

**Virtual Collaborative Environments for System of Systems Engineering and
Applications for ISAT**

David A. Dryer, Ph.D.
Assistant Professor
Department of Engineering Management
Kaufman Hall 129F
College of Engineering and Technology
Old Dominion University
Norfolk, Virginia 23529
(757) 683-5540 (phone)
(757) 683-5640 (fax)
ddryer@odu.edu

Abstract. This paper describes an system of systems or metasystems approach and models developed to help prepare engineering organizations for distributed engineering environments. These changes in engineering enterprises include competition in increasingly global environments; new partnering opportunities caused by advances in information and communication technologies, and virtual collaboration issues associated with dispersed teams. To help address challenges and needs in this environment, a framework is proposed that can be customized and adapted for NASA to assist in improved engineering activities conducted in distributed, enhanced engineering environments. The approach is designed to prepare engineers for such distributed collaborative environments by learning and applying e-engineering methods and tools to a real-world engineering development scenario. The approach consists of two phases: an e-engineering basics phase and e-engineering application phase. The e-engineering basics phase addresses skills required for e-engineering. The e-engineering application phase applies these skills in a distributed collaborative environment to system development projects.

1. Introduction

1.1. Background

The effects of globalization are dramatically changing the practice of engineering and technology in the areas of enterprise project activities and advanced engineering environments. The National Research Council's (NRC's) Committee on Advanced Engineering Environments expects further significant changes in engineering product design, project processes, as well as collaboration support, education and training within the near future (NRC, 2000). Product design and analysis is increasingly using web-based systems to assist the communication

between distributed team members. Attempts are being made to collapse project processes in terms of steps and time requirements in order for enterprises to increase engineering team efficiency. Organizations, such as the National Aeronautics & Space Administration (NASA), are conducting pilot enhanced engineering initiatives to help assess whether design and analysis teams can be distributed and more engineering activities combined or conducted in parallel to compress the resources and time required for front-end engineering efforts. Distributed collaboration systems to support such efforts are growing more and more complex, including grid-like network infrastructures connecting team members with secure high bandwidth, shareable distributed engineering data artifacts, distributed engineering tool sharing, and synchronous audio and video.

1.2. Global e-engineering environment

This resulting global enterprise environment is complex, dynamic, and produces many collaboration challenges among product development and manufacturing teams, as shown in Figure 1. Global presence means geographically distributed team members from diverse organizational and national cultures. Global organizations and collapsed project engineering cycles can create team instability as various skills are quickly applied to product design challenges. Unfamiliarity between team members is more likely due to less face-to-face interaction. Project characteristics will include reduced development cycles, greater engineering complexity, increased integration, and tighter budgets. Generating success in the new reality of global enterprises is much different than what was required in traditional engineering environments. Enterprises will need to transform and ensure product development teams thrive

in a virtual collaborative engineering environment. This environment and the teams working in it must be capable of high collaborative performance conducive to innovation within dynamic schedule, cost, and performance constraints.

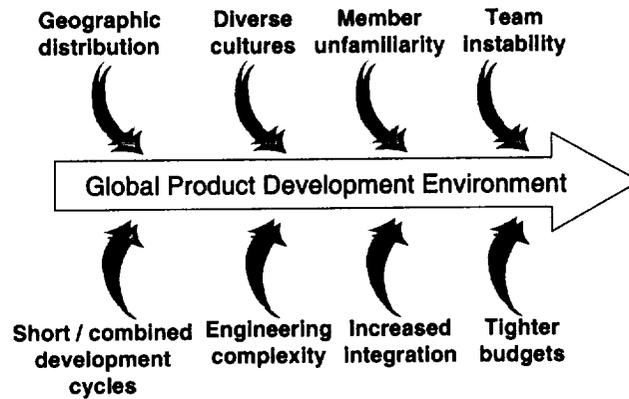


Figure 1. Global product development environment challenges.

We are calling teamwork in this environment, *e-engineering*, which is defined as ‘distributed collaboration in cyberspace using leading edge technologies enabling physically-dispersed, diverse teams to create integrated, innovative and competitive products, systems, and services.’ According to National Research Council studies (1999, 2000), an ideal virtual collaborative engineering environment or Advanced Engineering Environment (AEE) would ‘accommodate diverse user groups and facilitate their collaboration by helping to eliminate cultural barriers between groups from different parts of an organization, different organizations, or different areas of the world. There are a number of important benefits that can be achieved from effective team use of virtual collaborative engineering technologies and methods as follows (Mills, 1998):

- *Lower product development, design and production costs.* Cost is the first and foremost factor driving much of the interest in VCE technologies. Products can be developed with more interaction in less time at a reduced cost. This greater interaction and more rapid development time are accomplished through use of unique techniques and capabilities provided within a VCE environment.
- *Effective information sharing and generation.* The ability to easily share resources from remote sites is a critical component of a VCE environment. This allows all involved team members to access data, drawings, and documents to enhance design development and more quickly deal with specification changes. Such information sharing also provides an ability to evaluate the use of cutting edge technology early in the process and makes industry expert consultants more accessible. Sharing of information also enables team members to have a common understanding of all issues involved.
- *Improved communication.* The application of VCE removes geographical constraints and reduces time lost in traveling. It facilitates an enriched communication between and among participants. Team members will interactively evaluate virtual prototypes of product designs and evaluate alternative scenarios. They will be able to make decisions quicker since all team members share the same information.
- *Improved development programs.* VCEs will link physically dispersed teams for an integrated product and process development. This allows suppliers, users, and clients to provide feedback early in the engineering cycle, which enables team members to incorporate product lifecycle concerns. Such integration will also have an impact on the product quality.

In the following sections, we propose a system of systems or metasystems approach to e-engineering which can assist rapidly-organized project teams in meeting the challenges and needs of global engineering and manufacturing enterprises in virtual collaborative environments (VCEs). Applications of this approach to an ISAT case scenario are also described. Viewing enhanced distributed e-engineering environments as a metasystem provides several guiding principles from which to approach this problem. Systems of systems must be engineered in terms that provide effective design, deployment, operation, transformation, and evaluation. These new “higher order” systems must be focused on producing “systems of systems” performance as opposed to individual performance of subordinate systems. The design, deployment, operation, and transformation of higher level metasystems involve the integration multiple complex system processes to produce desirable results. These metasystems are themselves comprised of multiple autonomous embedded complex systems that can be diverse in technology, context, operation, geography, and conceptual frame. (Keating et. al., 2002). At the “metasystem” level, true optimization is a fallacy. Complex turbulent contexts and environments preclude optimization in the traditional systems engineering sense. Satisficing suggests that SoSE should focus on development of satisfactory solutions that are a continual refinement of the e-engineering environment. This perspective appreciates the continual evolution and tailoring of e-engineering system problem(s) requirements, boundaries, entities, and relationships throughout the a project’s e-engineering effort.

2. Concepts for the e-engineering project cycle

In order to help address the above challenges for global engineering, an important research question is ‘what changes are required for rapidly-organized engineering teams to quickly assimilate and execute at high e-engineering performance levels and how can these changes be quickly implemented and sustained?’ In order to provide some foundation to address this question, it is useful to first discuss selected project characteristics and virtual team concepts. This discussion will also help answer the portion of the research question concerning what critical adaptation areas are useful for project teams to perform at high e-engineering levels.

Project characteristics serve to partially define work conducted in virtual collaborative engineering environments. A project can be defined as ‘a temporary endeavor undertaken to create a (deliverable) unique product or service’ (PMI Standards Committee, 1996). Project phases are collectively known as the project life cycle, which defines a project’s beginning, phase sequencing, and end.

Many models exist for project life cycles (Dorfman, 1977). One extensively used model is the waterfall model containing sequential phases of requirements, design, build, test, and integration. Each waterfall phase should be essentially complete before the next phase begins. This model encounters problems when project requirements do not remain stable following completion of the requirements phase. One approach to enhancing the waterfall model is the prototyping model, which makes use of system prototypes with selected functionality to help determine accurate requirements. These prototypes are developed using compressed waterfall sub-models early in the requirements phase of a traditional waterfall model.

The spiral life cycle model (Boehm, 1988) is an innovation which permits combinations of conventional (e.g., waterfall) and enhanced (e.g., prototyping) to be used for various portions of

a project. Recently, the spiral model has been clarified by Boehm (2000) to capture its essence. Spiral development is a risk-driven process model generator with two main features. The first feature is a cyclic approach to incrementally grow a project's degree of definition while decreasing its degree of risk. The other main feature is the use of anchor point milestones to ensure stakeholder review and commitment during spiral cycles. Boehm has also clarified that the spiral model is not just a sequence of waterfall increments with activities following a single spiral sequence. On the contrary, the order of activities in the spiral is a guideline with the actual order of visiting or revisiting activities driven by ongoing project assessments. It is also important to emphasize that the spiral model is a process-oriented model where each cycle includes assessment and improvement of project processes as well as project deliverables.

Spiral development has a focus concerning software projects (e.g., Muench, 1994), but can be applied more generally to project life cycles, including e-engineering projects. The spiral model's emphasis on improving both project processes and deliverables fits well with the challenges of integrating e-engineering process improvements during the project life cycle. With the expected compression of project timelines and constrained budgets in advanced engineering environments, the spiral model's cyclic development approach and embedded iterative risk assessments can accelerate e-engineering team development, accurate project requirements definition and decrease project uncertainty and risk. The anchor point milestones can serve to formalize progress concerning e-engineering performance levels, as well as the approval and hand-off of external deliverables.

6. E-engineering interactions, dynamics, and technology environment

One critical aspect of e-engineering is for project teams to understand and apply the various types of distributed collaborative interactions. A general model of distributed collaboration dynamics is shown in Figure 2 (Dix et. al., 1998). A common environment is established and entities populate the environment, including project participants (e.g., team members and external stakeholders) and artifacts (e.g., documents and virtual or physical prototypes). Interactions can occur between participants and between participants and project artifacts. Direct communication interactions are conducted between participants using synchronous and asynchronous tools, including audio, video, and text messaging. Participants interact with project artifacts by controlling artifacts and receiving feedback using artifact tools. Participants also indirectly interact with each other through these artifact tools. Two forms of this indirect interaction are feedthrough and deixis. Feedthrough occurs when a participant's manipulation of shared artifact objects is viewed by others (e.g., rotation of a 3D CAD design). Deixis occurs when referencing an artifact aspect to other project participants (e.g., pointing with a cursor to a feature of the CAD drawing).

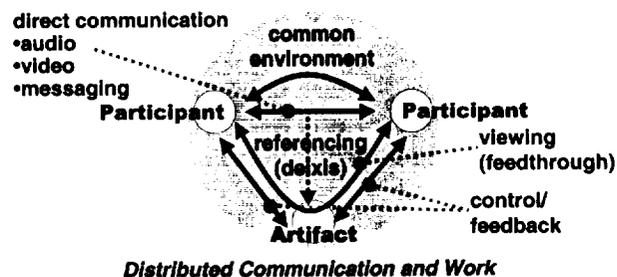


Figure 2. Distributed communication and work interactions.

This distributed collaboration interaction model can be extended and viewed more analytically in terms of an object-oriented approach to project processes and interactions.

Object-oriented extensions could be especially useful for modeling distributed collaborative processes in specific project scenarios. One object-oriented approach to business modeling treats project processes as objects, as well as other entities of the application domain (Bider & Khomyakov, 1997). In terms of the above distributed project collaboration model, objects can include entities (such as virtual team members, data artifacts, and supporting collaboration tools), as well as the interactions (such as direct communication and data artifact interactions) between such entities. In this domain, relations between conceptual objects are as important as objects themselves and entity interactions are active, not passive. Such distributed collaboration objects are complex and dynamic and the properties of objects can be represented with the help of: history, events, and activities. History is the time-ordered sequence of all the previous states of objects. The time-ordered history is most important for objects that represent collaborative processes as it shows the evolution of the process in time. Events present additional information about transitions from one state of an object to another, including date, time, impacted objects, and event attributes. Activities represent distributed collaboration actions that take place in the project domain, like the various types of collaborative model interactions described above. Such a distributed communication and work interaction model is now enhanced to represent e-engineering interactions and associated technology areas to enable this interaction, as shown in Figure 3. The two main areas are user-centric tools, representing direct participant-to-participant communication and artifact-centric tools, representing participant-to-artifact interaction. User-centric tools can take both asynchronous (e.g., email) and synchronous (e.g. electronic conferences, video connections, audio, and text messaging) direct communication forms. Artifact-centric tools can include project management and scheduling applications, product and process simulation, and discipline-specific tools needed for various project deliverable scenarios.

The same categories of groupware interactions from Figure 2 also apply and are shown in Figure 3. In order to conduct such interactions in a globally distributed environment, various technology layers are represented. A human-computer interaction (HCI) layer needs to exist between participants and the people or artifact-centric application tools. Such HCI technology includes computer input and display devices. Application interfaces are required to transfer communications and artifact data between various project applications. These interfaces involve translation protocols and data format standards. Network infrastructures need to exist to transmit interaction data to other distributed locations via local and wide area networks. A knowledge repository can also be incorporated to manage a complex project's artifacts, facilitating configuration management and quality control.

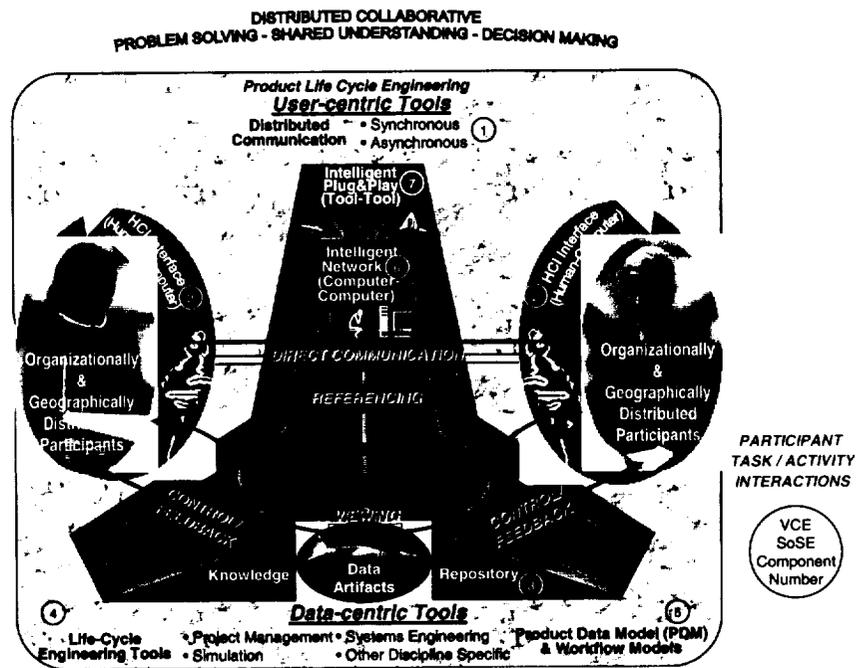


Figure 3. Virtual collaborative metasystem distributed interaction environment

It is important for e-engineers to understand the above technology areas involved in distributed collaborative work environments. Critical issues in applying this technology include

defining user requirements, tool selection, network requirements, systems requirements, and emerging technology standards. Obtaining user requirements for collaborative tool use in projects can be difficult and requires personnel with a good technical understanding of collaborative tools as well as project tasks. The team leader and members must coordinate with information technology staff in developing these requirements. One option is to develop a “strawman” list and present this list to a group of team members for validation. With a completed list of requirements, tools are identified to meet the stated requirements from the existing collaborative toolset in the organization or adding additional tools. Network requirements also need to be taken into consideration, since deploying collaborative tools can have a severe impact on a network. In order to get the distributed collaboration infrastructure working (especially between organizations), firewall security issues need to be resolved as well as the environment’s impact on network bandwidth. System requirements for the e-engineering environment can include upgrade of hardware peripherals, including headsets and desktop cameras for videoconferencing. The existing technology infrastructure needs to be leveraged as much as possible, since a majority of required e-engineering capabilities can typically be met with existing tools.

The issue of open standards is also important to understand when deploying and upgrading e-engineering infrastructures. Emerging standards for distributed collaboration include T.120 standards addressing real time data conferencing (audiographics), the H.323 standard addressing video (audiovisual) communication on local area networks, and the H.324 standard addressing video and audio communications over low bit rate connections such as modem connections. Even though these standards are being widely used, many tools still are using proprietary protocols and this can impact integration of these tools within a distributed project environment.

Virtual team and task dynamics

Another aspect of e-engineering interactions deals with team and task dynamics during projects. The task performance of teams using e-engineering technology to communicate and collaborate can be viewed as a series of stages (McGrath, 1990). These task stages are 1) *inception*, 2) *problem solving*, 3) *conflict resolution*, and 4) *execution*. Inception involves defining project goals. The problem solving stage deals with development of solutions to project technical problems. Conflict resolution occurs when different points of view and approaches need to be reconciled. Also, different cultural and organizational perspectives could require resolution. Finally, execution involves performing project tasks and overcoming project and organizational barriers that inhibit performance. These task dynamic stages are not necessarily sequential and certain stages may not be required, depending on the project scenario and complexity. A team might go from inception directly to execution for more repeatable, prescriptive project scenarios or repeat iterations between problem solving and conflict resolution with difficult scenarios. Duarte & Snyder (1999) have identified four virtual team social dynamic stages, which parallel the above task dynamics. Social stage 1, *interaction and inclusion*, is where the team identifies and maps individual skills to project needs, establishes communication and work procedures, and develops initial plans. Social stage 2, *position status and role definition*, involves member role definition and status relationships. In social stage 3, *allocation of resources and power*, the team addresses allocation of resources and member power relationships. Social stage 4, *interaction and participation*, involves performance of

collaborative work including interaction, participation among the team members, and overcoming productivity barriers.

Implementation issues

Understanding and developing proficiency in the above aspects of virtual team interactions and dynamics are essential for attaining rapid, high performance levels in e-engineering environments. Collaborative interaction and technology areas, as well as task and social virtual team stages are critical e-engineering areas to understand, establish, and continually improve during the project life cycle. Initial assessments need to be made of team member global distribution, technology capability, and skill levels in distributed collaboration as well as relevant engineering-specific disciplines. Individuals and the entire team need to be trained in identified e-engineering skill and knowledge area deficiencies. Virtual collaborative functionality needs should be mapped to project activities and technology solutions identified to enable this capability. Collaboration technology, task, and social processes should be iteratively assessed and continually improved during the project life cycle.

4. The model for e-engineering team adaptation (MeTA)

Now that a foundation of literature has been discussed and e-engineering-related concepts identified, an initial framework is proposed, called the Model for e-engineering Team Adaptation (MeTA) to help improve the performance of such global engineering teams. As part of a word's structure, *meta* can indicate change, (e.g., metachromatism – a change in an organism's color

caused by variation of physical conditions). Similarly, MeTA can be thought of as a framework of changes implemented to a project team's dynamics, required by varying the team's physical and information technology environment to virtual collaborative engineering. The model, which is pictured in Figure 4, uses an adaptation of the spiral software development approach (Boehm, 1988), to integrate e-engineering process and product development activities. In order to quickly 'spin up' to high project performance, the team conducts various e-engineering process and product-centric cycles.

MeTA is a process-oriented model where each cycle includes assessment and improvement of project processes as well as project deliverables. Similar to other spiral models, each cycle of the model goes through actions portrayed as a quadrant. MeTA uses action categories of identify, plan, execute, and assess. The model has two main phases in the spiral itself: e-engineering basics and e-engineering application. As with the general spiral development model, MeTA is not just a sequence of waterfall increments with activities following a single spiral sequence. The order of activities in the spiral is a guideline with the actual order of visiting or revisiting activities driven by ongoing project assessments. In fact, project activities or sub-activities could be happening simultaneously in multiple MeTA cycles or phases. Anchor point milestones reviews are conducted in the assessment actions quadrant concerning e-engineering performance levels and external deliverable progress and hand-offs.

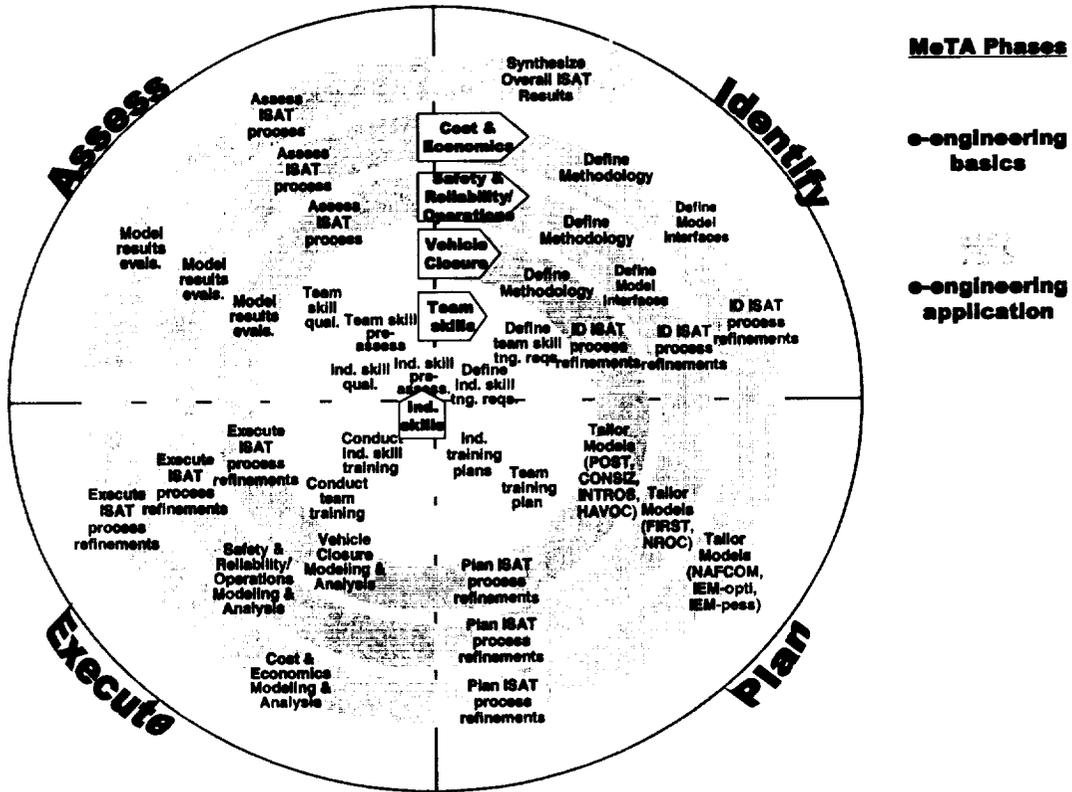


Figure 4. Model for e-engineering team adaptation (MeTA) phasing and sample ISAT Case activities.

4.1. The e-engineering basics phase

The focus of the MeTA basics phase deals with an engineering team quickly reaching proficiency in basic e- engineering process areas. In this area of the model, both individual and team e-engineering skill cycles are addressed. MeTA action quadrants of identify, plan, execute, and assess are used to bring the team to required skill proficiency. For both the individual and team skill cycles, proficiency assessments are initially conducted by the team leader or an external source.

E-engineering skill deficiency areas are identified and individual training is planned and executed to achieve proficiency in these areas. Individual training is then followed by individual qualification assessments to establish proficiency. Individual areas can include collaboration tool skills and virtual team process concepts, project management and scheduling, as well as engineering-discipline skills required for a specific project scenario.

The team skill cycle also starts with initial proficiency assessments of the team's e-engineering performance, by the team itself or by external evaluation. E-engineering team deficiency areas are identified and team training and exercises planned and executed to achieve proficiency in these areas. Team training is then followed by team qualification assessment to establish proficiency. Teaming skills include performing at proficient levels in virtual team task and social dynamics as well as working effectively using distributed synchronous and asynchronous collaboration tools.

4.2. The e-engineering application phase

In the second MeTA phase, application, the focus shifts to the team applying its e-engineering proficiency to system development or other deliverable goals. This does not mean that e-engineering process refinement activities are over, just that they are now focused on supporting project development goals. MeTA action quadrants of identify, plan, execute, and assess are now used to support iterative project deliverable development cycles. In Figure 4, the e-engineering application area is tailored to the ISAT case scenario, with vehicle closure, safety & reliability, operations, and cost & economics modeling and analysis cycles.

It should be stressed that MeTA cycles and activities should be tailored to specific e-engineering project scenarios, but there are certain MeTA 'invariants' that define the essence of this model. These invariants use Boehm's spiral software development model invariants (Boehm, 2000) as a start point. The first invariant is that MeTA is a process-driven model, concerned with a team's e-engineering process improvement as well as deliverable task progress. As such, MeTA must contain phase areas directed at e-engineering process proficiency (e.g., basics phase) as well as deliverable progress (application phase). The second invariant is that MeTA is a risk-driven assessment model where iterative process and deliverable assessments determine the type and level of effort of upcoming activities. The sequence of spiral activities is just a guide. In reality, project activities or sub-activities in multiple cycles could happen simultaneously in multiple MeTA cycles or phases, depending on these project assessments. The third invariant is that MeTA contain anchor point milestones formalize progress concerning e-engineering performance levels, as well as the approval and hand-off of external deliverables

6. e-Engineering applications to an ISAT case scenario

NASA's Inter-center Systems Analysis Team (ISAT) is conducting a pilot enhanced distributed engineering initiative. ISAT engineering analysis activities include individual assessment discipline teams conducting vehicle closure, safety and reliability, operations, and economic modeling, which are very sequentially interdependent. Following implementation of a semi-distributed environment and Product Data Management (PDM), several of the specific discipline activities became parallel in nature (Fletcher, 2001b) with teams working independently before sharing model input and output parameters. Applications of the e-

engineering approach to the ISAT case scenario are now described. Figure 5 shows an e-Engineering *Entity* view of the ISAT case task environment, where entities include key participants (shown in yellow) and artifacts (shown in green) as described previously in Figure 3. Entities are organized in terms of analysis activities and general sequencing of these activities in the ISAT case scenario. Participants are also identified by team role and geographic NASA center location.

This e-Engineering Entity view is then enhanced to an Entity/Interaction view shown in the diagonal and upper portions of Figure 6. This can be treated as a type of systems engineering functional or behavioral view of the distributed engineering environment. This Entity/Interaction view is necessary to capture before e-Engineering infrastructure and methodologies are tailored and implemented for a program or project scenario. This view drives the type of technology implementations which can meet team distributed functionality needs for projects and work packages. Both user and data-centric interactions shown in Figure 3, which are necessary for effective performance of distributed ISAT analysis tasks, are mapped to scenario user and artifact entities. User-centric communication interactions include audio, video, and messaging interaction channels between participant users and analysis sub-teams. Data-centric interactions include shared application control, viewing, referencing, storage of artifact files, and sharing of project files. In terms of the ISAT scenario, Figure 6 shows the need for user-centric interactions at the overall ISAT team level, but also at each analysis sub-team level. All participants should have the capability to conduct synchronous dialog with other sub-team member and dialog between sub-teams on an individual or group basis. Such dialog is more natural with synchronous video, audio, and instant messaging capabilities. Also shown is the need for data-centric interactions within and between sub-teams. Within sub-teams, users need to be able to

control, store, reference, and view modeling and analysis applications. Between groups, for viewing and referencing interaction channels as well as file sharing are needed for ISAT team activities, including model error checking, model parameter input and output between dependent model interfaces, and synthesis of analysis across models.

Another logical view of e-Engineering entities and functional interactions is shown in Figure 7, which has similar information in a matrix organization. Entities are organized along the diagonal, with e-Engineering interactions shown at matrix intersections for participant – participant and participant-artifact interactions. ISAT Team interaction requirements are shown along the top row. Clusters of sub-team requirements are also shown for between participant and model interactions.

By comparing this functional view with current or proposed implementations, an e-Engineering impact analysis can be conducted to analyze the traceability of functional requirements to implementations. As an example, Figure 8 shows this integrated functional and implementation view using the observed implementation of the ISAT engineering environment. On the lower half of the view current e-Engineering technology and processes can be identified for team, participant, and data model interactions. User-centric communication was dominated by an overall ISAT team room videoconferencing between centers. Data-centric communication employed an enterprise product data model (PDM) solution, which allowed integration between standalone analysis models, data storage, and file sharing. An e-Engineering impact analysis of the ISAT case highlights priorities for improvement for future distributed environments.

ISAT e-Engineering traceability issue #1 deals with the inadequacy of room videoconferencing to meet user-centric communication needs. ISAT consists of multiple teams and a group audio and video channel is inadequate for user communications by individuals

within and between teams. Possible technology solutions include multi-cast desktop videoconferencing, similar to the e-Engineering classroom at Old Dominion University, which allows individual and group audio, video, and application sharing. Such tools should include the capability to interactively view any conference participant (whether speaking or listening), have a list of current participant names, allow file transfer, and whiteboarding.

ISAT e-Engineering traceability issue #2 also deals with the inadequacy of room videoconferencing to meet user-centric communication needs. As shown in the functional view, multiple conferences between participants and sub-teams could be required at the same time. With a one channel room videoconferencing solution, only one session can occur at a time within a collaborative area. Desktop videoconferencing or other multichannel solutions can allow for parallel distributed dialogs to occur, which maps better into the parallel nature of the ISAT workflow. Such simultaneous multiple conferences are possible within a single center's collaborative area, or from individual participant desktops.

ISAT e-Engineering traceability issue #3 deals with the ability remotely view, reference, and control data model applications. If analysis tasks are conducted on standalone computers, without the ability for application sharing, the collaboration workflow becomes more like information sharing, where model input parameter values and results are "throw over the wall" and transferred to the next analyst or sub-team. Application sharing is essential to remotely view, reference, and control data model applications between multiple team members, which can help in process and error checking, as well as compress analysis times and resources requirements.

6. e-Engineering ISAT case analysis conclusion

This section of the ISAT case research was conducted by Dr. David Dryer at Old Dominion University and contains a preliminary system of systems engineering approach for the iterative design, implementation, and improvement of e-Engineering project teams. This approach includes use of the MeTA model for quickly increasing and maintaining basic and applied e-Engineering proficiency. The approach also outlines a systems engineering process to assist in identifying distributed collaboration interaction requirements for a particular task environment, designing infrastructure solutions, and graphically assessing the traceability impact of current and proposed environments.

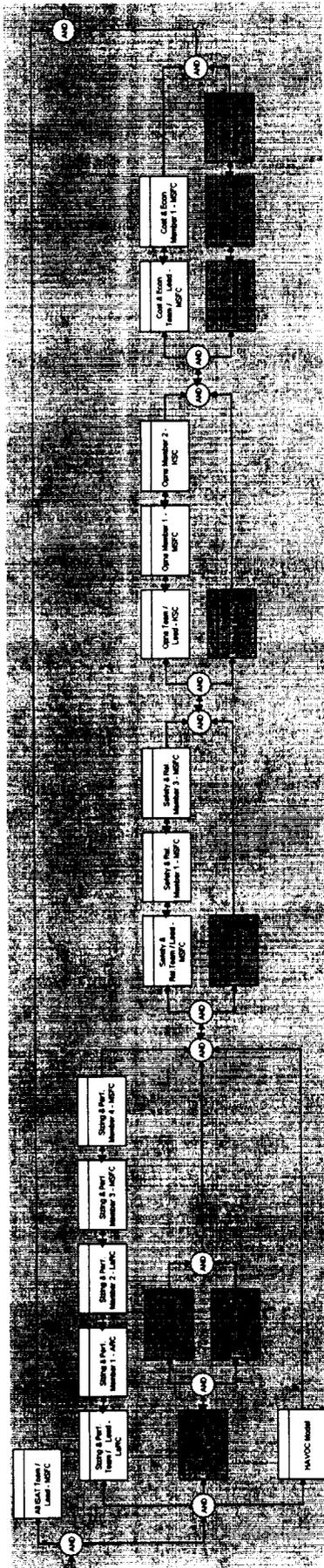


Figure 5. Entity (Participants and Model Artifacts) View of ISAT Case activities.

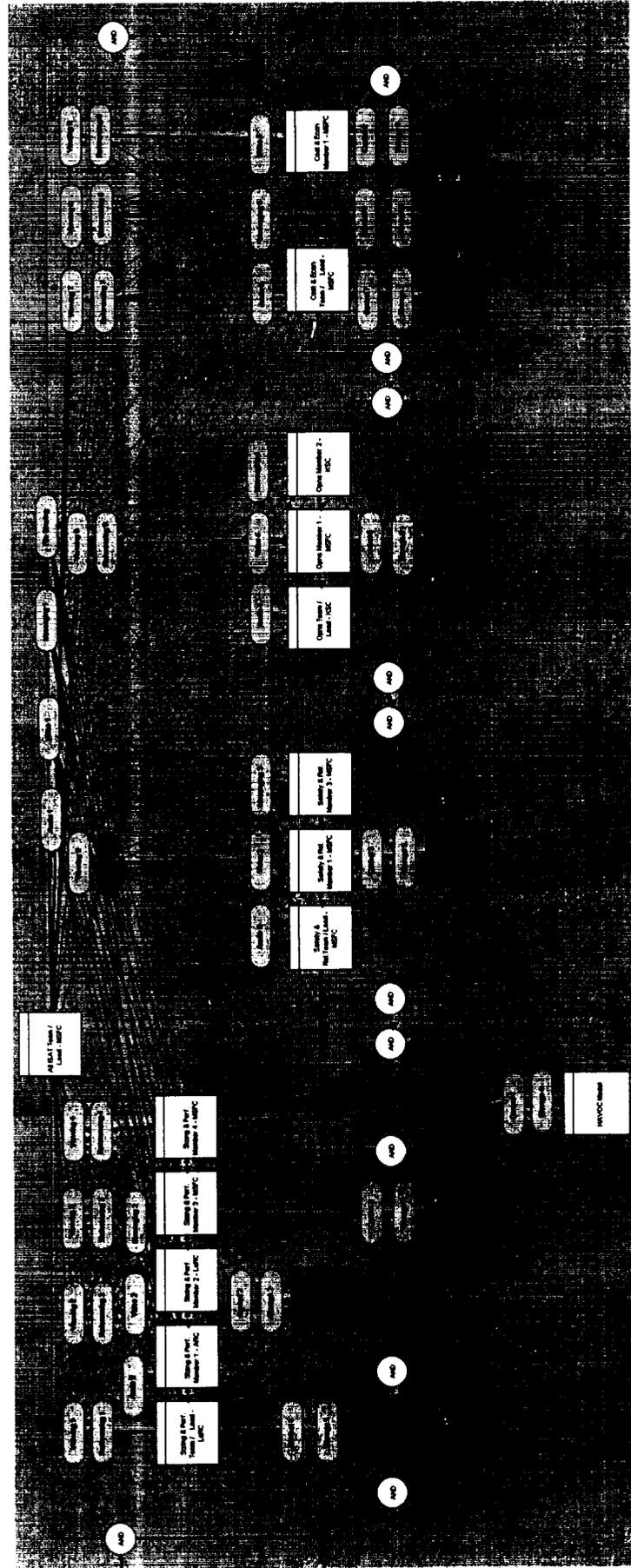


Figure 6. Enhanced Entity and e-Engineering Interactions (User and Data Centric) View of ISAT Case activities.

e-Engineering Interaction Functional View

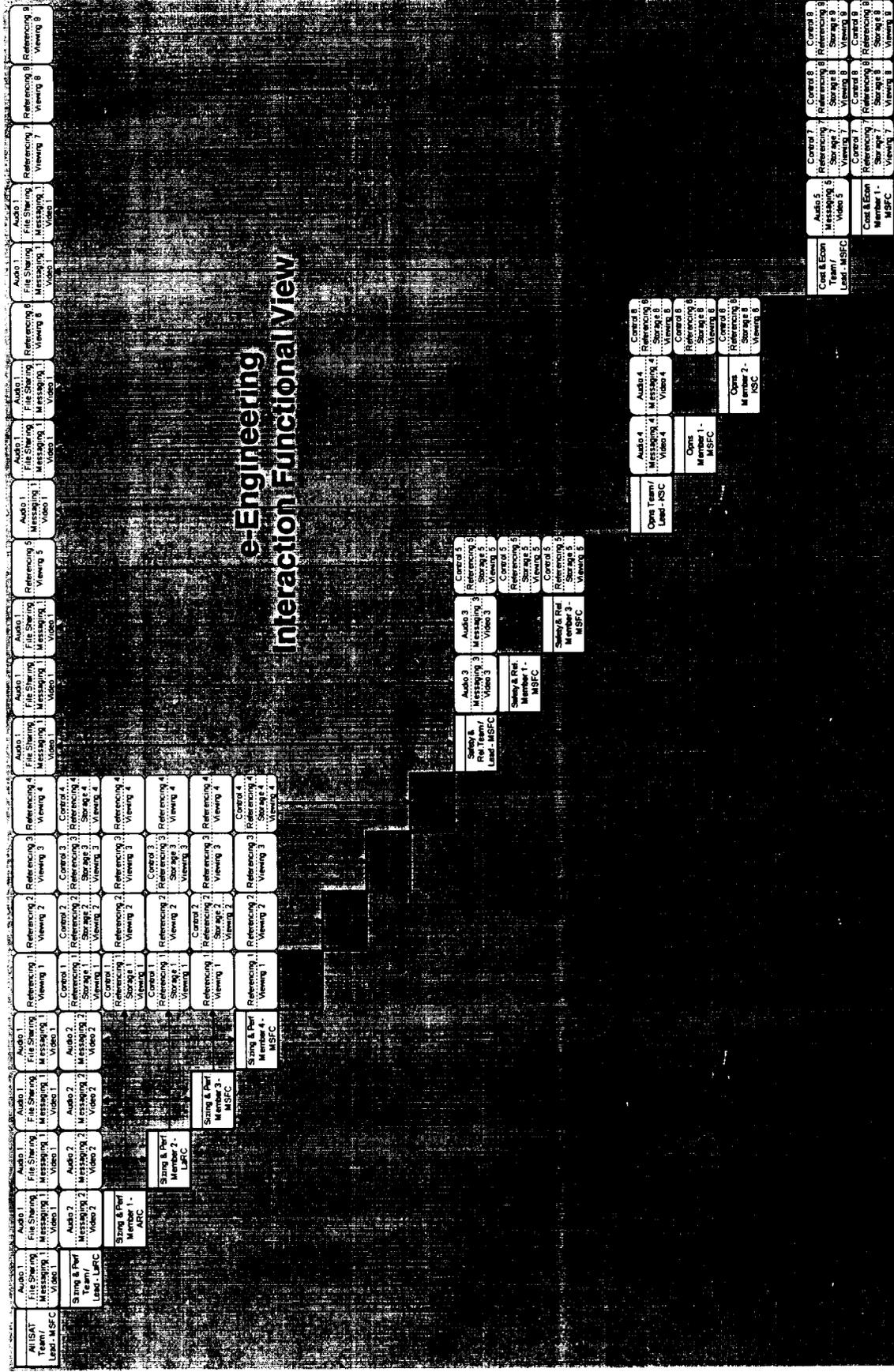


Figure 7. ISAT Case e-Engineering Entity/Interaction Functional View.

References

- Bider, I. and Khomyakov, M., 1997, One practical object-oriented model of business process. In Klimov H., Rumpe B., Simmonds I., (Eds.), OOPSLA'97 Workshop on Object-oriented Behavioral Semantics. Institute Für Informatik. Technische Universität München, TUM-19737, 25- 31.
- Boehm, B., 1988, A spiral model of software development and enhancement. *Computer*, 21(5), 61-72.
- Boehm, B., 2000, Spiral development: experience, principles, and refinements. *Special Report* CMU/SEI-00-SR-08, ESC-SR-00-08, W. J. Hansen (Ed.), Pittsburgh, PA: Carnegie Mellon University.
- Dix, A., Finlay, J., Abowd, G., & Beale, R., 1998, Human-Computer Interaction (2d edition), London: Prentice Hall Europe.
- Dorfman, M., 1977, Requirements engineering, *In Software Requirements Engineering*. Second Edition, R. H. Thayer and M. Dorfman, (Eds.), Los Alamitos, CA: IEEE Computer Society Press, 7-22.
- Duarte, D. L. and Snyder, N. T., 1999, *Mastering virtual teams*. San Francisco: Jossey-Bass Publishers.
- Fletcher, T. D. (2001b). Team task analysis of ISAT – Inter-center Systems Analysis Team during the August 2001 workshop as viewed from LaRC. Prepared for National Aeronautics and Space Administration.

Keating, C., Rogers, R., Unal, R., Dryer, D, Sousa-Poza, A, Safford, R., Peterson, W. (2002)

Systems Of Systems Engineering. *Proceedings of 23d Annual American Society for Engineering Management (ASEM) Conference (October 4-7), Tampa, FL.*

McGrath, J., 1990, Time matters in teams, *In Intellectual Teamwork: Social and Technical*

Foundations of Cooperative Work. J. Galegher, R.E. Kraut, and C. Egido (Eds.), Hillsdale, N.J.: Erlbaum.

Mills, A., 1998, *Collaborative Engineering and the Internet.* Society of Manufacturing

Engineers, Dearborn, MI.

Muench, D., 1994, *The sybase development framework.* Oakland, CA: Sybase Inc.

National Research Council, 1999, *Advanced Engineering Environments: Achieving the Vision:*

Phase 1, Washington, D.C.: National Academy Press.

National Research Council, 2000, *Advanced Engineering Environments: Design in the new*

Millennium: Phase 2, Washington, D.C.: National Academy Press.

PMI Standards Committee, 1996, *A Guide to the Project Management Body of Knowledge.*

Project Management Institute, William R. Duncan, Director of Standards, Newtown Square, PA.

References

- Bider, I. and Khomyakov, M., 1997, One practical object-oriented model of business process. In Klimov H., Rumpe B., Simmonds I., (Eds.), OOPSLA'97 Workshop on Object-oriented Behavioral Semantics. Institute Für Informatik. Technische Universität München, TUM-19737, 25- 31.
- Boehm, B., 1988, A spiral model of software development and enhancement. *Computer*, 21(5), 61-72.
- Boehm, B., 2000, Spiral development: experience, principles, and refinements. *Special Report* CMU/SEI-00-SR-08, ESC-SR-00-08, W. J. Hansen (Ed.), Pittsburgh, PA: Carnegie Mellon University.
- Dix, A., Finlay, J., Abowd, G., & Beale, R., 1998, Human-Computer Interaction (2d edition), London: Prentice Hall Europe.
- Dorfman, M., 1977, Requirements engineering, *In Software Requirements Engineering*. Second Edition, R. H. Thayer and M. Dorfman, (Eds.), Los Alamitos, CA: IEEE Computer Society Press, 7-22.
- Duarte, D. L. and Snyder, N. T., 1999, *Mastering virtual teams*. San Francisco: Jossey-Bass Publishers.
- Fletcher, T. D. (2001b). Team task analysis of ISAT – Inter-center Systems Analysis Team during the August 2001 workshop as viewed from LaRC. Prepared for National Aeronautics and Space Administration.

Keating, C., Rogers, R., Unal, R., Dryer, D, Sousa-Poza, A, Safford, R., Peterson, W. (2002)

Systems Of Systems Engineering. *Proceedings of 23d Annual American Society for Engineering Management (ASEM) Conference (October 4-7)*, Tampa, FL.

McGrath, J., 1990, Time matters in teams, *In Intellectual Teamwork: Social and Technical Foundations of Cooperative Work*. J. Galegher, R.E. Kraut, and C. Egido (Eds.), Hillsdale, N.J.: Erlbaum.

Mills, A., 1998, *Collaborative Engineering and the Internet*. Society of Manufacturing Engineers, Dearborn, MI.

Muench, D., 1994, *The sybase development framework*. Oakland, CA: Sybase Inc.

National Research Council, 1999, *Advanced Engineering Environments: Achieving the Vision: Phase 1*, Washington, D.C.: National Academy Press.

National Research Council, 2000, *Advanced Engineering Environments: Design in the new Millennium: Phase 2*, Washington, D.C.: National Academy Press.

PMI Standards Committee, 1996, *A Guide to the Project Management Body of Knowledge*. Project Management Institute, William R. Duncan, Director of Standards, Newtown Square, PA.