATECH, Sr. Computer Scientist, X3H7 Object Information Management, Vice Chair, Object Management Group, Database Special Study Group (X3/DBSSG)
Mike Imber LBMS Evelyn House, 62 Oxford St. London, WIN 9LF U.K.	+44 71 636 4213 100031.700@com puserve.com	CASE Data Interchange Format -- CDIF, (Chairperson) ISO/IEC/JTC1/SC7/WG11
Dr. Peter Janecek X/Open Co. Ltd Apex Plaza, Forbury Road Reading, RG1 1AX U.K.	+44 834 508311, ext 2239 p.janecek@xopen.co.uk	X/Open -- CAE Development Manager

Larry L. Johnson Texas Instr. P.O. Box 869305, ms 8406 Plano, TX 75086	214-575-5229 Fax: 214-575-3138 larry.johnson@dseg.ti.com	CALS Industry Working Group Enterprise Integration Working Committee (Vice-Chair)
Derek Kaufman 1010 El Camino Real Suite 380 Menlo Park, CA 94025	415-323-7992 ext 225 Fax: 415-323-8204 d.kaufman@xopusw.com	X/Open (Technical V.P, North America)
Margaret Law NIST CSL, Technology Bldg. Room A266 Gaithersburg, MD 20899	301-975-3255 Fax: 301-590-0932 law@ecf.ncsl.nist.gov	X3H6, CASE Tool Integration Models, ISO/IEC/JTC1/SC7/WG11
Jack Liu Digital Equipment Corp. 110 Spitbrook Road M/S ZK02-1/Q18 Nashua, NH 03062-2698	603-881-2590 Fax: 603-881-0120 jliu@tle.enet.dec.com	X3H4 IRDS, X3H6 CASE Tool Integration Models
George Maney CIMFLEX 1810 Embarcadero Road Palo Alto, CA 94303	415-424-0500 ext 334 Fax: 415-493-2645 gmaney@teknowledge.com	Rapid Response Manufacturing
Frank Manola G7E Laboratories 40 Sylvan Road, MS 62 Waltham, MA 02254	617-466-4289 Fax: 617-290-0627 fm02@gte.com	X3H7 Object Information Management
Erik Mettala DARPA SISTO 3701 North FairFax: Drive Arlington VA 22203-1714	703 696-2219 Fax: : 703 696-2202 Email: mettala@darpa.mil	DARPA SISTO (Deputy Director)
K. C. Morris NIST Bldg 220, A127 Gaithersburg, MD 20899	301-975-3081 kc@cme.nist.gov	ISO 10303 Standard for the Exchange of Product Model Data
Bruce Murrill, Tech. Director Network Mgmt. Forum 67, Corder Road Ipswich IP4 2XB Suffolk, England	+44 473 288595 !bmurrill@attmail.com	Network Management Forum (Technical Director)
Thomas R. Rhodes NIST Div 872 Bldg. 225, Rm B266 Gaithersburg, MD 20899	301-975-3295 Fax: 301-926-3696 trhodes@nist.gov	NIST Software Engineering Group, North American PCTE Initiative

Michael B. Richardson The Boeing Company M/S 7L-24 P.O. Box 24346 Seattle, WA 98124	407 783-0220 ext 340 jonesm@atc.boeing.com	X3H4 Information Resource Dictionary System
Richard Soley 492 Old Connecticut Path Framingham, MA 01701	508 820-4300 Fax: : 508 820-4303 Richard_Soley@omg.org	Object Management Group (Vice President and Technical Director)
John P. Solomond Director Ada Joint Program Ofc Pentagon 3E118 Washington, DC 20301	703-614-0209 Fax: 703-685-7019 solomond@ajpo.sei.cmu.edu	Portable Common Interface Set - -PCIS (Program Manager), ECMA TC33, (Liaison)
Geoff Speare 492 Old Connecticut Path Framingham, MA 01701	508-820-4300 Fax: 508-820-4303 geoff@omg.org	Object Management Group
Bruce Speyer MCC 3500 W. Balcones Center Dr. Austin, TX 78759-6509	512-338-3668 Fax: 512-338-3897 speyer@mcc.com	MCC IE Division, Enterprise Integration Pilots and Applications (Manager), EINet
Misako K. Sterbenz P.O. Box 2053 21500 Oakwood Blvd. Dearborn, MI 48121	313-594-4788 Fax: 313-337-5808 misako@pt0240.pto.ford.com	Rapid Response Manufacturing
Brian Stucke WL/MTIA Bldg 653 2977 P St., Ste 6 Wright-Patterson AFB, OH 45433	513-255-7371 Fax: 513-476-4420 stuckeba@mlgate.ml.upafb.af.mil	IDEF User's Group, U.S. Air Force ManTech Integration Division
Tom Ternus IBM Corp. CIM Integration Center Bldg. 235, 4404 1000 NW 51st Street Boca Raton, FL 33432	407-443-0644 Fax: 407-443-9367	Core Services Integration Project Group
Dr. Craig Thompson, Program Manager DARPA Open OODB Project Texas Instruments PO Box 655474, MS 238 Dallas, Texas 75265	214-995-0347 Fax: 214-995-0304 thompson@csc.ti.com	X3H7 Object Information Management, Object Management Group, Object Data Management Group X3 OODB Task Group
Gio Wiederhold DARPA SISTO 3701 North FairFax: Drive Arlington, VA 22203	703-696-2218 Fax: 703-696-2202 gio@darpa.mil	DARPA

Jim Willits Hewlett Packard Mailstop #102 3404 E. Harmony Road Ft. Collins, CO 80525	303-229-3926 Fax: 303-229-3526 j_willits@cnd.fc.hp.com	Network Management Forum, OMNIPoint (Planning Manager)
Jerry Winkler P.O. Box 2308 FairFax: , VA 22031	703-425-4558 Fax: (call first) jwinkler@nasamail.nasa.gov	X3H4 IRDS (Chairperson)

Appendix B. Call for Participation

Call for Participation
Workshop on Application Integration Architectures
February 8-12, 1993
Dallas, Texas

Background: In the last several years, many organizations, including both national and international standards bodies and industrial consortia, have developed architectures for application integration (frameworks, repositories, IPSEs, SEEs). Some of these efforts include Object Management Group, Open System Foundation, X/Open, Portable Common Tools Environment, CAD Framework Initiative, PDES/STEP, EIA CDIF, DARPA STARS, ECMA/NIST Reference Model for Frameworks of Software Engineering Environments, ODMG, X3 OODB Task Group, ANSI X3T3 ODP, ANSI X3H2 SQL, ANSI X3H4 IRDS, ANSI X3H6 CASE Integration Models, and ANSI X3H7 Object Information Management. Many of these efforts are based on or moving toward an object model as the foundation for data and service integration.

To deal with the open-ended requirements of application integration, a services architecture can provide a means of registering services implemented by a variety of facilities. Inheritance maximizes the potential for sharing and reuse of basic core services. Diverse services can then be obtained using a generic services interface. The interface acts as a neutral intermediary that shields service-requesting applications from irrelevant or unnecessary details of service-providing implementations. The interface provides the mechanism for offering a wide range of services using common request syntax and semantics.

Several important industry groups are planning to participate in the workshop. OMG and X/Open are joint sponsors. The workshop will be a neutral venue where diverse perspectives can be aired and possibly brought into better alignment. Individuals actively supporting several key consortia and standards committees are planning to attend.

Purpose of the Workshop: There is a growing need to converge and align the many application integration architecture efforts. The objective of this 5-day workshop is to construct a plan for how the participating organizations can cooperate to realize the shared vision of a common industry-wide integration architecture. This objective includes a focus on the potential for a generic services interface and API that would allow the services of different implementations to be provided in a consistent fashion. Expected outcomes of the workshop include:

- A taxonomy of services
- A directory of groups working on a service
- Requirements for a common interface and API for providing services
- Strategy for cooperative work

Attendee Information: The goal of the workshop will best be achieved if attendance is limited to those directly involved in technical development of the various efforts. Two to three invitations for each effort are available.

Workshop on Application Integration Architectures

Workshop Organization: The workshop will begin with a 1-day plenary session covering the workshop objectives. A spokesperson for each group will be requested to make a 30 minutes plenary presentation to cover mission statement, scope, constituency, history, liaison to other groups, technical progress and plans. The technical presentations should include a system architecture and descriptions of system components (or services). It should also list references to each group's standing documents.

During the remainder of the workshop, there will be parallel breakout sessions each focusing on specific topics. These may include:

- mission/goal statements (e.g., identify similarities, differences, ...)
- defining terms (e.g., open, portable, interoperable, service, etc.)
- systems architecture comparison
- API, services interface and request brokering
- data/object models
- how repository and framework efforts relate
- template for (object) services (e.g., requirements, reference model, generic interface specification, programming language bindings, relevant standards, ...)
- categorization of (object) services (who's doing what)
- focus on a specific object service (e.g., change management, ...) to provide critical comment from multiple points of view.
- how different are domain-specific (CAD, CASE, ...) frameworks
- convergence plan formulation (actions, benefits, risks)

Workshop Preparation: Each workshop attendee must represent (informally---no commitments are required as part of the workshop) one or more standards groups, consortia, or multi- organization (including government) projects. To participate, individuals need to provide beforehand (1) one reference copy of current source documentation on their group's effort (one copy from each group represented), (2) an architecture diagram (or other description of the system), (3) recommendations for potential breakout sessions (paragraph on each), (4) copy of a presentation on their group's activity if they are the group's spokesperson (see above), and (5) a Position Paper (2--5 pages). Position papers may focus on approaches to converging work in this area, evaluation of areas where consensus that could lead to standards is high, or on some (proposed) breakout session topic.

Benefits of Participation: The benefits of this workshop are expected to be to the individual groups, measured in terms of information and action. Participants should come away with a much clearer understanding of how their organization can contribute to a major industrial move toward open application integration architectures and how to leverage other groups' efforts. In addition, a concrete action plan should result for converging some efforts or closing gaps. A Workshop Report, assembled during and immediately following the workshop by the participants will capture inputs from plenary and breakout sessions.

Submission: Persons wishing to attend this workshop should provide an early response of your intent as soon as possible. Include with the response your name, address, voice phone, fax phone, e-mail, and group(s) represented. Attendees should then submit workshop materials (described above) to the workshop coordinators listed below.

Bob Hodges
Texas Instruments Inc.
6550 Chase Oaks Blvd.
PO Box 869305, MS 8482
Dallas, TX 75023
e-mail: 4937603@mcimail.com
214-575-3442

Craig Thompson
Texas Instruments Inc.
13510 N. Central Expressway
PO Box 655474, MS 238
Dallas, TX 75265
e-mail: thompson@csc.ti.com
214-995-0347

Hotel Arrangements: The Application Integration Architectures Workshop will be held at the Dallas Parkway Hilton in Dallas, TX. A block of rooms for workshop attendees is being held at the government rate of $74.00 (available to all attendees) until January 24, 1993. Please make reservations directly with the hotel by calling (214)661-3600 or (800)356-3924; mention the **"AIA Workshop"**.

The Dallas Parkway Hilton is located just west of the intersection of I-635 and the Dallas Parkway and is visible from I-635. If traveling to the hotel by car, take the north DFW airport exit, travel east from DFW Airport on I-635 about 15 miles, exit north at Dallas N. Parkway, then go west a short distance on the I-635 service road, and exit at the hotel. Alternatively, one-way ground transportation from the airport is available for around $25 (taxi) or $11-$12 (by supershuttle, using the courtesy phone in the luggage area).

The Hilton provides an hourly shuttle to the nearby, posh Galleria Shopping Center, where many shops and restaurants are available.

Workshop Fee: The workshop fee is expected to be approximately $75. The exact fee will be determined at the workshop based on actual costs and the number of participants. The fee will not be prorated on number of days attended. Please plan to pay by check (not plastic) at the door.

The workshop fee will be used to cover the costs of the meeting facilities and refreshments. A continental breakfast and afternoon beverages will be included. No other meals will be provided. There will be a reception the evening of Feb. 8.

Appendix C. Workshop Program

1. Preliminary Schedule (template) for the AIA Workshop

Feb. 8 (Monday)

7:30 Registration and Continental Breakfast
8:30 Plenary -- Welcome and Opening Remarks -- Thompson and Hodges
9:00 Plenary -- Presentations by Groups (see below)
10:00 Break
10:20 Plenary -- Presentations by Groups (cont.)
12:00 Lunch on your own
1:20 Plenary -- Presentations by Groups (cont.)
3:20 Break
3:40 Plenary -- Working Session: Workshop Goals and Outcomes
5:30 Adjourn
6:30 Reception (until 8:30 PM -- cash bar)

Feb. 9 (Tuesday)

7:00 Registration and Continental Breakfast
8:00 Plenary -- Presentations by Groups (cont.)
10:00 Break
10:20 Plenary -- Presentations by Groups (cont.)
12:00 Lunch on your own
1:20 Plenary -- Presentations by Groups (if needed)
3:20 Break
3:40 Plenary -- Working Session: Management Issues I
5:30 Adjourn
7:30 Birds of a Feather Sessions (optional)

Feb. 10 (Wednesday)

8:00 Breakout Session #1, #2, #3, #4 (parallel sessions -- see below)
12:00 Lunch on your own
1:30 Breakout Session #5, #6, #7, #8 (parallel sessions -- cont.)
5:30 Adjourn
7:30 Birds of a Feather Sessions (optional)

Feb. 11 (Thursday)

8:00 Breakout Session #9, #10, #11, #12 (parallel sessions -- cont.)
12:00 Lunch on your own
1:30 Breakout Session #13, #14, #15, #16 (parallel sessions -- cont.)
5:30 Adjourn
7:30 Birds of a Feather Sessions (optional)

Workshop on Application Integration Architectures

Feb. 12 (Friday)

8:00 Plenary -- Workshop Result Statement, Outcomes, Next Steps
8:00 Parallel Breakouts (optional)
12:00 Adjourn

2. Plenary -- Presentations by Groups

Monday and Tuesday will be dominated by descriptions of the groups represented at the workshop. The purpose is to insure that attendees develop a clear idea of the charter, progress, plans, roadmap, and liaisons of each group. Each session listed below will have 20 minutes. We recommend a concise 15 minute presentation plus 5 minutes for questions, but speakers are free to restrict questions and use all 20 minutes for presentations. Speakers are also free to use less than 20 minutes or to "pass" if they do not wish to make a presentation.

In the list below, we assume THE FIRST PERSON LISTED WILL MAKE THE PRESENTATION. As far as the workshop schedule goes, we plan to do late binding on who will present, but the people listed in a session should coordinate with each other beforehand to make sure a speaker is identified. If no presenter for a session attends the workshop, we will take up to 10 minutes of comments from the audience on any known status about that group's activity.

1. ICEIMT (International Conference on Enterprise Integration Modeling Technology) -- Bruce Speyer
2. X/Open -- Derek C. Kaufman, Vic Goddard, Peter Janacek
3. Unix International -- Jack Bissell, Larry Brown
4. OSF -- invited but not planning to attend at this time
5. ISO Network Management Forum (NMF) -- Bruce Murrill
6. OMG -- Geoffrey R. Lewis, Craig Thompson, Richard Soley
7. ECMA TC33 (PCTE), ECMA/TGEP, PCIS or NAPUG, ECMA/TC33-TGRM, North American PCTE Initiative -- Hugh Davis, Ian Thomas
8. CDIF -- Mike Imber
9. ISO/IEC JTC1/SC21 (Ad Hoc Group on APIs) -- Jon Becker
10. ISO/IEC JTC1/SC7/WG11 (Description of Data for SW Eng.) -- Mike Imber, Margaret Law
11. ISO/IEC JTC1/SC21/WG3 (Reference Model of Data Management) -- Elizabeth Fong
12. ANSI X3H2 (Database), ISO/IEC JTC1/SC21/WG3 (SQL DBL Rapporteur Group) -- Jim Melton, David Beech
13. ANSI X3H4 (IRDS), ISO/IEC JTC1/SC21/WG3 (IRDS Rapporteur Group) -- Jerry Winkler, Bob Hodges, Roger Burkhart, Jack Liu
14. ANSI X3H6 (CASE Tool Integration Models) -- Bill Harrison, Jack Liu, Hal Pierson, Kathy Chapman
15. ANSI X3H7 (Object Information Management) -- Bill Kent, Frank Manola, Craig Thompson, Elizabeth Fong
16. ANSI X3T3 (ODP), ISO/IEC JTC1/SC21/WG7 (ODP Reference Model) -- Eng Chew, Ed Stull, Cal Taylor
17. ANSI X3T5 (OSI), ISO/IEC JTC1/SC21/WG4, 5, 6 and SC27/WG28 -- Henry Lowe
18. DoD Corporate Information Management (CIM) Initiative - Fred Hathorn
19. CALS Industry Steering Group Information Integration Working Group (IIWG) -- Larry Johnson
20. DARPA programs -- Gio Wiederhold, Erik Mettala, Bob Balzer, Jay M. Tenenbaum
21. DARPA STARS -- John Foreman
22. Software Engineering Institute -- Alan W. Brown, David Carney
23. Navy PSESWG -- Alan W. Brown, David Carney, Patricia Oberndorf
24. USAF Integration Toolkit and Methods (ITKM) Program -- Brian Stucke
25. ODMG -- Mary Loomis
26. CASE Communique -- Edie Bailey, Eric Black

27. PDES/STEP -- K. C. Morris, Yuwai Yang, Hank Noel
28. Rapid Response Manufacturing Consortium -- William (Bill) Cain, Neil Christopher
29. CAD Framework Initiative -- Tom Rhyne
30. SEMATECH -- Glenn Hollowell
31. POSC-- Alan Doniger
32. MCC CAx Consortium -- Ken Drake

3. Plenary Working Sessions and Breakout Sessions

Plenary working sessions on Monday and Tuesday will focus on Workshop Outcomes and on Management Issues respectively. The Final Plenary Session will focus on Workshop Results and Next Steps.

Wednesday and Thursday will be breakout sessions, which will consist of four parallel tracks each ending in a plenary summary. Most breakout sessions will have the following structure (below). Some breakout sessions may be continuations on a topic (e.g., topic name I, II, III, ...) and may have a simpler session structure.

Template for a Typical Breakout Session:

20 min.	refine topic and scope, identify issues and outcomes for this session; list attendees; select scribe and presenter (same or different persons)
120 min.	group discussion on breakout topic
10 min.	summary statement from each attendee
30 min.	break for session attendees, presenter completes foils
60 min.	plenary - 15-20 minute summary per breakout session by session presenter
Total	4 hours

We have reserved 4 rooms for breakouts: 1 large, 1 medium, and 2 small. The breakout sessions listed below are currently planned. But it is the purpose of the late afternoon session on Monday ("Workshop Goals and Outcomes") to revise this list and the deliverables of each session. Please come to the workshop prepared with suggestions for improved or alternative breakout sessions.

Monday later afternoon

Plenary Working Session: Workshop Goals and Outcomes

- why is this workshop important to your organization? what significant events will happen if differences are not resolved?
- maximizing workshop value--during and after; review proposed workshop objectives, breakouts, deliverables; suggest changes; dynamic replanning allowed.

Tuesday later afternoon

Plenary Working Session: Management Issues I

— this is the first in a Management Track of breakout sessions; it is plenary since it is important for all workshop participants to be sensitive to the fact that it is not just technical differences that divide different groups. The key idea is to begin to organize each group's schedule so outcomes can feed other groups. Working together involves identifying barriers and roadblocks; these may stem from similarities and differences in sociology, politics, length of process, governing laws, copyright issues, liaison rules, approaches (specification by committee, by selection, by construction), how much user influence, timelines of different groups, government role, consortium role, standards role, etc.

— what do our "customers" want; which of their problems are we solving (e.g., interoperation of solutions based on standards); are we creating new problems

— deliverables:

- identification of roadblocks (e.g., creeping scope problem, identify
- where consortia and standards are (not) aligned?);
- identification of risk mitigation strategies;
- Roadmap showing schedules and liaisons;
- action items (e.g., MOAs);

Wednesday morning breakouts

Breakout #1: Management II (continues Management I)

— deliverable: refine Roadmap showing when each producer group expects to completes technology and consumer group expects to adopt/adapt technology, e.g., comparison matrix of PCTE, OMG, OSI, UI, ...

Breakout #2: Object Model I

— why "Yet Another Object Model"?; are different object models needed for different purposes? (e.g., C++, Smalltalk, ..., IDL, EXPRESS, SQL3, ...)
— classification of features of object models
— core model proposal
— deliverable: Object Model Features Matrix

Breakout #3: Object Services Architecture I

— deliverable: definitions, design guidelines, architecture diagrams;
— Service Coverage Matrix (who's doing what & in what depth when);
— Service Description Template;
— Service Dependency Matrix;
— description of specific services.

Breakout #4: Interchange Formats

— specification languages
— relationships to object modeling

— deliverables: Application Domain Coverage Matrix/Map (to cover PDES/STEP, CDIF, ...); integration issues and strategies.

Wednesday afternoon breakouts

Breakout #5: Management III (continues Management II)

Breakout #6: Object Model II (continues OM I)

Breakout #7: Object Services Architectures II (continues OSA II)

Breakout #8: Definition of Terms: Repository, Framework, Service Architecture, Interoperation, Open, ...

Thursday morning breakouts

Breakout #9: Management IV

Breakout #10: Object Model III

— strategies for interoperation of different object models
— harmonization versus integration

Breakout #11: Object Services Architectures III

— can same service (e.g., change management, transactions, ...) generically serve multiple domains
— how are domain specific frameworks different for CASE, CAD, ... CAx?
— how will different frameworks interoperate?

Breakout #12: Specification Methodologies

— reference models, API issues, language bindings, services template, module interconnect formalism, specification languages, referencing other standards, etc.

Thursday afternoon breakouts

Breakout #13: Management V (continues Management IV)

Breakout #14: Infrastructure Issues

— messaging and request brokering models
— CORBA, RPC, OSI, ... relationships
— network, OS, programming language bindings

Breakout #15: tbd based on progress

Breakout #16: tbd based on progress

Friday morning breakouts

Plenary -- Workshop Results and Next Steps

— what is the right thing to happen after the workshop
— summary statements from participants

— draft workshop summary statement (progress we made, next steps)
— review of action items

4. Reception and Birds-of-a-Feather Sessions

A reception is planned for Monday evening 6: 30 to 8: 30 p.m. In addition to scheduled daytime sessions, we are leaving Tuesday, Wednesday, and Thursday evenings free for birds-of-a-feather sessions. Anyone can organize these sessions and reserve a room. We will make two rooms available each evening. These may be open or closed sessions though room preference will be given to open sessions.

5. Workshop Report

The AIA Workshop Report will consist of the following:

- Introduction Thompson and Hodges
- Plenary and Breakout Session Reports -- 1-3 pp. each -- session scribes
- Conclusion Thompson, Hodges, volunteers tbd
- Appendix A Position Papers by Workshop Attendees (if received by workshop)
- Appendix B Presentations by Workshop Attendees (PLS. BRING A HARDCOPY)
- Appendix C List of Attendees
- Appendix D Revision Comments (if any, see below)
- Appendix E other tbd (e.g., session deliverables, ...)

We plan to complete the workshop report according to the following schedule.

26 Feb.	session scribes send sections to thompson/hodges
19 Mar.	draft report completed and sent to attendees for review
29 Mar.	review period ends (comments to be sent to thompson/hodges); correction comments will be incorporated in-line, controversial comments will appear in an appendix.
12 Apr.	distribution of workshop report by [*TBD*]

Appendix D. Workshop Document Register

The following documents were available during the Application Integration Architectures Workshop. Copies of documents must be obtained from individuals or groups who provided them to the workshop.

1. Application Integration Architectures Workshop Document Register
2. Application Integration Architectures Workshop Call for Participation
3. Application Integration Architectures Workshop Program
4. Application Integration Architectures Workshop Attendance Roster
5. Application Integration Architectures Workshop Template for a Typical Breakout Session
6. Presentation - Application Integration Architectures Workshop Management Sessions
7. Presentation - Application Integration Architectures Workshop Object/Data Model Sessions
8. Presentation - Application Integration Architectures Workshop Architecture Sessions
9. Presentation - Application Integration Architectures Workshop Roadmap Sessions
10. Software Infrastructure Landscape Diagram, Craig Thompson (see Figure 1)
11. Presentation - Application Integration Architectures Workshop Introduction
12. Presentation - International Conference on Enterprise Integration Modeling Technology
13. Presentation - Open Systems and X/Open
14. Presentation - Unix International Distributed Application Development
15. UI December Status Report for DADSIG
16. UI RMSC Investigative Team Report 5.0, October 1992
17. Presentation - Network Management Forum
18. Position Paper - The Roadblocks to Achieving Integrated Management of Networked Information Systems, Bruce Murrill (see Appendix G)
19. An Open Management Roadmap, Part 1, Network Management Forum, Draft, 14 Feb 92
20. Comparison of the OMG and ISO/CCITT Object Models, Report of the Joint Forum/OMG Task Force on Object Modeling, Editor, Colin Ashford, February, 1993
21. OSI Architecture and Managed-Object Model, Colin Ashford, February, 1993
22. Network Management forum OMNIPoint Starter Kit Order Form
23. Presentation - Object Management Group
24. OMG Publications Order Form
25. Presentation - North American PCTE Initiative
26. Position Paper - PCTE Issues, Hugh Davis (see Appendix G)
27. Presentation - CASE Data Interchange Format (CDIF)
28. CDIF - Framework for Modeling and Extensibility, Draft, October, 1992
29. Presentation - ISO/IEC/JTC1/SC7/WG11 Description of Data for Software Engineering
30. Position Paper - WG11, Description of Data for Software Engineering,, Peter Eirich, Convenor, JTC1/SC7/WG11 (see Appendix G)
31. Position Paper - IEEE-CS P1175, Task Force on Professional Computing Tools, Peter Eirich (see Appendix G)
32. Position Paper - IEC TC 93, Design Automation, Peter Eirich (see Appendix G)
33. Presentation - ISO/IEC 10032 Reference Model for Data Management (RMDM)
34. Position Paper - RMDM Issues, Bill Olle (see Appendix G)
35. Objects in ANSI/ISO SQL3, David Beech
36. Object SQL: Language Extensions for Object Data Management, Leonard J. Gallagher
37. Database Language SQL: Integrator of CALS Data Repositories, Leonard Gallagher, Joan Sullivan
38. Presentation - IRDS Standards Direction
39. Position Paper - IRDS, Roger Burkhart, Bob Hodges, and Jerry Winkler (see Appendix G)
40. IRDS Services Architecture Technical Report Working Outline
41. IRDS Service Coverage Matrix
42. Presentation - X3H6, CASE Tool Integration Models

43. Position Paper - X3H6 CASE Tool Integration Models (see Appendix G)
44. Proposal to Develop a New X3 Standard, Tool Integration Messages
45. Position Paper - The Need for Object Model Interoperability, Frank Manola (see Appendix G)
46. X3H7 Object Model Features Matrix, Draft, November, 1992
47. An Overview of Open Distributed Processing, ISO and CCITT, Eng Chew
48. ODP Prescriptive Model Advances to CD Status, Cal Taylor, from First Class, OMG Newsletter, Vol. II, Issue 5
49. Presentation - CALS Industry Steering Group, Information Integration Working Group
50. Presentation - DARPA Persistent Object Bases Program
51. Presentation - NGCR Project Support Environment Reference Model
52. NGCR PSESWG Overview
53. NGCR Reference Model for Project Support Environments, Version 0.9, Feb. 2, 1993
54. Issues in the Development of a Project Support Environment Reference Model, Alan W. Brown, David J. Carney, Peter H. Feiler, Patricia A. Oberndorf, December, 1992
55. Presentation - CASE Communique
56. Presentation - Overview of the Standard for the Exchange of Product Model Data
57. Presentation - ISO/TC184/SC4 , Standard for the Exchange of Product Model Data
58. Presentation - Rapid Response Manufacturing Consortium
59. Presentation - SEMATECH
60. Presentation - Domain Specific Support Architectures (DSSA)
61. IRDS Context Reference Model Technical Report, September 1992
62. Template for Describing Object Services
63. Position Paper - Overview of the STEP and the STEP Standard Data Access Interface (see Appendix G)
64. Position Paper - NCMS Rapid Response Manufacturing Program (see Appendix G)

Appendix E. Group Information Templates

Group Name: X/Open Co., Ltd.

Contact:

Vic Goddard
Apex Plaza
Forbury Road
Reading
RG1 1AX
UK
44 734 508311 x2255
v.goddard@xopen.co.uk

Mission:

To bring to users greater value from computing through the practical implementation of open systems.

Classification:

Vendor Consortium with User influence

Integration Domain:

Current Focus: Enterprise Integration

Areas of Applicability:

Areas of Technology:

Focus Area: Data Models, Object Models, Interchange Formats, Repositories, Database Management, Operating System, Distribution, Name Space, Programming Language, Graphical User Interface

Dependent On:

Are you basing your work on OO Technology? Yes

Can your work be applied to non-OO Technology? Yes

Which Object Model do you use? OMG

Group Deliverables:

Information not available at this time

Dependencies Liaison:

Information not available at this time

Group Name: SEMATECH

Contact:

Glenn Hollowell
2706 Montopolis Dr.
Austin, TX 78741-6499
(512) 356-7166
Fax: (512) 356 - 3575
glenn_hollowell@sematech.org

Mission:

Focus US industry resources to deliver manufacturing systems that enable US companies to lead in semiconductor manufacturing. Scope of this work includes: equipment control systems, factory control systems, and factory automation systems.

Classification:

Non-government Industry Consortium

Integration Domain:

Current Focus: Computer Integrated Manufacturing, Software Development Process

Areas of Applicability:

Areas of Technology:

Focus Area: Process Models, Repositories, Distribution, Methodology, Requirements, Metrics

Dependent On: Message Models, Tool Models, Database Management, Operating Systems, Name Space

Are you basing your work on OO Technology? Yes

Can your work be applied to non-OO Technology? Yes

Which Object Model do you use? OMG

Group Deliverables:

Dependencies Liaison:

Group Name: X3H7 - Object Information Management

Contact:

Glenn Hollowell
2706 Montopolis Dr.
Austin, TX 78741-6499
(512) 356-7166
Fax: (512) 356 - 3575
glenn_hollowell@sematech.org

Mission:

Evaluate object technology usage across numerous standards organizations (accredited and consortia) who are defining object technology extensions to their domain-specific standards. Find common ground for influencing convergence of this object technology usage.

Classification:

ANSI Accredited Standards Development Technical Committee

Integration Domain:

Current Focus: Cross-Domain Object Technology

Areas of Applicability:

Areas of Technology:

Focus Area: Object Models

Dependent On: Data Models, Message Models used in Tool Models, Interchange Formats, Repositories, Database Management, Operating System, Distribution, Name Space, Methodology, and Requirements

Are you basing your work on OO Technology? Yes

Can your work be applied to non-OO Technology? Yes

Which Object Model do you use? Evaluating numerous models

Workshop on Application Integration Architectures

Group Deliverables:

Title	Description	Target Date
Technical Report	Comparison of Object Models, Object Reference Models, and recommendations for harmonizing object technology usage in domain specific standards.	

Dependencies Liaison:

Group	Product Technology	Relationship
X3H2 (SQL)	Object Technology	Close Coordination
X3T3 (ODP)	Object Technology	Close Coordination
X3H4 (IRDS)	Object Technology	Close Liaison
X3J4 (COBOL)	Object Technology	Close Liaison
X3J9 (PASCAL)	Object Technology	Close Liaison
X3J13 (LISP)	Object Technology	Close Liaison
X3J16 (C++)	Object Technology	Close Liaison
X3T2 (ASN-CLID)	Object Technology	Close Liaison
X3T5.4 (NM)	Object Technology	Close Liaison
X3H6 (CTIM)	Object Technology	Close Liaison

Group Name: X3H2 - Database Language SQL

Contact:

David Beech
Oracle Corp.
Redwood Shores, CA 94065
(415) 506 - 6420
Fax: 506 - 7203
dbeech@oracle.com

Mission:

Develop SQL beyond SQL-1992 including object extensions.

Classification:

ANSI Accredited Standards Development Technical Committee

Integration Domain:

Current Focus:

Areas of Applicability: CAD/CAM, CASE, Enterprise Integration, Office Automation, Management

Areas of Technology:

Focus Area: Data Models, Object Models, Process Models, Database Management, Distribution, Name Space

Dependent On: Repositories, Programming Languages

Are you basing your work on OO Technology? Yes

Can your work be applied to non-OO Technology? Yes

Which Object Model do you use? SQL3

Group Deliverables:

Title	Description	Target Date
Database Language SQL	Standard ("SQL3")	1995

Dependencies Liaison:

Group	Product/Technology	Relationship
X3H4	IRDS	Liaison
X3H7	OIM	Liaison
X3J16	C++	Liaison

Group Name: X3H4 - Information Resource Dictionary System (IRDS)

Contact:

Jerry Winkler
PO Box 2308 Fairfield, VA 22032
(703) 425 - 4558
jwinkler@nasamail.nasa.gov

Mission:

Provide standards for services and facilities needed to specify, integrate and manage an enterprise's information resources and assets.

Classification:

ANSI Accredited Standards Development Technical Committee

Integration Domain:

Current Focus:	Enterprise Integration
Areas of Applicability:	CASE, Product Data Management, Office Automation, Electronic Commerce

Areas of Technology:

Focus Area:	Data Models, Object Models, Process Models, Repositories, Name Directory, Requirements
Dependent On:	Interchange Formats, Database Management, Distribution

Are you basing your work on OO Technology? Yes

Can your work be applied to non-OO Technology? Yes

Which Object Model do you use? TBD

Workshop on Application Integration Architectures

Group Deliverables:

Title	Description	Target Date
IRDS Services Architecture Technical Report	Technical Report defining the architecture for separately developed standards for core model, base services and content modules for IRDS.	1/94
US Contribution to IS 10728 IRDS Services Interface	Change proposals to existing ISO standard introducing OO concepts into IRDS.	11/93
IRDS Conceptual Schema Normative Schema Language	Logic-based approach for capturing semantics of IRDS content.	6/94

Dependencies Liaison:

Group	Product/Technology	Relationship
ISO IRDS RG	Requirements, Change Proposals, technical Review	Supporting
X3H2 SQL	SQL2 and SQL3	Using deliverable
X3H6 CTIM	Repository	Supplied
CDIF CASE Info Model	Export/Import	Using
X3H7	Object Model Concepts	Using
DARPA KIF	Knowledge Interchange	Cross-communication
PCTE		Needed

Group Name: Object Management Group (OMG)

Contact:

Geoff Speare
492 Old Connecticut Path
Framington, MA
(508) 820 - 4300
Fax: (508) 820 - 4303
geoff@omg.org

Mission:

To promote cross-platform interoperability using object technology.

Classification:

Vendor/User Consortium

Government and Industry Participants

Integration Domain:

Current Focus: All domains

Areas of Applicability: All domains

Areas of Technology:

Focus Area: Data Models, Message Models, Object Models, Tools Models, Interchange Formats, Repositories, Database Management, Distribution, Name Space, Methodology

Dependent On:

Are you basing your work on OO Technology? Yes

Can your work be applied to non-OO Technology? No

Which Object Model do you use? OMG Object Model

Workshop on Application Integration Architectures

Group Deliverables:

Title	Descriptions	Target Date
CORBA	Specification	1.1. now; 2.0 first half 1994
Object Model Core Specification		now
Object Model Components Specification		ongoing
Object Model Services Specification		ongoing, starting 1H93

Dependencies Liaison:

Informal liaisons with many related groups

Group Name: NCMS - Rapid Response Manufacturing Program (RRM)

Contact:

Bill Waddell, RRM Program Manger
National Center for Manufacturing Sciences
3025 Broadway
Ann Arbor, MI 48108
(313) 995-0300
Fax: (313) 995 - 4004
billw@ncms.org

Mission:

This program provides the needed to effectively enable engineers to reduce the time required to design and manufacture products in response to rapidly fluctuating market demands. The objective of the program is to shorten time-to- market, improve quality-to-cost, and enhance product reliability in order to provide the US manufacturing infrastructure competitive advantage in a variety of global market sectors. Rapid Response Manufacturing will be accomplished by coordinating and extending the application of feature-based solids modeling, knowledge-based systems, integrated data management, and direct manufacturing technologies in a cooperative computing environment. Progress of the program will be measured by the design and fabrication of a different family of parts at the site of each participating manufacturer.

Classification:

User Consortium

Government and Industry Partnership

Integration Domain:

Current Focus: CAD/CAM/CAE

Areas of Applicability: Enterprise Integration

Workshop on Application Integration Architectures

Areas of Technology:

Focus Area:	Data Models, Tool Models, CAD/CAM/CAE Applications, Manufacturing Process Models, Metrics
Dependent On:	Message Models, Object Models, Interchange Formats, Database Management, Operating System, Distribution, Name Space, Programming Language, User Interfaces, Graphical User Interfaces, Methodology, Requirements

Are you basing your work on OO Technology?

Can your work be applied to non-OO Technology?

Which Object Model do you use? Still Under Evaluation

Group Deliverables:

Title	Description	Target Date
Reference Architecture	A detailed reference model specifying data models, data management and protocols for CAD/CAM/CAE integration	9-94
Integrated Product and Process Model	Reference model of mechanical design and manufacturing process information	9-97
Engineering Environments	Test and validation environments participant site	9-97
Knowledge-Based Applications	A suite of applications such as process planning and variant design	9-97
Design and mfg. libraries	Libraries of mechanical features, materials and manufacturing processes	6-95

Dependencies Liaison:

Group	Product/Technology	Relationship
PDES/STEP	Data Representation	PDES Testbed at NIST

Workshop on Application Integration Architectures

Group Name: North American PCTE Initiative (NAPI)

Contact:

Fred Hathorn
1225 Jefferson Davis Highway
Suite 910
Arlington, VA 22202
(703) 746-7924
Fax: (703) 746-7396
fhathorn@ddi.c3i.osd.mil

Mission:

Provide a forum for North American Interests in the PCTE Standard

Classification:

Standards Development Organization

Government and Non-government Participation

Integration Domain:

Current Focus: CASE

Areas of Applicability:

Areas of Technology:

Focus Area: Data Models, Message Models, Object Models, Tool Models, Process Models, Interchange formats, Repositories, Name Space, Distribution

Dependent On:

Are you basing your work on OO Technology? Yes

Can your work be applied to non-OO Technology? Yes

Which Object Model do you use?

Workshop on Application Integration Architectures

Group Deliverables:

Title	Description	Target Date
PCTE Extensions	Updated Standard. Specific components are expected to include OO capabilities, efficient fine-grained object management services and harmonization with OMG	
CORBA Validation Test Suite	Validation test suite for measuring PCTE conformance	

Dependencies Liaison:

Group	Product/Technology	Relationship
OMG	CORBA, ORB	harmonizing technology
ECMA/TC33	PCTE Specification	baseline document
X3H4	IRDS standard	cross-communicate
X3H6	messages	cross-communicate
X3H7		

Group Name: Network Management Forum (NMF)

Contact:

Bruce Murrill
67 Corder Road
Ipswich
Suffolk IP4 2 XB
UK
44 473 288 595
Fax: 44 473 288 595
bmurrill@attmail.com

Mission:

To accelerate the availability of management solutions for networked information systems.

Classification:

Vendor/User Consortium

Integration Domain:

Current Focus: Management

Areas of Applicability:

Areas of Technology:

Focus Area: Object Models

Dependent On: Name Space, Programming Language, Methodology, Requirements

Are you basing your work on OO Technology? Yes

Can your work be applied to non-OO Technology?

Which Object Model do you use? OSI Management Information Model ISO/IEC 10165-1

Workshop on Application Integration Architectures

Group Deliverables:

Title	Description	Target Date
OMNIPoint 2	Set of implementation agreements and specifications for the management information systems	Fourth Quarter '94

Dependencies Liaison:

Group	Product/Technology	Relationship
X/Open		Partner
OSF		Partner
OMG		Partner
UI		Partner
NIST		Partner
CCTA (UK)		Partner

Workshop on Application Integration Architectures

Group Name: Electronic Industries Association (EIA) CASE Data Interchange Format Technical Committee

Contact:

Mike Imber
LBMS 62 Oxford St. London, W1N 9LF, UK
(44) 71 636 4213
Fax: (44) 71 636 2708
100031.700@compuserve.com

Mission:

To produce standards family to support exchange of CASE information between tools and repositories.

Classification:

ANSI Accredited Standards Development Organization

Integration Domain:

Current Focus:	CASE
Areas of Applicability:	CASE, possibly CAD/CAM, Enterprise Integration, Management

Areas of Technology:

Focus Area:	Data Models, Interchange Formats
Dependent On:	

Are you basing your work on OO Technology?	No
Can your work be applied to non-OO Technology?	Yes, ERA Data Model
Which Object Model do you use?	

Workshop on Application Integration Architectures

Group Deliverables:

Title	Description	Target Date
CDIF Overview - Framework		
Transfer Format - General Rules		
Transfer Format - Syntax		
Transfer Format - Encoding		
Integrated Meta-Model - Foundation		
Common Data Modelling - Data Flow Model, Data Inventory, Presentation Location, Presentation Shape, Common Presentation	Standards, Interim Standard & Proposed Standards	June 1993
	EIA/ANSI Standard	Mid-94
Integrated Meta Model - State/Event, Physical Relational DBMS Standards	Interim Standard	Late '93
Integrated Meta Model - User Interface, Program Structure, Logical Network, DBMS Physical Network, DBMS File Network, DBMS Constraints, Presentation - Repeating Struct.	Standards, Interim Standard	Late '94
Integrated Meta-Model - Logical Hierarchical DBMS, Physical Hierarchical DBMS, Project MGMT - Estimating, Tracking, Testing,	Standards, Interim Standard	Post '94

Dependencies Liaison:

Group	Product/Technology	Relationship
X3H4	IRDS Import/Export	Delivering Solution to them
ISO/IEC JTC1/SC21/WG3	IRDS, Possible IRDS Import/Export	Delivering Solution to them
ECMA TC33	Possible PCTE Import/Export	Delivering Solution to them

ISO/IEC JTC1/SC7/WG11 CASE Information Meta-Model	Delivering Base Document	
PDES/STEP SPC	CASE Information Meta- Model	Delivering Base Document
IEEE P1175	CASE Information Meta- Model	Cross-Communication
JTC1/SC22/WG11	CLID Standard	Using as layout to Meta- Model
Codasyl FIMS	FIMS Model	Using as layout to Meta- Model

Group Name: Case Communique (CCQ)

Contact:

Edie Bailey
3404 East Harmony Rd. MS#7
Fort Collins, CO 80525 - 9599
(303) 229 - 6160
Fax: (303) 229 - 6611
bailey@fc.sde.hp.com

Mission:

Provide an open forum dedicated to the cooperative development of industry acceptable standard specifications for control integration in CASE and application framework environments based on user requirements.

Classification:

Standards development

Non-government vendor/user consortium

Integration Domain:

Current Focus: CASE

Areas of Applicability: CAD/CAM, Enterprise Integration, Office Automation, Management

Areas of Technology:

Focus Area: Message Models, Requirements

Dependent On: Data Models, Object Models, Tool Models, Process Models, Interchange Formats, Repositories, Database Management, Operating System, Distribution, Name Space, Programming Language, User Interfaces, Graphical User Interfaces, Methodology, Metrics

Are you basing your work on OO Technology? Yes

Can your work be applied to non-OO Technology? Yes

Which Object Model do you use?

Workshop on Application Integration Architectures

Group Deliverables:

Title	Description	Target Date
Architecture for control Integration	Description of constraints and guidelines for supporting and creating standard specifications for request/notification based message communication between applications	Mid-1993, Requires agreement between CCQ, CIA, OMG, CFI and X3H6
Operation Specifications (OP Specs) for common CASE/framework requests/notifications	Evolving development of abstract descriptions of request/notification specifications for mapping to messages supported by actual framework technologies	First draft op specs due out mid-1993 initial op specs available today

Dependencies Liaison:

Group	Product/Technology	Relationship
CASE Interoperability Alliance	Message Architecture	Direct Interaction through joint committee (aligning toward a single standard for control integration)
OMG	message architecture	see above
CFI	message architecture	see above
ANSI X3H6	Message standardization	Influencing through contributions - liaison in progress
CDIF	Data Interchange Formats/Data Defs	Using CDIF work as applicable
Standards Coordination Activities		Cross-Communicate

Group Name: CALS Enterprise Integration Working Committee

Contact:

Larry L. Johnson
Texas Instruments
PO Box 869305
Plano, TX
(513) 575 - 5229
Fax: (513) 575 - 3138
or David Judson
(513) 255 - 7371

Mission:

Provide a profile of enterprise integration spanning cultural, organizational and technical issues

Classification:

Vendor/User Consortium

Government and Industry Partnership

Integration Domain:

Current Focus:	Enterprise Integration
Areas of Applicability:	CAD/CAM, CASE, Office Automation, Management

Areas of Technology:

Focus Area:	
Dependent On:	Data Models, Message Models, Object Models, Process Models, Interchange Formats, Repositories, Database Management, Operating System, Name Space, Methodology, Metrics

Are you basing your work on OO Technology? No

Can your work be applied to non-OO Technology? No

Which Object Model do you use?

Workshop on Application Integration Architectures

Group Deliverables:

Title	Description	Target Date
Profile for Enterprise Integration	White Paper	12/92 (completed)
	Technical Paper	6/93
	Validation White Paper	10/93
Activities Report	CALS Expo	11/93
Dependencies Liaison:		
Group	Product/Technology	Relationship
PDES/STEP		Using
IRDS		Using
OMG		Using

Appendix F. Workshop Evaluations

Workshop attendees were asked to complete a workshop evaluation form covering:

- value of workshop; plenary/breakout format; improvements for next time; other.
- final thoughts on: integration issues; concerns; recommendations; action items; next steps; other.

This appendix anonymously lists workshop attendees comments, sorted by topic. All comments received are listed verbatim with only minor editorial and formatting changes.

F.1 Value of the Workshop

"I support any further initiatives of this kind. Probably the most valuable five days I've spent this year."

"Principal value of workshop was awareness of other work and at least the illusion of a "big picture" view of how these efforts all relate together, plus the beginning of practical working partnerships between groups that didn't understand each other as well."

"The workshop emphasis seemed to transform from Application Architectures to Standards Coordination. I found most of the discussions valuable."

"Valuable start to evolve something that will be even more valuable."

"Immense value in promoting awareness between groups primarily."

"Highly valuable, particularly in finding out about the other groups, and gathering contact information and documentation (or sources of documentation)."

"Extremely valuable in education about the issues/concerns of groups."

"Great information for users. I know this was a standards group oriented workshop, but I feel everyone received a form of new information. I was primarily in the management track. If this group meets again I would like to see management focus more on organizational issues pertaining to utilization of these standards, focusing on what the user's perspective is and considerations for organizational change."

"The informal meetings and contacts could very well lead to formal collaboration with our program of work."

"The workshop is a BIG success. Although I am familiar with about 50% of the groups gathered, I still gained a lot of information and insight into the huge scope of the problem. We have been able to achieve in a relatively low cost, quite a lot."

"Value -- 1. Definition of a shared vision (beginning of commitment). 2. Formation of contacts/liaison with other SDO/Cs, consortia. 3. Sharing of information - directives, what groups are working on. 4. Identification of overlap and common areas of concern."

F.2 Workshop Format

"By targeting key contributors to respective groups, the workshop allowed high bandwidth communication across groups."

"I do think the workshop approach needs to have focused sessions and limited attendance in order to provide valuable results."

"The workshop deliverables, template/road map, glossary, and prototype for coordination are very good."

"Providing an opportunity for attendees to express their concerns and desired outcomes is valuable."

"Format was good, one presentation or statement per group is adequate/necessary."

"Twenty minute introductions were of particular use and should be considered for future workshops."

"Less time devoted to groups attending, giving show-and-tell - require group info in advance."

"I thought the workshop was definitely valuable. The main thing I would have liked to see more of is an exploration of overlap/possible combination of different standards groups. Some of the discussions lingered too long on the theoretical without spending time on more solid issues. I would like to see some presentation from groups at future meetings, perhaps 5 minutes. Longer presentations from selected groups (i.e., those who weren't at this meeting) would be nice as well."

"I would still want to maintain at least some degree of plenary overview of the principal represented groups, but the breakout working groups should remain the principal core of the workshop."

"The format was excellent. I would have preferred the first 2 days to be position advocacy and problem statement than just introductions of groups. In fact, 1 day of defining terms and problem statement then 1 day of breakouts then 1 day of "considered" position advocacy and another of breakouts might have been better."

"Format works well. Another workshop should not involve background discussions - material can be distributed in advance."

"Breakouts should be somewhat more structured, with at least a bit of agenda and the selection of someone in each session to serve as facilitator of each discussion (which happened anyway in some but not all sessions). For this workshop however, the exploratory character of some breakouts served its purpose well."

"Useful and generally well done. Would be improved if breakout sessions were more focused and provided with moderators/leaders prepared to keep sessions on track."

"Break-out sessions were valuable but could have been more effective if the subjects were known ahead of time and attendees could have prepared inputs."

"The breakout themes were well-planned, and tracked by consensus (e.g. planning the next breakouts) as the workshop progressed."

"The formally organized time should be a smaller percentage of the days/nights of the meeting. Even with the Birds-Of-a-Feather sessions, it seemed too difficult to find time for follow-up smaller discussions between a few people without having to miss mainstream session activity. While there's a legitimate interest in getting a lot of work done, I think the kind of work that gets done in less-structured settings should be given more of a chance to happen."

"Shorten workshop by a day."

"Length should be cut to increase attendance - three days?"

"Making sure that attendees provide contact information and documentation should be done for future meetings. Similarly, a more rigorous requirement that attendees provide position papers and supporting documentation in advance should be adopted."

"I suggest that we number documents and maintain a document register."

"Format ---

- Generally good -- facilitation was excellent.
- Focus of breakouts was initially fuzzy so figuring out which session to attend was difficult.
- Need to continue to keep breakouts to 2-3 at one time."

Improvements ---

- Refine focus of conference and breakouts
- Shorten workshop to 3-4 days
- Hold workshop in warm locale
- Use clip or pin-on name tags for use daily."

F.3 Suggestions For Next Workshop and Next Steps

"It is important to identify the common vision of the group early in the process."

"We need to work harder on e-mail!"

"It has been proposed to have a funded group to coordinate a yearly congress. This group could also coordinate thematic conferences, e.g., messaging services, service interfaces, OBJ models, etc. "A continued forum for technical discussions" may already imply this."

"For the follow-on meeting, don't have the long plenary for beginning intros to organizations."

"Reserve more time and tracks on technical subjects."

"Next workshop: we have, out of this workshop, recommended a process(es). One of the work items at the next workshop should be "business process improvement" for the recommended process."

"Recommend that next 'Congress' have position papers in order to be invited to come. (Limited attendance so it does not turn into a zoo)."

"Focus more on technical interactions with white papers, etc. done ahead of time, so time will not be wasted by ironing out terminology."

Glenn Hollowell volunteered "to serve to "structure" a group and program committee for the next "congress" meeting in early December '93."

Jerry Winkler volunteered "to draft a model for interaction of consortia and formal standards."

"Make sure you retain Birds-Of-a-Feather ad-hoc meeting facilities for both open and closed meetings. This is often where the real work gets done."

"Must have recommendations that move ideas forward; this cannot be an end in itself."

"Next time, we should allow specific SDO/Cs to host breakout session in which that group can review their directions and status for some work item and then other groups can critique their progress.

"Very concerned about [government organizations] organizing the next workshop. Government is not noted for responsiveness to input from outside, or the conduct of successful conferences, etc."

"Here are some generic standards management problems to watch out for:

- inaccessible standards problem: it is too hard to get standards or understand their status. Why not make them available by e-mail for free?
- creeping scope problem: this occurs when a group perceives a need for a related standard and adds it monolithically to an existing large standard rather than referencing a solution.
- why not compromise--use my solution: this occurs when one group is interested in compromising with other groups only when the other group adopts the first group's solutions.
- standards-by-exhaustion problem: The current CD-DIS-IS progression produces documents that are too big, too hard to understand, too hard to change and are too complex. This favors insiders, groups with large resources, individuals with infinite patience, and can result in standards-by-exhaustion. While the fittest, most committed SDO/C developers survive, this process appears to result in suboptimal, slow standards. While an open process, the process could be improved by actively encouraging face-to-face reviews of work items in progress at academic, industrial and other SDO/C forums. This review process should be the responsibility of individuals groups, but should become part of the process.
- insular irrelevant problem: this occurs when a group attracts a small faithful flock of like minded members and grows out of touch with industry.
- the SDO that wouldn't die problem: this occurs when a group loses direction and fails to self-destruct."

F.4 Integration Issues; Concerns; Recommendations; Action Items; Next Steps; Other

"Before coming to the workshop I assumed that our problem in trying to integrate three architectures was a unique problem - I now realize that it is a common problem."

"The principal technical contribution of the workshop was to recognize the need for compositional architectures to support the selection and configuration of integration-related services. This need needs to be communicated to groups in the process of developing more-or-less monolithic line-ups of independent or layered services, including OMG, X3H4/H6, Unix International, etc."

"The greatest value of the workshop for me was in describing an object model/object services integration perspective of the next generation of standards, if we can manage their coordination. What helped me was the concept of horizontal, supporting domains vs. vertical, user domains representing significant industry areas."

"I am concerned that there was too strong a push from one participating group to adopt its existing standards as being the solution to this larger problem. It is important to have all the participants listen and modify their positions rather than to present them and "dig-in"." *[ed: no group named in comment.]*

"Most of my concerns are related to how the heads of all the groups will react to the output of this workshop. There seemed to be some consensus among the attendees, but that consensus needs to be transferred to the rest of the groups (i.e., groups that did not attend and people who did not attend). Making sure the deliverables are finished and distributed in a timely manner is critical."

"This workshop was very worthwhile. I hope we are able to carry the enthusiasm back to our individual groups and garner continued support for coordinating integration standards efforts."

"Guidelines on 'domain-specific' integration are needed"

"What are integration strategies for migration? Focus was pretty heavy on OO technology, we need to deal with services that deal with integration of other persistent storage technologies with OO. It seems things like IRDS and CDIF are trying to deal with these kinds of issues but it really wasn't talked about much. We should be building flexible and adaptable architectures. There is still a lot of expensive "glueware" being built."

"One of the packets of information not requested was a high level organization diagram included with the template."

"I expect to be able to use the road map to narrow time options for data representation, architecture, interface for our program."

Appendix G. Participant Position Papers

G.1 The Road-blocks to achieving Integrated Management of Networked Information Systems

Position Paper submitted to the Workshop on Application Integration Architecture Workshop, Dallas, Texas, February 1993.

Bruce Murrill
Network Management Forum
40 Morristown Road
Bernardsville
NJ 07924

Investment in information technology offers tremendous opportunities for organizations to reduce operational costs and to speed up their response to changes in market conditions. Unfortunately, management of networked information systems has become a costly nightmare for many corporations, government agencies, and public network operators. The growth in complexity of the networks themselves, spurred by the liberalization of the telecommunications industry in many countries and the trend toward decentralized computing, has caused a corresponding growth in the complexity of managing those networks.

The goal of all operators of information networks is to drive down costs while improving the service levels delivered to both internal and external clients. But at a time when corporations should be relying on information exchange to stay competitive, the lack of integrated management tools represents a double cost. First is the cost of managing the network itself - a people-intensive, error-prone undertaking that cannot be easily automated. Second, and perhaps more significant, the lack of standard methods for integrating management capabilities prevents companies from using the information network to the fullest extent to solve business problems and respond to business opportunities.

No single management solution represents the complete answer to the complex management problems of today: a single technology cannot be expected to deliver a total solution.

However, it is possible to embark on a sensible path toward the solution, using a mix of technologies that recognizes both the installed base of management systems and the emergence of common management platforms. It is possible to employ object oriented technologies in combination that address the needs of local sites as well as centralized data processing locations, and that work consistently between those parts of the network managed internally and those parts managed by external service providers.

This path, this combination of technologies, is called OMNIPoint.

OMNIPoint stands for Open Management Interoperability Point. Driven by user-specified requirements and defined through the collective efforts of every organization whose work touches the management area, OMNIPoint 1 is a point along a well-articulated and realistic path toward integrated, automated management of networked information systems.

OMNIPoint 1 is a set of standards, implementation specifications, testing methods and tools, and object libraries that make possible the development of interoperable management systems and applications.

OMNIPoint 1 defines a complete infrastructure that, when implemented, enables management systems to interoperate and exchange information in a common way. While building on the stability of international standards, it goes well beyond them to specify exactly what suppliers must implement in order to satisfy a specific user need. OMNIPoint 1 gives suppliers the information they need to create off-the-shelf technology or to employ such technology in the development of a management system.

OMNIPoint recognizes that many object oriented approaches to managing networked information systems have been, and will be used. It therefore begins to build an integration framework for users, and management applications developers, of such systems. While the focus of OMNIPoint is based on the OSI approach to object modelling, it recognizes that interoperability with SNMP and CORBA like paradigms will be essential to achieve end to end service management of networked environments.

Considerable work has thus already been done within the NMF, and with the OMNIPoint Partners, to compare the object paradigms and to start the specification and development of tools and algorithms to map between them. In particular we have a draft document which compares the OSI and OMG object models, identifying similarities and gaps. In the immediate future, work is scoped, planned and resourced to specify:-

- mechanisms to permit the interworking of OSI based systems with others based on specifications such as OSF/DME and OMG CORBA, and
- notation translation algorithms and tools needed to be provided to translate between the various notation techniques such as GDMO and IDL/I4DL.

We believe that significant numbers of management applications will be developed using object oriented development tools and products such as software development systems and OO databases. The requirement to be able to use the above tools, in an OO applications environment, while mapping into the OSI interoperability paradigm is high.

In addition to the above, because of the need to import device based management information in order to gain an end to end perspective, NMF, in conjunction with Internet participants, has been active in mapping between OSI and SNMP. Draft deliverables (currently lodged as draft Internet RFC's) have been produced describing the following:

- Internet SMI MIBII in GDMO format
- Internet SMI Party MIB in GDMO format
- Mapping algorithm from SMI to GDMO
- Proxy agent specification

More details of the above will be available at the meeting.

NMF recognizes that an object and object model explosion could seriously compromise the credibility and widespread up take of object oriented technology. It is prepared to work with others to ensure that solutions are found that allow widespread interoperability of management information in order that the service needs of users of networked information systems can be managed effectively.

The OMNIPoint program is an ongoing industry wide activity supported by NMF, OMG, X/Open, OSF, UI, COS, SPAG, the Open Systems Testing Consortium (both Europe), NIST, UK CCTA (GOSIP), European Commission project CTS3/NM, INTAP and TTC (both Japan), OSINET and the three regional OSE Workshops (OIW, EWOS and AOW). In addition support from, and a number of specifications of T1M1, TTC (Japan), ISO and CCITT have been incorporated in the recently delivered OMNIPoint 1.

G.2 The Changing Role of IRDS in Application Integration

Position Statement of X3H4 Members Attending The Application Integration Architectures Workshop

The IRDS Move Toward an Object-Oriented Perspective

Information Resource Dictionary System (IRDS) standards have historically been associated with describing the information resources of an enterprise. In the partitioning of integration architectures into data, control and presentation aspects, dictionary systems have been concerned primarily with problems of data integration. The information to be managed, however, consists of active assets that interact with the business processes of the enterprise. The IRDS standards are now moving toward management of objects that represent or contain these active information assets. In this dynamic environment, data and the operations that affect the data should not be separated. Extensible, encapsulating operations and inheritance of both properties and operations blur the distinction between a passive repository that integrates data and an active framework that integrates processes.

Much of the knowledge of an enterprise is embodied in the processes used to carry out the business functions. Competitiveness in today's markets is driven both by collected information (intellectual property) and the flexibility and effectiveness of the enterprise processes. Shortening the cycle time of business processes is often the key to profitability. Enterprise integration depends on the coordinated management of both information and processes. Object information management provides a working paradigm that can unify these two perspectives.

A Services Architecture for Integration of Components

An architecture is a unified and coherent structure that specifies components and the interrelationships of those components that establish how they fit and work together. A services architecture deals with the specification and organization of services provided by heterogeneous executable components. Components in a services architecture can be added or removed from a system, either statically as part of a system configuration or dynamically after a system is running. The architecture establishes a structure to integrate these components or service "building blocks" into a single coherent topology. Services can then be provided as a unified structure with a common external interface. The IRDS Services Architecture defines a structure for the incorporation of diverse services including those for defining and managing information content. It also defines a common services interface to supply those services to using applications.

Analysis of representative standardization efforts points to similar definitions of services with at least some duplication of effort. This overlap in base services is usually not related to differences in subject areas or domains. With the transition of existing efforts toward object orientation, tightly coupled collections of services can be partitioned into components that can be reused or combined in flexible ways. Other standards sources (e.g., OMG) are beginning with a partitioned set of services intended to be used as components. Analysis of representative services indicates that the same services, after factoring, are usually more similar than different across defining sources.

Separate services that share a common object management foundation can supply the components of an extensible services toolkit. With inheritance and specialization, the available service components can be used and combined in construction of specialized facilities and tools. To allow services to be used as generic building blocks, we must agree on a core object model that allows them to inherit and reuse capabilities from each other. The work of specifying the services can then be allocated to the most appropriate group with the hope of getting a result that will be usable by others. Partitioning the

specification of services among contributing groups should hasten progress and reduce redundancy and conflicts.

Partitioning the specification of standard services can also lead to partitioning the implementation of those services. Ultimately, a services architecture should provide users with flexibility for configuring available services that leads to cost effective solutions. Specializing and configuring the common base services can allow those solutions to be tailored to specific content areas or enterprise needs. These configurations of services can themselves be standardized as service profiles that allow interoperability across heterogeneous implementations.

IRDS Content Definition

Uniform Representation of Schema Levels

A services interface provides the basic mechanisms to enter, manage, manipulate, and retrieve the contents of a repository, database, or integration framework. These services may be used to manage contents of many different kinds. For an IRDS, these contents typically describe the information assets of an enterprise. To describe such information assets, the enterprise itself must also be described to some level of detail.

An important role for a repository is to provide a consistent form of definition for all its contents. The IRDS Framework (IS 10027), a guiding document for IRDS standards, establishes a multi-level schema framework to contain these definitions. This schema framework is based on a principle referred to as "level pairs." These level pairs require that IRDS content at one level always be linked to controlling schema definitions at a higher level. IS 10027 defines a progression of levels from enterprise-specific data, to controlling schema definitions (some of which can be standardized as "content modules") to meta schema definitions and finally to a schema that defines the meta schema.

The current international IRDS standard (IS 10728 - IRDS Services Interface) defines access services on the bottom two content levels. The top two levels are frozen by the current version of the standard. Virtually all the specified services are level-independent: they operate identically on either the base data level or schema level that controls it.

Explicit meta levels, which can be accessed and processed by a full array of services, is an important principle for any repository or integration framework. To avoid additional complexity, it is also desirable that the representation and services of a meta level be as uniform as possible with the representation and services of the base level. Due in part to its foundation on an SQL data model, the IS 10728 IRDS standard satisfies both these goals. SQL databases have long represented schema data in the same kinds of tables as they use to store base data.

An object-oriented integration framework should provide explicit meta definitions using the same basic kinds of objects, under the same object model, as the objects they define. Many existing object systems (e.g., Smalltalk, CLOS, OMG/CORBA) already provide such explicit meta objects, but some do not (e.g., C++).

Current IRDS Content Definition

The current IRDS standard (IS 10728) defines a data management system that provides only a fixed set of predefined services on its content. These include operations for basic create/retrieve/update/delete (CRUD) functions, and also additional support for versioning and naming services.

The IRDS Services Interface Standard uses SQL2 as the means to define its basic data model. The SQL2 language is used throughout the standard to define the data structures that hold schema definitions, and

many of the operations in the services interface implicitly or explicitly populate tables that hold IRDS content. The use of SQL throughout the standard, however, is somewhat misleading. The title of the IS 10728 standard is "IRDS Services Interface," and this standard ultimately defines only a set of executable operations that manipulate field level data as passed by input or output parameters in procedure calls. The SQL language is used only as a formalism to define an IRDS data model, and not directly as either a Data Definition Language (DDL) or Data Manipulation Language (DML) for IRDS content.

By defining an IRDS system entirely by means of operations that manipulate its content, the IRDS standard is already object-oriented in one basic respect: it encapsulates all its stored state behind the behavioral abstraction defined by its executable operations. These operations, however, are fixed by the standard and no mechanism for defining new operations is provided. A variety of language bindings, including C and Ada, are now being prepared for eventual inclusion as annexes of the existing standard.

Object-Based Content Definition

To move IRDS toward further object-oriented capability, the IRDS Rapporteur Group (the IRDS standards group under ISO/IEC JTC1/SC21/WG3) recently accepted a U.S. proposal to add extensible operations to a future version of the IRDS Services Interface standard. The U.S. proposal defines a series of additional IRDS data structures which can describe user-defined operations against IRDS content. External methods can be specified which provide the implementation of the extensible operations.

With the acceptance of the Extensible Operations proposal, the IRDS standard is now moving quickly through its transition from a fixed-function data manager to an extensible object manager that can support a variety of external object implementations. The U.S. IRDS standards strategy is to continue building on the existing international IRDS standards by preparing a series of change proposals that add and refine new capabilities. These include object-level versioning and multiple inheritance. Because of the use of SQL within the current standard, the U.S. is also closely monitoring the work on SQL3 that adds object-oriented structure to the SQL family of database languages.

To support a comprehensive object-based services architecture, more recent IRDS work suggests that the underlying object model eventually needs to contain a complete formalization of object services and the operations that characterize them. This formalization would be accomplished by adding additional schema objects to those that define operation interfaces. These objects would further constrain the valid ways to request operations, and would declare the observable results of operations using formal software specification techniques. They would specify the operations belonging to a service in an implementation-independent way, and would allow services to be specified using a small and consistent set of basic primitives.

An effort to pursue such a Service Definition Formalism, and to represent the formalism by a set of explicit meta objects, should be conducted through close liaison and/or joint effort of a number of different standards groups having skills to contribute and an interest in the result. It is also important that such a standards effort tap into the academic and research communities to obtain guidance on the needed techniques and to help validate the possible solutions.

Logic-Based Content Definition

The current IRDS standard formalizes IRDS content using a data model and a fixed set of access operations. The object-oriented extensions now in progress would also allow the content to be formalized by a set of extended domain-specific operations. Neither of these levels of formalization, however, gives an adequate basis for defining the meaning of the information that is stored by the data model or manipulated by operations.

An important role for an integration architecture is to integrate information from multiple sources so that it can be used as a unified whole. Data representation standards (such as the work of X3L8 and SC14) are

being developed to specify the meaning of shared information. However, when information has been represented in different ways, and has been collected using different rules or assumptions about what it means, its integration can be accomplished only by supplying a formal semantics of the information of interest. A formal semantics specifies how a represented form of information can be interpreted as logical statements about objects belonging to a domain of interest. Rules for translating languages to and from this abstract interpretation can also be specified.

To establish a basis for integrating IRDS content, the U.S. recently completed a technical report on the IRDS Conceptual Schema. This report follows in the tradition of the ISO Technical Report TR9007 ("Concepts and Terminology for the Conceptual Schema"), which established the basic concepts for interpreting information as statements about a logical universe of discourse. To this logic-based foundation, the U.S. added a classification structure to hold a variety of primitive and defined concepts. Support of multiple languages to express these concepts and their occurrences was also proposed.

The U.S. report recommended that one or more "normative languages" be defined that would have the ability to express any statement of formal logic. An initial normative language based on a visual form of logic called conceptual graphs was proposed. Alternative normative languages could include the Knowledge Interchange Format (KIF) developed as part of the DARPA-sponsored knowledge sharing effort. The task group working on the IRDS Conceptual Schema has met repeatedly with representatives of the knowledge sharing effort, and is currently working to make sure that the underlying semantics of KIF will be consistent with any normative language that IRDS might adopt. The work of the PDES Dictionary Methodology Committee on a Semantic Unification Meta Model (SUMM) has also been incorporated in this work.

The U.S. work on the IRDS Conceptual Schema has also been contributed to a new ISO/IEC JTC1/SC21 special working group on the conceptual schema and data modeling facilities. This special working group is planning how to reestablish active conceptual schema work at an international level. Such work needs to be positioned where it is generic to any particular technology such as communications or data storage, and also generic to any particular application domain such as CIM, CASE, CAD, etc. One of the applications of the logic foundation should be to describe the elements of a services architecture. Core components of a services architecture should be also be generic to any particular technology or application domain.

X3H4 Historical Perspective - Lessons Learned

Standards Window of Utility

X3H4 missed two windows of opportunity with regard to the development of the first Information Resource Dictionary System (IRDS) standard, i.e., X3.138-1988. In the first instance, X3H4 failed to see the value in the work by the British Computer Society on Dictionary Systems. This work was presented to X3H4 early in the development of what was to become X3.138, and members of the development group visited X3H4 to promote their ideas, but those ideas fell on polite (NIH) ears. This lack of interest in harmonizing ideas between the two countries was the first lost opportunity.

The IRDS standard was ready for national standardization in 1983-84, but the committee decided to contribute this work in 1985 to the international community with the hope of obtaining approval as an international standard. The proposal was soon bogged down in the international standards process and it was 1987 before the document was extricated to resume the process of national standardization. By this time, the IRDS was considered an international topic. The public review process pointedly demonstrated this shift in perspective. The standard was finally approved as a U.S. standard in 1988, but very few implementations have occurred. The delay produced a standard which was of a form no longer needed (i.e., command language) and which no longer satisfied the requirements effectively.

Not having learned our lesson, X3H4 produced a services interface that was contributed to the international community in 1986. This proposal was also developed solely in the U.S. and then brought to the table; it was not an international collective idea; it was a U.S. idea. The U.S. entity-relationship (E-R) based proposal was dropped by the ISO IRDS Rapporteur Group in 1987 when the international community converted the document to a relational model basis. X3H4 decided in 1988 to progress its E-R based services interface. An alternative to this services interface (still E-R based) was accepted in 1989 by X3H4, and the U.S. then proceeded with the standardization process. This latest U.S. service interface became X3.185-1992. There are no implementations of this standard, and few are expected. The ISO IRDS services interface is now an ISO standard, but few implementations are expected of it either. Both of these standards are 5-6 years too late.

In these two cases we learned that our choices are:

(1) work together (i.e., harmonize ideas across committees and across international boundaries) to satisfy (to some degree) the needs of all the participants, or

(2) work independently to (possibly) develop a standard more quickly and without the grief that sometimes accompanies harmonization, but with the risk that the resulting standard may not work well with other standards.

We believe that the added effort needed to harmonize ideas throughout the standards development process will yield a standard that is useful and very likely to be implemented. Conversely, market economics will tend to discourage products that meet some isolated standard and conflict with others.

There is an open window of opportunity for object-oriented standards, and the need for standards based on the technology is becoming critical. Time is limited for OO based standards because the pace of technology evolution is rapid, and our IRDS lessons make it imperative that we begin the process of harmonization before that window closes.

There is a growing international recognition of the potential of object-oriented technology and a vast international market for products based on these concepts. Today, the standardization process, for all practical purposes, is only meaningful with an international focus. Standalone American National Standards efforts will have only limited success since standards must meet international market needs as well as national needs.

A Direction toward Harmonized Standards

Reduce Competition in Standards Development

Separate standards that address overlapping functionality compete for the limited resources available from supporting organizations, confuse potential users, slow progress, and in the end, cost both the consumers and the producers (i.e., implementors) of the standards. This is particularly true where competing specifications lead to inconsistent interpretations. Competition between standards development organizations does not necessarily promote the best technical solutions, but it does create barriers to communication and prevent use of the best solutions based on their merit.

The bottom-line is that there is too much that needs to be standardized, and there are pockets of expertise best suited to focus on the development of particular capabilities needed for standards. With limited resources, the most effective approach is for us to work together to create more rapidly harmonized standards that can serve as the foundation of integrated environments supporting customer needs more effectively.

Integrate Consortia and Standards Body Contributions

Workshop on Application Integration Architectures

Standards development organizations and consortia have the same general objectives. Standards organizations generally represent a broader community, and their work is performed by volunteers following a very open development and review policy. Consortia provide dedicated resources and perform work in a relatively closed environment. Consortia also provide services, such as validation, that standards organizations generally do not provide. Both operating paradigms are needed to yield workable and lasting solutions. New modes of cooperative work between standards groups and consortia are needed to assure the best products are produced in a timely manner.

Finally, standards must be defined in an international context to support global commerce. National standards bodies and consortia both need to position their contributions in an international scope.

The Application Integration Architectures Workshop is a first step that could lead to working relationships between the key groups that are driving integration standards using object oriented concepts. Minimally, the participants in the workshop should leave with a better understanding of the state of the work in a broad array of groups. At best, this workshop will initiate cooperative efforts that will speed the standardization process and lead to a useful suite of standards that can be implemented to solve our growing problems with managing information.

Roger Burkhart
Bob Hodges
Jerry Winkler

G.3 X3H6 Position Paper for AIA - February 8, 1993

X3H6 would like to see AIA address the issues surrounding tool integration in a break-out session. In particular, a discussion of the overlap between control and data integration could be very fruitful both in terms of bringing forward possible new solutions, and also in learning what other groups are also aware of and perhaps addressing these problems.

The following text provides a scope for the proposed discussion.

Tools form the substitutable elements of the environment and provide the functional and human interfacing capabilities of the environment. Tools are integrated in several dimensions including the following:

- Control integration, i.e. they may share the flow of execution on one or more processors, and they may invoke one another by means of messages, using a common format and semantics for the data either explicitly or implicitly contained in those messages.
- Data integration, i.e. they may share the data in one or more repositories (including file systems and data streams), or interchange data in messages, but in all cases using a common format and semantics for that data.
- Presentation integration, i.e. they may share the use of human-interactor elements such as display space, keyboards, pointing devices, etc.
- Process integration, i.e. they may share the use of other tools for services in other domains such as process-control, copyright-control, etc.

CTIM's primary focus at this time is on the CASE Tool Integration messages. The committee has done some work to outline the scope of the Tool Infrastructure task and the SD-3 is in letter ballot. It appears that the CASE messages team will be levying requirements on the infrastructure group. That group will in turn develop standards to satisfy those requirements or levy requirements on underlying mechanisms. At this point it seems that although the CASE messages and infrastructure standards will evolve separately, operationally, they will be interdependent.

Separation of integration issues into dimensions is somewhat artificial, because the commonly-used dimensions are certainly not orthogonal. For example, purely control integration cannot be achieved without integrating data models and schemas at least to the extent of the concepts that are explicitly or implicitly contained in control messages. Presentation integration similarly overlaps partially with data integration, as a unified "look-and-feel" requires commonalty of concepts so that they can be presented similarly, even more so if a single user interface or presentation manager is shared by multiple tools.

G.4 The Need for Object Model Interoperability

Position Paper submitted to the Workshop on Application Integration Architectures

Dallas, Texas, February 1993

Frank Manola
GTE Laboratories Incorporated
40 Sylvan Road
Waltham, MA 02254

Future information processing environments will consist of a vast network of heterogeneous, autonomous, and distribute} (HAD) computing resources (hardware, programs, and data). There is increasing agreement that modeling such a system as a distributed collection of objects provides the appropriate framework for integrating HAD resources, in both distributed computing and telecommunications environments. This is illustrated by the number of standards activities related to HAD systems, including communications, distributed systems, database, and programming language standards, that are moving toward adopting, or have already adopted, an object-oriented approach [Fong et. al., 1991]. These activities include not only those of official standards bodies, such as ANSI, but also those of industry consortia, such as the Object Management Group (OMG). The use of an object-oriented approach in integrating heterogeneous and autonomous components is also a characteristic of recent developments in personal computer application integration software.

A number of proposals for generic Distributed Object Management System (DOMS) architectures have been developed, for example, the Common Object Request Broker Architecture (CORBA) defined by the OMG [OMG, 1991]. A typical feature of such architectures is the use of a common object model to provide a shared set of abstractions understood and supported by all components. This is similar to the use of a common global data model in a heterogeneous DBMS, except that in a DOMS an object model is required to model the diverse behavior provided by the objects in the system. General acceptance (or standardization) of the object model and interfaces of such an architecture would facilitate independent development of interoperable objects and supporting software in a DOMS environment (which is, e.g., the goal of the OMG in specifying and promoting such an architecture).

The number of architectures and their associated object models, together with the object models associated with programming languages in which individual objects (or object types or classes) would be implemented within such architectures, indicates that the issue of interoperation between objects in all these models should be investigated.

For this reason, one of the work items of the X3H7 technical committee (Object Information Management) is the investigation of object model interoperability, and possibly the development of an interoperable object model. This work item carries forward work begun by the ANSI OODB Task Group [Fong et. al., 1991]. The work item currently involves the compilation of information about various object models with respect to a common set of object model "features", in the form of a matrix. The data collected in this process will be the basis of further analysis to determine, for example, specific features of object models that create interoperability problems, and how these problems might be addressed.

The most straightforward approach to achieving the desired interoperability would be the universal adoption of a single architecture, and its common object model. This is the approach that has been assumed in the development of a number of application integration architectures and object models. However, it is also possible (even likely!) that no one model will achieve universal adoption. In this case, the next best situation would be for there to be agreed-upon mappings between the features of the most widely-used object models found in a distributed system.

It seems to me that both these possibilities need to be fully explored. Specifically, there is a need to directly address the technology of object model interoperability, rather than yet again assuming that everyone will adopt a single object model. In addition, the development of this technology needs to take into account the rapid development of object technology. The most popular object models in use today are based on what might be termed "first generation" object technology. Any technology for interoperation should take into account features of newer object models such as [Chambers, 1992; Kifer and Lausen, 1989; Richardson and Schwartz, 1991].

At GTE Laboratories, we are currently exploring some of the issues involved in object model interoperation through exploring the idea of what might be termed a "reduced instruction set" or "RISC" object model [Manola and Heiler, 1992], based on work on the formal foundations of object models, e.g., [Beeri, 1990; Danforth and Tomlinson, 1988; Agrawal, 1991], and related sources. Such a model would consist of a few basic, but in combination very powerful, facilities to allow features of existing object models to be defined as combinations of the basic facilities, possibly using Meta-Object Protocol techniques [Kiczales, des Rivieres, and Bobrow, 1991]. The intent is to provide a common framework for understanding heterogeneous object models, by allowing their semantics to be defined in terms of a single set of concepts. This framework could then be the basis for understanding differences among object models, for defining mappings between different object models, or even for defining new, application-specific models.

We believe that much of the technology required for a RISC object model already exists in various forms, but needs to be pulled together in a clean way. However, the idea of a RISC object model is but one possible approach to investigating object model interoperability. We do not insist that our approach is the best way to address the issues; what is important is that the issues BE addressed.

References

Agrawal R. (ed.) (1991) Data Engineering, 14(2), Special Issue on Foundations of Object-Oriented Database Systems, IEEE Computer Society.

Beeri C. (1990) "A Formal Approach to Object Oriented Databases", Data & Knowledge Engineering, 5: 353-382.

Chambers C. (1992) "Object-Oriented Multi-Methods in Cecil", Proc. ECOOP '92.

Danforth S. and C. Tomlinson. (1988) "Type Theories and Object-Oriented Programming", ACM Computing Surveys, 20(1): 29-72.

Fong E., W. Kent, K. Moore, and C. Thompson (eds.) (1991) "X3 / SPARC / DBSSG / OODBTG Final Report", 17 September 1991.

Kiczales G., J. des Rivieres, and D. G. Bobrow. (1991) The Art of the Metaobject Protocol, MIT Press.

Kifer M. and G. Lausen. (1989) "F-Logic: A Higher-Order Language for Reasoning about Object, Inheritance, and Scheme", Proc. 1989 ACM SIGMOD Intl. Conf. on Management of Data, pages 134-146.

Manola F. and S. Heiler. (1992) "An Approach to Interoperable Object Models", to appear in M. T. Ozsu, U. Dayal, and P. Valduriez (eds)., Distributed Object Management, Morgan Kaufmann, 1992.

OMG. (1991) "The Common Object Request Broker: Architecture and Specification", OMG Document Number 91.12.1, Rev. 1.1, Object Management Group.

Richardson J. and P. Schwartz. (1991) "Aspects: Extending Objects to Support Multiple, Independent Roles", Proc. 1991 ACM SIGMOD Intl. Conf. on Management of Data, pages 298-307.

G.5 PCTE Contribution

From: H.F.Davis@win0109.wins.icl.co.uk

Subject: PCTE contribution

I regret that I shall not be able to attend the AIA workshop after all. Here is some status information about ECMA PCTE and some of the questions that would have been pertinent for me at the workshop. They are, in the spirit of the workshop, personal views.

1. Status of ECMA PCTE

1st edition ECMA standards (Abstract Specification, C and Ada bindings) were published in 1990-91 and 2nd editions are being prepared to take account of many comments. The standards and the comments are all available by anonymous FTP at ftp.sda.com. They should be published in June and submitted to ISO for fast-track processing. Thus we hope to have ISO PCTE standards about in the first half of 1994. There is also a complete C++ draft available imminently via FTP.

The 2nd editions and C++ binding have been the top priority for TC33 throughout 1992 but in January 1992 (sic) TC33 recognized the following high priority work areas:

- conformance test specifications
- standard schemas,
- data exchange,
- fine grained data,
- OMG technology (I think this really means CORBA)

TC33 has been actively working on data exchange and standard schemas with EIA/CDIF and now ISO/IEC JTC1/SC7/WG11 (DDSE). Other items are perhaps more intimately dependent on the base PCTE standards. The need for OO support had only slightly less support than the items listed and was also acknowledged to some extent in support for a C++ binding, despite perceived instability in the C++ language.

Also, a 3rd edition of the NIST/ECMA Reference Model should be published jointly by NIST and ECMA in June 1993.

2. Future of PCTE

The emphasis in management and promotion of PCTE to date has been on creating a single line of development (i.e. controlling the centrifugal forces generated by evolving technology) and preparation of good standard definitions so that conforming implementations are equivalent. This, and the proposed use of fast-track processing, have created an impression that ECMA PCTE is seen as perfect and final. In fact, much of our single-mindedness arises from the need to stabilize PCTE so that it can be implemented and used and can evolve on the basis of experience.

The AIA workshop provides the opportunity for consideration of technical issues that will affect the future evolution of PCTE. There are three main questions for the PCTE community:

a) What is needed besides PCTE to provide a complete, coherent framework for SE tools and what is needed to ensure that the available components can be integrated into such frameworks? PCTE isn't a complete framework (we tried!).

b) What is needed to enable tools to work in different environments and to interoperate? PCTE isn't the only candidate for the services it provides. The work we've started on data exchange and standard schemas are just an initial "coarse-grained" stab at these questions. Discussion on X3H6-list (e-mail) has been concerned with tackling the issues with rather more sophistication.

c) How should PCTE be extended or complemented to provide more OO support? There is a strong demand for this. Can there be one solution or must more than one be provided, i.e. is there consensus or convergence in the OO community? (I like Frank Manola's premise that we should not assume everyone will adopt a single object model.)

Even if we took the conservative view that OO technology is too immature in SEEs to rush into this, answering the first two questions leads us into (c) anyway.

3. Impact of PCTE

I understand that some attending the workshop will think that PCTE cannot or should not succeed because they feel the more thoroughgoing behaviorally OO approach supersedes it. However, there are also some attending who see the possibility and virtue of combining these different approaches. The question I would have liked to ask other participants is d) If PCTE becomes part of the context (optionally), how does it affect the objectives of your activity? My belief is that in most cases it is either neutral or else it is beneficial in that it fixes one of too many variables in SEE frameworks, and thus allows more specific and practical specifications to be defined. In other cases there may be rivalry, but we shall be able to concentrate on practical solutions to the problems for tool writers arising from alternative framework components.

4. Final thought

Will the proof of success of this workshop be that there are no more AIA workshops?

Have fun!
Hugh Davis

G.6 Overview of the STEP and the STEP Standard Data Access Interface

Position paper for Workshop on Application Integration Architectures

February 8-12, 1993
K. C. Morris
kc@cme.nist.gov
Factory Automation Systems Division National Institute of Standards and Technology
Bldg. 220 / A127
Gaithersburg, MD 20899

This paper summarizes

- the work that is currently being done within ISO 10303 (a.k.a. STEP -- ISO/TC184/SC4), and
- the role of the STEP Standard Data Access Interface for sharing data.

Particular emphasis is given to the STEP Standard Data Access Interface since coordination with other standards is especially important for applications using this interface. The paper is intended to describe these topics as a basis for discussion at the Workshop on Application Integration Architectures at Texas Instruments.

1 Overview of the STEP

The International Standard for the Exchange of Product Model Data (STEP) provides a basis for communicating product information at all stages in the product life-cycle, covering all aspects of product description and manufacturing specifications. The fundamental components of the STEP are product information models and standards for sharing information corresponding to such models.

The development of STEP is supported by numerous countries, businesses including PDES, Inc. which is a consortia of industrial corporations (PDES stands for Product Data Exchange using STEP), commercial software vendors, and universities [Furlani90]. NIST is itself active in some areas of model development and in the development of the exchange mechanisms. In addition, NIST administers the National PDES Testbed. The Testbed is used in the development of information models being proposed as part of STEP, as a facility for conducting prototype implementations of systems using the STEP models and exchange mechanisms, and in the investigation of the suitability of new technologies to the application areas covered by STEP [McLean90].

Until recently the focus of the standard has been on exchanging data files for sharing product data; now there is an effort underway to define a mechanism for sharing such information more dynamically and at a finer level of granularity through the use of database management systems. Within the STEP community the different types of data sharing are referred to as levels of implementation: Level 1 refers to sharing by means of an exchange file; level 2 refers to data sharing using a standard application program interface; and level 3 refers to data sharing using a database management system as the means of data storage and access [Alte88c]. These more sophisticated mechanisms must be coordinated with other standards in order to succeed.

The underlying assumption when sharing data using STEP is that the data in question corresponds to an agreed upon integrated conceptual schema. While this is extremely useful for the exchange data in a neutral file format, sharing data directly through a database management system is much more complex. Among the things to be considered in such an environment are

- how to access the data,

- how to limit access to data in such a way that a foreign system would be able to access only that data which is desired and not other information in the system,
- how to locate data in a shared and distributed system, and
- how to integrate the STEP models for data into an enterprise's global data system.

These are considerations which have only begun to be addressed by the STEP development community and are also of concern in other standards.

The initial thrust of the STEP development effort has been to build information models to represent structure and semantics to be associated with shared data. This was and is a difficult task. It requires agreement on standard product information models, a language for representing these models, and the specification of an exchange format. The requirement to support the models of existing CAD and CAM systems has made achieving consensus on the content of the standard product information difficult because the models often overlap and conflict. For example, a curve through space can be represented as a b-spline, as a list of curve segments, or as a non-uniform-rational b-spline (NURB). The STEP modelers have undertaken the very difficult job of defining mappings between the different representations of the same information.

The need for language capable of reflecting rich semantics and data structures resulted in the specification of the conceptual modeling language EXPRESS [ISO11]. Among other things this language contains many "object-oriented" features. For instance the language provides for the representation of constraints and the representation of classes of data in both hierarchies and networks simultaneously. The format of the STEP exchange file mirrors the EXPRESS language. The STEP Standard Data Access Specification (SDAI) [ISO22-WD], which is currently under development, is based on the requirement for a means of dynamically accessing data defined using the EXPRESS language and also directly reflects the EXPRESS language.

The initial release of STEP (targeted for release in 1993 as a Draft International Standard under ISO/TC184) will consist of a group of clearly and formally defined information models (covering application areas including but not limited to geometry, presentation, and drafting), a language for specifying those information models, EXPRESS, and a protocol for representing exchange files based on these models. Subsequent releases will expand the scope of product information covered and will include SDAI. Thus, NIST through its involvement in the development of STEP has an active interest in object-oriented information models, in the manipulation by application programs of information bases using such models, and in mapping conceptual models and manipulations onto a common object-oriented service.

2 The STEP Standard Data Access Interface Overview

SDAI is a project within the Implementation Working Group of the STEP (ISO/TC184/SC4/WG7). This interface will provide a standard mechanism to permit application programs to access product data such as that found in STEP. Interfaces like SDAI have been prototyped by many researchers. The specification of an interface to STEP data for application programs is considered a high priority for standardization. ('STEP data' refers to the information models included in STEP.)

The information models of STEP are intended to disambiguate data for the purpose of data sharing across enterprises. However, until an interface is defined for accessing data, the data must be exchanged using the mechanism of file transfer. While this capability is much better than what exists today (proprietary data files or ambiguous data files), the ability to share data will be greatly enhanced if data can be accessed directly from shared databases. The need to access data directly from a database is emphasized by considering the amount of data needed to describe a product throughout its life-cycle.

SDAI is the first attempt at standardizing a runtime interface to STEP data. The primary requirement for SDAI is for a means of dynamically accessing data described by these models that are represented in the conceptual modeling language EXPRESS. SDAI should isolate the application program from the type of underlying data storage technology, which includes hardware as well as software paradigms. At the same time the interface should allow the application program to make use of specific programming language paradigms as desirable.

The SDAI specification will contain several parts: a functional definition and several specific programming language bindings. The functional definition specifies the functionality of the interface. For the functional definition it is assumed that the data to be accessed can be described using the EXPRESS language. The initial specification will include the SDAI functional definition accompanied by language bindings to C, C++, and FORTRAN.

The first draft of the SDAI specification to be distributed outside of the Implementations committee was the topic of a workshop held in October 1990. Feedback from that workshop and subsequent prototyping activities have led to many improvements in the specification. The next draft of the document to be distributed outside of the committee will be available in the spring of this year.

Several considerations which have arisen during the development of SDAI are worthy of mentioning here:

- the scope of the first version of SDAI,
- the form of the application interface, and
- support for methods.

The first version of the interface will provide for simple access to data with limited support for concurrency control and more sophisticated database features. However, it is recognized as a requirement that future versions of the interface should provide for advanced data management features such as transparent location of data, transaction management, version control, and configuration control.

The second issue addresses the format of the interface: whether the interface should be language based (in the fashion of SQL) or specified as function calls for use in an application program. The functional interface was chosen by consensus based on industrial requirements.

The final topic, support for methods associated with STEP data, is being considered for the second version of SDAI. Support for this feature may be done in conjunction with the EXPRESS Language project (ISO/TC184/SC4/WG5). At a requirements gathering workshop for version 2 of the EXPRESS language the need for the capability to represent methods in EXPRESS was identified.

If you are interested in getting involved in the Implementation Working Group of STEP here are some points of contact:

SDAI ISO Activity: Jan Van Mannen, STEP WG7 Convener,
jvm@informatics.rutherford.ac.uk

WG7 mailing list: wg7-request@cme.nist.gov
(for discussion of sdai-related topics)

EXPRESS User's Group mailing list
express-users-request@cme.nist.gov

3 Conclusion

The problem of sharing data is being approached in two ways. The first approach is being addressed by STEP and other product data standards. This approach provides common semantics for understanding of

data used in the information environment. The other approach is being addressed in several arenas which are developing software technology and related standards. The second approach provides tools and interfaces to tools for integrating the large amount of software components and data found in an information environment. While these two approaches are not incompatible and there is a distinct need for both of them, there is also a need for an overall framework within which they both can operate.

Many of the issues discussed here that are emerging in STEP may be better addressed within other standards; however, without a framework it is hard to determine what should fall within the scope of STEP, what should be outside of STEP, and (of those things that should be out of scope) whether they will be covered by other standards in a suitable manner or in a timely fashion. In summary, we are beginning to see islands of standards. Now there is a greater need than ever before for a cohesive architectural framework for tools and the associated standards.

4 References

STEP, the Standard for the Exchange of Product Model Data, is a project of the International Organization for Standardization (ISO) Technical Committee on Industrial Automation Systems (TC 184) Subcommittee on Industrial Data and Global Manufacturing Programming Languages (SC4).

Funding for NIST's involvement in STEP is provided in part by the Department of Defense's Computer-Aided Acquisition and Logistic Support Office (CALS) and the Defense Advanced Research Projects Agency (DARPA).

[Alte88c] Altemueller, J. "PDES / STEP Implementation Levels", ISO TC184/SC4/WG1 Document N282, October 1988.

[Furlani90] Furlani, C. "Status of PDES-Related Activities (Standards & Testing)", NISTIR 4432, National Institute of Standards and Technology, Gaithersburg, MD, October 1990.

[ISO1] International Organization for Standardization, "ISO 10303 Industrial Automation Systems and Integration- Product Data Representation and Exchange - Overview and Fundamental Principles", Draft International Standard, ISO TC184/SC4, 1992.

[ISO11] International Organization for Standardization, "ISO 10303 Industrial Automation Systems and Integration - Product Data Representation and Exchange - Description Methods: The EXPRESS Language Reference Manual", Draft International Standard, ISO TC184/SC4, 1992.

[ISO21] International Organization for Standardization, "ISO 10303 Industrial Automation Systems and Integration - Product Data Representation and Exchange - Clear Text Encoding of the Exchange Structure", Draft International Standard, ISO TC184/SC4, 1993.

[ISO22-WD] International Organization for Standardization, "ISO 10303 Industrial Automation Systems and Integration - Product Data Representation and Exchange - Standard Data Access Interface Specification", Working Draft, ISO TC184/SC4, 1992.

[McLean90] McLean, C.R., "National PDES Testbed Strategic Plan 1990", NISTIR 4438, National Institute of Standards and Technology, Gaithersburg, MD, October 1990.

[PDES91] PDES, Inc., Myles Barrett, George Coletta, Cliff Langenbach, Katherine C. Morris, Chris Mudgett, Augusto Nieva, "A Strategy for Implementing a PDES/STEP Data Sharing Environment", May 1991.

G.7 NCMS Rapid Response Manufacturing Program

A Summary for the Application Integration Architecture Workshop

Dallas, Texas, February 8-12, 1993

In September 1991, a Joint Venture proposal entitled "NCMS Rapid Response Manufacturing", was submitted by NCMS (acting as the joint venture coordinator) to the Department of Commerce Advanced Technology program (ATP) administered by the National Institute of Standards and Technology (NIST). The joint venture partners include TI-DSEG, Ford, General Motors, United Technologies and six leading edge suppliers of CAE/CAD/CAM systems and

technologies. This proposal was the largest of twenty-seven grants awarded in the spring of 1992 by the ATP. The program is focused on pre-competitive collaboration in the development of specific CAE/CAD/CAM applications for the support of automated concurrent engineering. Total funding for the five year program was set at $45.8 M. Industry will fund $26M and the ATP will fund $19.8M. Cooperative Research and Development Agreements are planned to add well over $1M per year to the total program effort. The program officially began on October 1, 1992.

The intent of the program is best described in the introduction to the Research and Development Agenda section of the proposal: "This program provides the effort needed to effectively enable engineers to reduce the time required to design and manufacture products in response to rapidly fluctuating market demands. The objective of the program is to shorten time-to-market, improve quality-to-cost, and enhance product reliability in order to provide the U.S. manufacturing infrastructure competitive advantage in a variety of global market sectors. Rapid Response Manufacturing will be accomplished by coordinating and extending the application of feature based solids modeling, knowledge-based systems, integrated data management, and direct manufacturing technologies in a cooperative computing environment. Progress of the program will be measured by the design and fabrication of a different family of parts at the site of each participating manufacturer".

Rapid Response Manufacturing will be accomplished by coordinating and extending the application of integrated product and process modeling. knowledge-based applications and direct manufacturing technologies. Each participating firm will measure progress of this program relative to seven key system capabilities. These seven capabilities are:

1. Establishing complete models of design and process data.
2. Improving access to product and process knowledge.
3. Accurately producing the first part.
4. Developing products in a single iteration.
5. Demonstrating portability of product models among manufacturers.
6. Creating new designs from mathematical variations of proven designs.
7. Manufacture parts directly from design models.

Each Manufacturing firm has selected a different product family for development. The processes presently used to produce these parts will be the base-line against which the progress will be measured. The program consists of research and development in four interrelated technical areas. They are integrated product and process modeling, engineering environment, knowledgebased applications, and direct manufacturing.

Product and process data will be united in a single comprehensive model so that changes in either product or process data will affect all related downstream functions. To insure interoperability, models that

represent common characteristics and processes will be developed. Product models will cover geometry, part features, tolerance information, design and manufacturing constraints, assembly information, specifications and notes, and materials information. Process models will cover process plans, operator work instructions for fabrication and assembly, numerical control tool paths and set up instructions, machine tool control, tool designs (for fixtures, jigs, and dies), and dunnage. This effort will be based in large measure on the work of ISO TC184/SC4 i.e. ISO 10303 commonly referred to as STEP (Standard for the Exchange of Product Model Data).

The engineering environment is the computer hardware and software which constitutes the information highway for this program. Hardware includes file servers, workstations, and networking lines and equipment. The environment will support data repositories containing company and factory specific information for engineering materials, standard components, design analysis characteristics, process specifications, design guides, manufacturing processing equipment, and cutting tools. The databases will be structured to support direct information access by engineers and will support access by knowledge-based application software. The environment will include data management, version control, and configuration control facilities for product models.

The knowledge-based applications will draw on the integrated product and process model data as well as the resources contained within the engineering environment. These applications include variant design, cost estimation, generative NC programming, and computer-aided process planning.

Manufacturing test beds will be located at a central site in Michigan, with remote sites at Texas Instruments, Pratt & Whitney, and Oak Ridge. These sites will be established to validate Rapid Response Manufacturing by directly manufacturing products from design software. Traditional machining equipment as well as various types of freeform fabrication machinery will be used in this effort.

Architecture development for this effort comes under the engineering environment technical thrust. The architecture must be instantiable within the life-span of the RRM program and as such must rely on the use of standards and technologies available within the next five years. The integrated product and product model technical thrust and its emphasis on STEP will have a significant impact upon the architecture.

The RRM program is in large part a program of software applications to facilitate the design and manufacture of mechanical components. The architecture will focus on the application services layer, the underlying execution services layer, and the information required to support these services. Network/Communications services and hardware architecture will be addressed but are not the focus of this program.

This architecture will draw heavily from emerging technologies and standards. There is multi-level importance to the use of standard services in the RRM architecture. However, the use of these standards must be backed with vendor support, they must not conflict or overlap with other standards, and to be of use to this program they must be stable within the RRM timeframe.

The RRM program should be considered by standards organizations to be consumers or users of standards. The opportunity is to use standards as soon as they stabilize and to provide feedback to the standards organizations. The difficulty is in discovering the overlaps and gaps between closely related efforts. In may cases the problem becomes one of how each standard is intended to be used with another. In some instances standards seem to compete with one another and the problem becomes which one to support. What often happens is that no support is given and company specific solutions are instantiated

The Application Integration Architecture Workshop will go a long way in assisting users of standards such as the RRM consortium to avoid these difficulties. This informal gathering of people associated with standards organizations, vendors, and users will do well to produce a roadmap of standards. Organizations such as RRM can serve the community well by identifying services requiring standards and through the trial use of emerging standards.

Workshop on Application Integration Architectures

Neil Christopher
Texas Instruments Inc.
6500 Chase Oaks Blvd. M/S 8408
Plano, Texas 75023
phone: 214-575-3360
fax: 214-575-6198
email: neil@ease.dseg.ti.com

G.8 Issues Discussion from ISO-IEC JTC1/SC7/WG11 (Description of Data for Software Engineering)

POSITION PAPER TO THE APPLICATION INTEGRATION ARCHITECTURES (AIA) WORKSHOP

ISSUES DISCUSSION FROM ISO-IEC JTC1/SC7/WG11 (Description of Data for Software Engineering)

February 5, 1993
Peter Eirich
Westinghouse
Convenor, JTC1/SC7 Working Group 11 of the International Standards Organization (ISO) and the International Electrotechnical Commission (IEC)

(In accordance with the workshop instructions, these are individual opinions. There has not been time for adequate circulation and discussion within WG11.)

A. INTRODUCTION

On behalf of WG11, I would like to commend the AIA Workshop participants for the effort they are undertaking to help establish consistency among the different elements of an integration architecture, and to create a basis for cooperation among the organizations working on different aspects of this problem. Several WG11 participants will be present at the workshop and I regret that I could not also be present as well. WG11 is most interested in the workshop results and would like to remain involved.

In this paper I will summarize how the work of WG11 relates to three important aspects of an integration architecture that may be discussed at the AIA workshop:

1. data integration, through the use of standardized models reflecting the "semantics" of software engineering data

2. data exchange formats for use between individual tools and/or information repositories

3. representations of a generalized object model to facilitate the exchange of software designs (particularly those based on an object-oriented approaches)

B. DATA INTEGRATION

Within an application integration architecture there is a need for tools and applications to be able to communicate reliably. Other position papers I have seen thus far have identified different kinds of integration that make such communication possible.

One of these is data integration, which I view as enabling the contents of one tool or application to be meaningfully understood by another. Within the domain of software engineering activities it is the purpose of WG11 to make such data integration possible among software engineering tools and user activities. This is to be accomplished by developing standard data models for a series of software engineering subject areas, as well as an underlying abstract model that defines the interrelationships among those subjects. The applicability of this set of models will span the entire software engineering lifecycle.

These models will effectively define the "semantics" of software engineering data. Their contents will be consistent with, and in fact will specify usable "data representations" for, other international software engineering standards.

How should this relate to the activities of the AIA Workshop? It is a "given" (in my mind) that, at any point in time, there will always be multiple means for communication among tools within software engineering environments in use around the world. What is important is this: if Tool A sends information to Tool B, then the information sent by A should be received and interpreted *consistently* by B regardless of the means of communication used. For example, it should make no difference to the end result if:

1. A batches up a lot of information into an exchange file format and sends it to B, or

2. B interrogates A using a series of "interactive" calls, such as might be defined by a standard for control integration, or

3. A exports information to a repository (or a database), and B later retrieves this information from the repository (or, B might access a different repository that had received an information transfer from the original repository), or

4. ... some other communication approach is used.

Such a consistency of communication can be achieved, and can only be achieved, when the various communication methods share common models for defining the content of software engineering information (e.g., a software design). WG11 is working to build and standardize such models for the purpose of software engineering representations.

By virtue of the number of different standards groups collaborating in the WG11 effort, a variety of different elements within future software engineering environments should be able to share this common "semantics" for software content. These collaborating efforts include:

1. ECMA TC33 (PCTE) - which will be able to use the WG11 models as the basis for PCTE Schema Definition Sets that will govern the maintenance of software engineering tool data within PCTE implementations.

2. ISO-IEC JTC1/SC21/WG3/IRDS RAPPORTEUR GROUP, and US ANSI X3H4 - which will be able to have the IRDS Content Modules for software engineering data based on the WG11 models.

3. ISO TC184/SC4 (STEP)/WG3/SOFTWARE PRODUCTS, and US ANSI IGES/PDES ORGANIZATION SOFTWARE PRODUCTS - which will be able to develop the STEP Application Protocol schemas for software products (i.e., for those familiar with STEP, the WG11 models will serve as the basis for a suite of Application Reference Models (ARMs) covering software products). These will fit within the larger product data exchange scope of STEP, and will enable the software aspects of a product description to be incorporated into the overall product description.

4. IEC TC93 (DESIGN AUTOMATION) - which is looking to WG11 for the development of common models for software that can define, in connection with STEP, the software components found within electronics products.

Indirectly, through the results produced by these different standards activities, the use of the WG11 common models will help to accomplish data integration across a variety of tools and components to be found within future software engineering environments, and across tools and applications used on a stand-alone basis within enterprises. For example, a transfer of software engineering information included as part of an overall STEP product model, using either the STEP Part 21 exchange format or the STEP SDAI protocol, should yield fully equivalent results to a transfer of that same subject matter using the either the CDIF or WG11 transfer format(s) (which are anticipated to be essentially the same). As another example, a software design stored in an IRDS, and accessed using the IRDS Services Interface, should give equivalent results to the retrieval of that same design when recorded under a PCTE implementation, and using PCTE schema access mechanisms.

In addition to facilitating the communication of software engineering data among tools and repositories, these "semantic" models will also facilitate the fundamental integration of software engineering data within (and among) enterprises.

Both the EIA CDIF and IEEE Computer Society 1175 standards have been identified as major base documents for input to the work of WG11. These two standards activities are US-based but have multi-national participation. In addition, the US counterpart activity for WG11 has been in contact with ANSI X3H6, ANSI X3H7, and with the North American PCTE Users Group, to begin understanding how WG11 activities should relate to their efforts.

C. DATA TRANSFER FORMATS

In addition to the WG11 common models, the WG11 transfer format work should have a more direct impact on application integration. The WG11 format is expected to be based on the CDIF architecture, which permits an exchange format to be derived directly from the constructs used to build the model(s) for the subject area(s) covering data to be exchanged. This architectural approach is flexible and facilitates re-use, in that the format adapts to the contents of the subject models, and does not require corresponding reserved words to be pre-defined in the transfer format itself.

In general, within an application integration architecture, there are likely to be multiple alternative ways for the same information to be communicated. Given the variety of purposes that must be served, there are bound to be different communication formats tailored for these different purposes. As described above, it is the role of the common models to help ensure that these different modes of communication can be used in a consistent manner. However, for a variety of fairly obvious reasons (e.g., economic expense for tool vendors, training expense for enterprise users, difficulty of ensuring consistency of communication, etc.) it is not a good idea to have too many of these customized and tailored formats.

During the early planning stages of WG11, a count of the number of different activities developing an exchange format for use by software tools, planning to develop a format, or wanting to adopt a format, showed that industry was clearly heading for "too much of a good thing". To help alleviate this problem, WG11 will be taking into account the requirements of the different collaborating activities listed above, the requirements of the base modeling document providers (EIA CDIF and IEEE-CS 1175), and others that are interested, in order to create a robust exchange format. In particular, the best features of at least the EIA CDIF transfer format, the IEEE-CS 1175 STL, the ANSI X3H4 Import Export 1991-195, and the ISO IRDS and SQL import/export formats, and perhaps others, will be reviewed and considered during the design process.

The WG11 objective is a format that will be generally applicable to the needs of the software engineering community, and will also serve the needs of the broader communities represented by the (future) users PCTE environment and IRDS repository implementations. EIA CDIF, which is taking the lead in this area of WG11 development, has been working with both PCTE and IRDS activities for some time.

D. OBJECT MODEL REPRESENTATIONS FOR SOFTWARE DESIGN EXCHANGES

In some of the workshop inputs I have seen, there is much discussion of the need for compatibility among different object models that are either existing or under development. There has also been discussion of the potential for one all-encompassing model of objects that is generally adopted by developers. Perhaps this will be one of the eventual outcomes from the workshop.

WG11's interest would be to reflect the contents of such a consensus (and all-encompassing) object model within its standard set of models. In this way, one (or more) particular object models could be mapped against the abstract overall WG11 model, which in turn would reflect the consensus object model. This would define how each such object model would be reflected in the transfer format.

For example, as a specific object model (or set of alternative models) becomes commonly accepted within industry, and as software designers begin employing that model(s) as a means to represent their designs, the WG11 exchange format will provide a means for different designers and implementers to exchange their object-based design information. Even users of different object models, perhaps those related to older implementations, will be able to exchange information about their designs (at least to the extent the receiver's object model is able to represent the concepts contained in the sender's model). Also, through the integration aspects of the overall WG11 set of models, portions of those object-based designs may, also, be re-used by practitioners of non-object-based (!) design methodologies. This will help increase the recognition and acceptance of object-based methods among the community of software designers and implementers.

E. CONCLUSION

WG11 subscribes to the goals of the AIA Workshop, and is interested in working cooperatively with the other participating organizations to address the larger problem of effective software development and application integration.

If you are interested in being added to the e-mail general information distribution list for WG11, please send your e-mail address information to coallier@qe.bell.ca, and ask to be added to the "SDDSE" list.

CONTACT INFORMATION:

Peter Eirich, Westinghouse
through 2/25/93:tel: 410-993-5634
fax: 410-993-5997
e-mail: eirich@mstc1.bwi.wec.com
after 2/25/93:tel: 410-765-1000, and ask operator
e-mail: peirich@mcimail.com

G.9 Issues Discussion from IEEE Computer Society Task Force on Professional Computing Tools

POSITION PAPER TO THE APPLICATION INTEGRATION ARCHITECTURES (AIA) WORKSHOP

ISSUES DISCUSSION FROM IEEE COMPUTER SOCIETY TASK FORCE ON PROFESSIONAL COMPUTING TOOLS

PROJECT 1175

February 5, 1993
Peter Eirich
Westinghouse
Member, IEEE-CS P1175

(In accordance with the workshop instructions, these are individual opinions. There has not been time for circulation and discussion within the Task Force.)

A. BACKGROUND ON THE IEEE-CS TASK FORCE ON PROFESSIONAL COMPUTING TOOLS

The IEEE Computer Society (IEEE-CS) Task Force on Professional Computing Tools has been in existence for over 5 years, and is dedicated to assisting the developers and users of professional computing tools. Among its deliverables has been a listing of over 200 existing standards applicable to the use of tools for systems and software development. Its one standards development project to-date has been IEEE std 1175, IEEE Trial Use Standard Reference Model for Computing System Tool Interconnections.

The 1175 standard provides a format (the STL language) for the exchange of design information among software engineering tools. Feedback from the usage of the current Trial Use standard will be used to update the standard, but will also provide an important input to the international standardization efforts in JTC1/SC7/WG11 (please see the WG11 position paper to the AIA workshop).

B. CONTRIBUTION TO THE AIA WORKSHOP

The 1175 standard has, embedded within it, a basic and fairly general model of an "object". This is a fairly basic model and does not incorporate many of the detailed refinements covered in the X3H7 position paper that compares object models. However, it was created based on a fairly extensive review of the object-oriented literature available during the development of the 1175 standard, and the Task Force believes that it reflects many of the essential characteristics of objects. Perhaps most important, it relates the characteristics of an object to the kinds of elements found within older, more conventional methods for software design.

Since it appears likely that there will be extensive discussion of other (better-known) object models during the workshop, I wanted to bring this one to the attention of the participants as a source of ideas for consideration. Unfortunately, no one from the Task Force was able to plan to attend the workshop and make a presentation concerning this model. Perhaps there could be some follow-up discussions after the workshop, or a presentation at a subsequent workshop.

If users of object-oriented approaches wish to experiment with a textual language for exchanging object-oriented designs, I would recommend experimenting with the use of the STL language in 1175 for this purpose. Any feedback on the strengths or shortcomings of the coverage of objects in 1175 STL would be greatly appreciated, and will also be forwarded to JTC1/SC7/WG11 to assist with the design of the object coverage within WG11's software engineering data exchange format. The Task Force would also be pleased to forward any such feedback to other groups working on an object model.

The 1175 standard may be obtained from the IEEE Service Center (1-800-678-IEEE or 1-908-562-5420) for $65.50 (or $45.85 for IEEE members). (It may be possible to arrange for a complimentary copy to be sent to a standards committee for standards development purposes only.)

CONTACT INFORMATION:

Peter Eirich, Westinghouse
through 2/25/93:tel: 410-993-5634
fax: 410-993-5997
e-mail: eirich@mstc1.bwi.wec.com
after 2/25/93:tel: 410-765-1000, and ask operator
e-mail: peirich@mcimail.com

G.10 Issues Discussion from IEC TC 93 (Design Automation)

POSITION PAPER TO THE APPLICATION INTEGRATION ARCHITECTURES (AIA) WORKSHOP

ISSUES DISCUSSION FROM IEC TC 93 (DESIGN AUTOMATION)

February 5, 1993
Peter Eirich
Westinghouse
Secretary, Technical Committee 93 (Design Automation) of the International Electrotechnical Commission (IEC)

(In accordance with the workshop instructions, these are individual opinions. There has not been time for adequate circulation and discussion within TC 93.)

A. BACKGROUND ON TC 93

Technical Committee 93 (TC 93) of the International Electrotechnical Commission (IEC) covers Design Automation standards for electrotechnical products -- those products which utilize electrical current to accomplish their function, or which influence the electrical behavior of other electrotechnical products. TC 93's interests include facilitating, through standards, the effective and efficient design and manufacture of such products, including the software component now becoming increasingly common in electronic products. TC 93's scope of work also includes environments and frameworks used for designing electrotechnical products, and supporting part/component libraries for this purpose.

With regard to software, TC 93 looks to existing software-oriented industry consortia and standards activities to provide the basic means to represent software designs. Ideally, these representations of software will be suitable for re-use within the more comprehensive product descriptions that TC93 will develop -- ones that encompass both the electronics and the software aspects of a product. In turn, TC 93's work will be formulated in such a way as to be usable within, and contribute to, the even broader product representation scope of the ISO TC184 STEP effort.

TC 93 is interested in working cooperatively in order to help ensure both compatibility and interoperability among standards applicable to design automation for electrotechnical products.

B. ISSUES FOR THIS WORKSHOP

Of particular interest is the session covering the potential differences in requirements for CASE and CAD environments. Although I do not have any in-depth knowledge of the approaches being pursued in each area, it appears to me that both the CASE and CAD/CAE environment communities have been working intensively to produce different solutions for basically the same problem. I strongly suspect that there is nothing fundamentally different between the invocation and control of a CAD tool, a CAE tool, and a CASE tool. Therefore, why should there be different specialized kinds of environment solutions for each discipline?

The direct costs to industry from having to implement and support multiple kinds of design environments are obvious. Less obvious, but perhaps greater in magnitude, are the indirect costs from trying to accomplish concurrent engineering approaches when the CASE, CAD, and CAE tools, that must reconcile their data, conform to different operating environment standards.

Unless some decidedly different requirements for these two disciplines can be identified, it stands to reason that the totality of the resources working on both CASE and CAE/CAD environment problems could make more overall progress by dividing the work to be done -- and by specializing according to their respective strengths -- rather than re-inventing with a different twist what another group may have already accomplished.

For these reasons I am most interested in learning of the conclusions reached by the breakout session. Perhaps the AIA Workshop could explore ways to get both the CASE and CAE/CAD environment communities to work together, rather than separately in parallel. Considering the savings possible in industry from not having to implement and support two different kinds of tool environments, there is much to be gained.

CONTACT INFORMATION:

Peter Eirich, Westinghouse
through 2/25/93:tel: 410-993-5634
fax: 410-993-5997
e-mail: eirich@mstc1.bwi.wec.com
after 2/25/93:tel: 410-765-1000, and ask operator
e-mail: peirich@mcimail.com

G.11 ISO Reference Model of Data Management

The ISO Reference Model on Data Management (IS 10032:1993) has no position on object models or on object services. The timing of the model's development was such that these issues began to manifest themselves when the work was almost completed.

In addition, SC21/WG3 raised a formal question in about 1989 on standards needed for object oriented databases. The response to this question is currently the subject of an ISO letter ballot. Basically, the response indicates that considerable object oriented database functionality is being included in SQL3.

One can say that SC21/WG3 has an implicit position object models and object services. It would have been quite inappropriate for the Reference Model to have addressed these questions.

On the other hand, "integration frameworks" are the main thrust of the RMDM. It defines an architectural framework the major aim of which is to position the various database standards and to enable them to be integrated successfully.

RMDM is not about the best way to model data but more about the ways to use what RMDM calls a "data modelling facility".

There are two main aspects of this framework, namely the level pair concept and the processor architecture. The former provides a basis for integrating dictionary databases and application databases.

In this connection, the principle of "level pair parallelism" deserves mention. This principle may be paraphrased as saying that functionality should not be unnecessarily different on one level pair from what it is in another.

The most obvious example of this principle is that data should not be modelled differently in a dictionary database from the way it is in an application database. (If SQL92 is good enough for application databases, then it is good enough for dictionary databases. If another approach is preferred for application data bases then it should also be used for dictionary databases. The arguments in favour are both economic (less expensive products) and pragmatic.

The other aspect of the framework is the processor architecture. The diagramming technique used is to be commended to the AIA workshop. It is very useful for developing an understanding of issues.

Finally the "means of achieving data management standardization objectives" should be mentioned. There are several of these means. As well as "level pair parallelism" based thrust towards the same "data modelling facility" for each level pair, it is also proposed that the data modelling facility used for interchanging data in different kinds of distributed database systems also be standardized.

Standards to support distributed databases are also a major concern of the RMDM and indeed of SC21/WG3. Yet another formal question is being considered on this topic. RMDM identifies the role of a "Schema for Distribution Data". This schema may well be the next major candidate for standardization.

Bill Olle

Appendix H. Service Specification Template

In section F.1, we stated that a generic architecture establishes rules for component selection and interaction and that a specific architecture is an instance of the generic architecture.

Here we provide a sample list of architectural principles (taken from OMG's Object Services Architecture specification). This list is included to provide a specific example of a collection of architectural principles.

- independence and modularity of object services
- minimize duplication of functionality
- consistency among object services
- interoperability of object services when there are dependencies
- operation sequencing should be included where applicable
- extensibility of individual object services
- extensibility of the collection of object services
- configurability
- precise specifications
- complete specifications
- object service specifications should not contain implementation descriptions

In addition we provide a sample Description Template for an Object Service (taken from the DARPA Open OODB project, part of the Persistent Object Base program). This is meant to complement the SDO/C template that lists SDO/C group activities and status. The one focuses on groups; the other (below) focuses on individual service specifications. One groups may be responsible for many such specifications. Some groups only focus on providing a specification (standard interface defined precisely) and fail to record any requirements and rationale for their choice. The template below is intended to provide slots for several kinds of specifications and to allow inclusion by reference of other similar specifications so no monolithic specification results, allowing a divide-and-conquer approach to service description specification.

SERVICE SPECIFICATION (one per service)
I. Service Reference Model (Rationale)
  A. Introduction
    1. Problem/Limitation of Current Practice
    2. Objective/Purpose/Goal/Scope
    3. Rationale for Service's Separate Existence
  B. Relevant Standards and Related Work
  C. Service-specific Glossary
  D. Service-specific Requirements
  E. Reference Model/Design Space
  F. Design Issues
    1. Intra-service Issues
    2. Inter-service Issues
    3. Inter-service Dependencies
  G. Comparison of Systems using the Reference Model
  H. Bibliography
II. Service Interface Specification (abstract and one per PL binding)

A. API (application program interface)
B. Service Test Specification
C. Examples

III. Service Implementation Specification
A. Introduction
1. Objective
2. Subsetting Rationale (if less than proper subset is implemented)
B. Intra-service Architectural Design
C. User's Manual
D. Installation Guide

Appendix 1: Source Code
Appendix 2: Executable Test Cases

Appendix I. Project Summary Repository

From: /PN=proj_sum/MBX1=proj-sum@cme.nist.gov/EMS=INTERNET/ADMD=MCI/C=US/

Subject: Access to Project Summary Repository

At the Application Integration workshop in Dallas, NIST agreed to set up a repository for project summaries. This repository has been set up on a public site and can now be accessed using the following instructions.

Use the directory **pub/stdoview** for <dname> in the instructions below.

Lisa Phillips

++++++++++++++++++++++++

Factory Automation Systems Division (FASD)

National Institute of Standards and Technology Building 220, Room A127

Gaithersburg, MD 20899

++++++++++++++++++++++++

April 16, 1993

Introduction: There are three methods of accessing the public, on-line copies of public information and source code at FASD: anonymous ftp, a Kermit server, and an Email archive server. Each of these methods is briefly described below.

anonymous ftp: **ftp ftp.cme.nist.gov (or, ftp 129.6.32.4)**

name: **anonymous**

password: **<your-user-ID>**

cd <dname> (this is where the files to be downloaded are located)

Kermit server:

1. Use a communications package that supports the Kermit protocol.
2. Dial into the NIST modem pool at **1 301 948-9720**.
3. When prompted **Enter Username>** type in your last name.
4. Connect to the system by typing in:
   connect elib.cme
5. At the **Login:** prompt, type in **kermit** .
6. Answer the prompts to register yourself as a user.

You will then be logged into the Kermit server, and will be able to access all the files available on the system.

archive server: send E-mail to **library@cme.nist.gov** (request goes in the body of message; subject line is ignored) **`help`** to get the help file for use of the archive server
`send <dname>/<filename>` to receive a particular file

STDOVIEW DIRECTORY

The stdoview directory contains general information about a variety of standards activities. The information is divided into 2 subdirectories:

summary : contains short summaries of the activities and objectives of a variety standards organizations and consortium. The summaries are in a standard format described in **summary/instruct.txt**

template: contains information on standards organizations and consortium in a template format. The instructions for completing the template are in **template/instruct.txt**.

All of the files have been contributed by a participant in the organization that they describe. They are in ASCII and follow the standard format described in the instruct.txt files.

To submit a summary or template for inclusion in this system, follow the instructions given and Email your submission to

proj-sum@cme.nist.gov

Questions or comments on any of the information provided above should be directed to:

npt-info@cme.nist.gov

Appendix J. Consortia Standards Process Model

Proposed Model for Full Participation of Industry Consortia in the Standards Development Process

Within the "*formal*" standards development process, as represented by the American National Standards Institute (ANSI) and its Accredited Standards Development Organizations (SDOs), there are many mechanisms for "coordinating" standards development projects. The effectiveness of these mechanisms is dependent on the intention of the organizations to coordinate and on the individuals who actively support the coordination. There are a number of active coordination activities ongoing in support of standards directly relevant to Computer Aided Software Engineering. The U.S. SDOs involved in this coordination are the U.S. Technical Advisory Group (TAG) to SC7, EIA/CDIF, X3H6 and X3H7.

Missing from this standards development coordination process are the industry consortia, who develop industry standards outside the "*formal*" process in order to meet market demands more rapidly. These standards, which are developed in a closed environment, may conflict with standards developed by the SDOs in an open environment through consensus. In this operating mode, both groups miss an opportunity. The consortia standards often are not recognized as standards and this is often reflected in procurements, and the SDOs do not have the input of the industry consortia.

The AIA Workshop attendees believed that this "standoff" is not desirable, and established that:

1. SDOs serve a purpose; they provide an open forum for developing a consensus standard (national and international).

2. The industry consortia serve a purpose; they develop an initial consensus more rapidly.

3. The SDOs should take advantage of the products of the consortia, whenever possible, as they should reflect a generally agreed to approach to satisfying customer needs for at least some class of customers.

The attendees recommended that:

1. The industry consortia should consider the standards that they develop to be "*interim*" standards, where the final standard would be reflected in the product of the SDOs. The "*interim*" standard is an idea available in IEEE and EIA standards.

2. The "*interim*" standard should be submitted to the appropriate SDO(s) as early as possible (e.g., even before it becomes an "*interim*" standard) as a base document or as a change proposal to evolving work to begin the process of building consensus (while getting feedback from the SDOs).

3. The consortia should represent the advocacy position for the contribution by participating in the SDO process as a member of the SDO, and should, in good faith, consider comments on the contribution as contributions themselves. **NIH** must be absent from the perspective of both groups if progress is to be made, and neither group should expect that they have **THE** answer.