NATIONAL IVHS ARCHITECTURE DEVELOPMENT STRATEGY

James R. Blain
IVHS Architecture Systems Engineer
Jet Propulsion Laboratory

Ronald C. Heft
IVHS Architecture Development Task Manager
Jet Propulsion Laboratory

ABSTRACT

National information and control systems are emerging that require system architectures for deployment across the nation, e.g., air traffic control systems, military command and control systems, and other national information systems. The required characteristics for the national level architectures include modularity, openness, and evolvability. Modularity permits a variety of system configurations. Each may be designed to size a community's system with a level of functionality commensurate with the community's needs. Openness enables system interoperability with other systems at both horizontal and vertical levels. Evolvability requires the system to permit functionality and scope changes as the community and technology evolves.

Achieving these characteristics necessitates a formal strategy for developing architectures, for information and control systems, at the national level. A national level, information and control system is a system that is made up of systems, e.g., a supersystem. Developing an architecture is the modern approach for developing very large systems and supersystems. Formal system engineering and software engineering methodologies for developing national information and control system architectures do not exist. Most existing systems and software development methodologies need to be adapted to increase emphasis on front-end systems and software engineering methodologies. New system development methodologies for large information and control systems are emerging. Most do not support both the front-end systems engineering, i.e., defining system mission, and front-end software engineering, i.e., information modeling. Many of these methodologies are still implementation-oriented, resulting in premature commitment to system design solutions, prior to ensuring the design will satisfy all of the customer's needs. Further these system designs are usually limited by their architectural roots.

We have developed a formal strategy for supersystem architecture development, which includes newer methodologies such as *Strategies for Real-Time System Specification* by D. J. Hatley and I. A. Pirbhai, current United States General Accounting Office policy and the *NASA Systems Engineering Handbook.* Our strategy focuses on top level systems engineering methodologies by developing a mission definition document to: portray a vision of the mission, identify the operational, identify the security concepts, and develop the operational requirements needed to achieve the mission. The mission definition is used to generate a set of system functional requirements using functional analysis. The functional requirements are used to create a purely logical architecture (technology-free) capable of satisfying the system's mission. This Logical Architecture is modeled against mission scenarios and fine tuned to achieve the required functionality. Then the Logical Architecture is allocated to physical subsystems to enable the evaluation of alternative technologies in hardware and software configuration against mission scenarios. The best solution becomes the Candidate Architecture. The entire process is then iterated through several cycles until the optimal candidate architecture is obtained. The result becomes the Physical Architecture. This is the implementation blueprint.

The perspective for this paper is a survey of new system development concepts to identify the strategy to develop a national level architecture.

Version: Corrected Final 01/27/94

INTRODUCTION

A system can be defined as a set of interrelated components which interact with one another in an organized fashion toward a common purpose. Components of a system may be quite diverse, consisting of persons, organizations, procedures, software, equipment and/or facilities. A supersystem is a system that is made up of systems. A national level, information and control system is a system that is made up of a family of systems, e.g., a supersystem. Other large Information and Control (IC) systems may also be supersystems. Computer driven systems have been undergoing change in development methodologies over the last two decades. These changes are now addressing the proper approach for the development of an architecture for these systems. Architecture for a system is a relatively new term in the context of system engineering. One definition is: a system architecture is the highest level and earliest depiction of a system, where the system elements are named and the interconnections between the system elements are identified. A system architecture may be presented from several points of view. Today, the most common view is as an architecture model as depicted in Strategies *for Real-Time System Specification* by D. J. Hatley and I. A. Pirbhai. However, the approach for developing **a system architecture is often confused with that of a subsystem architecture, i.e., software architecture.** We prefer to view a system architecture as a non-specified depiction of a system in the form of the system's: mission definition, logical architecture and physical architecture. The system architecture must be completed before the system design can be specified for the deployment of that system in an operational environment. A supersystem architecture is a hierarchical depiction of the family of systems portrayed in the form of the system's: mission definition, logical architecture and physical architecture. These supersystem architectures are created to provide a non-specified depiction of a family of operationally related systems, each capable of autonomous functionality; but created synergistically as a system to enable greater operational functionality. Supersystem and system architectures are costly to develop and are not cost effective for most small systems. The product of architecture development is an implementation blueprint - a master plan - from which system design is specified. The approaches involved in developing system architectures and designs are the systems engineering approach and the systems analyst approach. The main difference between these approaches is their approach paradigm.

The systems engineering approach is customer's mission and system life-cycle oriented; starting in the proposal phase and ending when the system is removed from service. Systems engineering is an established discipline, that uses a top down approach to first transform the customer's required operational capabilities into a mission definition. The mission definition is integrated into a set of system functional requirements using functional analysis. The functional requirement set becomes the system functional specification and is used by systems engineering to oversee the overall system development. System engineering maintains oversight during the implementation phase, integrates subsystems into the system and conducts the system testing for final acceptance by the customer. System architectures may be depicted in functional flow block diagrams, and functional requirement sets.

The System Analyst approach is a computer system and software design oriented. The system analyst, as described in *Analysis and Design of Information* Systems by J. Senn, evolved from computing specialists applying their knowledge of computers for automating business functions. It was predicated upon the theory that it was easier for the computing specialist to understand business functions than it was for business specialists to understand computers. Thus, the System Analyst approach would review business operating procedures, confer with users and then determine what the computer needed to produce the desired output product using structured analysis. Software designs are usually depicted in data flow diagrams.

The advent of very large information systems is resulting in a merging of systems engineering and software engineering methodologies, due in part, to the complexities of computer information and control systems. Also, this merging was influenced by a staff study from the

United States General Accounting Office (GAO 92). This study addresses problems in developing large information systems that resulted in: large cost overruns, long development delays, and systems that do not meet the users' needs. The GAO found many cases of development shortcomings, i.e., inadequate planning for future user needs and premature commitment to a specific design, being the cause of systems not meeting the user's needs. The GAO recommended a top-down, structured approach in the systems engineering context. They provided a framework for developing information systems architectures. The framework expanded the architectural development processes required prior to committing to a specific design. The GAO's (GAO 92) framework identifies the necessity for accomplishing the following:

a. Mission Definition: Outlining a long-term vision of the mission, an operational and security concept, and the operational requirements.
b. A top-down Functional Architecture.
c. An Information Architecture.
d. A Data Architecture including a Data Dictionary.
e. An Application Architecture including security applications.
f. A Logical Architecture: an integration of the functional, information, data and applications architectures.
g. Alternative Physical Architectures composed of hardware, software, communications, security and data management capabilities. Also provides the alternative architectures audit trail of trade studies.
h. Target Architecture - the architecture chosen as the implementation blueprint.
i. Iteration of this development process several times, using lessons learned.

IVHS ARCHITECTURE DEVELOPMENT TASK

The Intelligent Vehicle-Highway System (IVHS) is the result of the Intermodal Surface Transportation Efficiency Act of 1991 **(ISTEA). ISTEA** established the IVHS Act; which requires the promotion of standards and protocol to promote **IVHS** technologies, the establishment of evaluation guidelines for IVHS operational tests and the establishment of an information clearinghouse. The IVHS Act also requires development of a completely automated highway and vehicle system which will serve as the prototype for future fully automated IVHS systems. The future IVHS systems are being developed via the *United States Department Of Transportation IVHS Strategic Plan Report To Congress,* December 18, 1992. Which calls for, in part, the advancement of existing Transportation systems in the functional areas of Traveler Information Systems (TIS), Traffic Management Systems (TMS), Commercial Vehicle Operations (CVO), Public Transportation Systems (PTS), and vehicle safety systems. The United States Department Of Transportation (USDOT) programs are to be advanced and integrated into an IVHS family of transportation systems. The future IVHS systems have been defined by twenty-seven IVHS User services, in the *NATlONAL IVHS Program P/an (DRAFT).* The development of most IVHS systems will be via the development of the IVHS Architecture; as a very large information and control architecture that encompasses present and future IVHS systems and their interfaces. Once the IVHS Architecture is developed; deployments of IVHS can begin across the nation.

Our task is to oversee the USDOT IVHS Architecture development. The IVHS system architecture must be a National Architecture - a master plan - to accommodate differing levels of IVHS capabilities for each community's needs across the nation. At the same time it must maintain interoperability and compatibility standards at the national level. The master plan will define the functional capabilities of the systems' interfaces and must provide upward compatibility with a variety of existing transportation management facilities in communities across the nation. The Architecture will have a built-in capability for expansion over its life cycle to meet the IVHS long-term mission objectives.

The architectural characteristics for the IVHS (information and control) architecture includes: horizontal openness, vertical openness and system agility. Horizontal openness permits variations in system configurations, in both size and functionality, to match each individual

community's requirements. Vertical openness permits the IVHS system to interface with all other IVHS systems at higher, lower and lateral levels. System agility permits the system to thrive in an environment of planned changes, to expand in scope as the community grows and to be capable of accommodating unpredictable changes in the system functionality.

Achieving these characteristics necessitates a formal strategy for developing information and control architectures at the national level. Developing a system architecture instead of a system design is the modern approach to developing large systems. However, as we were not the developers, we could only oversee the IVHS Architecture development. Oversight of a system development does not permit specifying the methodology, but we can specify the deliverables! Specifying deliverables that could ensure an adequate supersystem architecture required knowledge of supersystem architecture methodology and products. Therefore, we initiated a survey to identify a formal methodology for developing a supersystem architecture.

SCOPE OF THE SURVEY

The perspective for this survey is the application of new system development methods for a top level architectural development of a National IVHS System. New information and control systems methodologies surveyed include: *Analysis and Design of Information Systems* by J. Senn; *Essential Systems Analysis,* by McMenamin and Palmer; *Structured Development for Real-Time Systems* , Vol. 1, 2 and 3; by P. T. Ward and S. J. Mellor; and *Strategies for Real-Time System Specification* by D. J. Hatley and **I.** A. Pirbhai.

SURVEY RESULTS

Formal systems engineering and software engineering methodologies for developing national information and control systems architectures do not appear to exist. None of the surveyed methods actually provided the supersystem architectural methodology we were seeking for the National IVHS System. Existing systems engineering and software engineering methodologies need to be adapted to increase emphasis on front-end systems engineering (such as defining system mission) and software engineering (such as information modeling) methodologies. We decided to synthesize some of the new methods so they would provide the architecture development structure we needed to derive the deliverables for an IVHS architecture.

FORMULATION OF A SUPERSYSTEM ARCHITECTURE DEVELOPMENT METHODOLOGY

Our strategy for the IVHS (a supersystem) architecture development required a supersystem architecture development methodology, but we could not find one. Therefore, we had to formulate our own supersystem architecture methodology. **Our approach was a top-down system engineering approach.** We used the GAO framework (GAO 92) to provide the top level procedures, especially in the mission definition phase. This helps to ensure the procedure meets GAO audit requirements. We used the NASA "conceptual design" criteria, which calls for a depiction of mission needs and one or more credible, feasible designs. The depiction of mission needs reinforced the mission definition, identified the need for both graphic diagrams and written documents, reinforced the need for an information model, and provided general procures for developing system architectures. Further it emphasized the iterative development approach of performing several cycles of the system architecture process using lessons learned to improve each cycle. We agreed with "separating the essence of a system from its incarnation" from *Essential Systems Analysis,* by S. M. McMenamin and J. F. Palmer to help define a logical architecture. We liked the Ward & Mellor "heuristics approach to the evaluation of [architecture] models," which added emphasis to modeling the logical and physical architecture models against user's mission scenarios. We felt that the methods from *Strategies for Real-Time System Specification,* by D. J. Hatley and I. A. Pirbhai, provided the best real-time system engineering methodology for documenting our logical and physical architectures. In recent correspondence from Mr. Hatley, he stated the methods, from their book which he refers to as the **"HIP methods",** emphasize the

systems engineering approach and advocates establishing the system context with no limit on the size and scope of the system. Further, Mr. Hatley stated the "H/P methods" **emphasized that software [architecture] should not be considered in the higher layers of system definition,** as the decision to implement a given function in hardware, firmware, or software is dependent upon detailed technology decisions. We want to add "functions can be also performed by people", human operators can be Controller Processes. This is especially true for systems that will be automating human functions. We agreed with his de-emphasis of software, as the implementation decisions, at subsystem levels of specification, should not be not addressed until the physical architecture phase of the development. We also like his emphasis on the system engineering approach.

The "H/P methods" is a multi-perspective approach combining data flow decomposition with model components constructed in control and information space. These methods introduced new concepts to real-time information systems engineering, in the areas of requirement modeling, as a pure logical system model constructed from functional requirements. Requirements modeling is a method to partition a large finite state machine into pieces corresponding to the pieces of an analysis. The "H/P methods" also add architecture modeling of system configuration descriptions of classes of physical elements, categories of design elements, types of configurable items, etc., i.e., hardware and software components. The "H/P methods" modifies system analysis to include real-time system aspects of process control and timing. The "H/P methods" requirements model consists of a process model and a control model. The process model is constructed with a data context diagram, data flow diagrams, a data specification and a data dictionary. The control model is constructed with a control context diagram, control flow diagrams, and control specifications which include the large finite machine partitioning and timing requirements. We find that determining the terminators for the context of a system requires considerable knowledge of the system's functionality. Terminators should be autonomous systems functioning as sources or sinks for your system. If a terminator is required to provide functionality in the system (such as human operators), or your systems performs functions on the terminators (such as controlling its operation with data or control flows) they are not terminators and should be put inside the system's context.

Also, we included an **information model** in the requirements model as permitted in Appendix *C,* of *Strategies for Real-Time System Specification,* by D. J. Hatley and I. A. Pirbhai, which states, in part, "Information modeling [for] stored data systems: The data and control information and the access relationship of that structure to the process and control models needs to be specified. The information model is the third aspect to modeling a system and is not needed for all systems. The information model represents the customer's database." Hatley and Pirbhai portray the information model inserted into the requirements model's template, along with the process model and the control model. This type of a requirement model is roughly equivalent to the GAO's Logical Architecture (GAO 92) and what we call the logical architecture.

The (Physical) Architectural Model is the allocation of the functional model to a physical model, with all the subsystems interconnected. The architecture model is created to model the system design and depicts the configuration of all the physical modules of the system. We limited the Physical Architecture to a non-specified system depiction of hardware and software architectures. The Requirement model (Logical Architecture) is mapped into the architecture model including all the constraints of performance, growth, testability, safety, maintainability, reliability, system availability, and the interfaces. **The Architecture model should only contain the physical entities required to support the entities in the Requirements model.** The completed Architecture model consists of architecture context diagrams depicting the external physical data and control flows, architecture control and flow diagrams with architecture module specifications, architecture interconnect diagrams with architecture interconnect specifications and an architecture dictionary. Also a list of constraints against the system will be included. The Architecture Model is technology dependent and roughly equivalent to the GAO Physical Architecture (GAO 92) and what we call the Physical Architecture.

The requirement and architecture models complement each other and accommodate the allocation of requirements to the physical entities. The procedure is a leveled repletion of the functional requirement definition plans, followed by physical allocation of each level of the requirements going down through increasing levels of detail to encompass the entire system. The principle of **iterative development** has to be strongly emphasized for the development of supersystem architectures. The application of the mission definition: and logical and physical architectures should not be a "waterfall" process; wherein, each phase of the development is accomplished sequentially and then fall into the next phase without possibility of revision or change. There are several methods that may be used in the iterative development concept. One requires doing several **iterative cycles** of the "H/P procedures" using alternative architectures and lessons learned to improve each cycle. Any changes in the architecture requires another iteration to ensure the change does not adversely effect other areas of the architecture. Another method that may be used in the iterative development concept is the **concurrent development process,** where all phases are being developed together. Again, any changes in the architecture requires another iteration to ensure it does not adversely effect other areas of the architecture. Regardless of the iterative approach utilized, alternative architectures should be documented to provide an iterative development audit trail. The final result is the best physical architecture for that system, which is roughly equivalent to the GAO's Target Architecture (GAO 92) and what we call the Candidate Physical Architecture.

DERIVATION OF THE DELIVERABLES

Our development methodology as previously described permitted us to derive the contractual deliverables necessary to ensure the supersystem architecture development was technically complete and capable of fulfilling that mission. Our system's engineering approach was to specify three blocks of documentation for architectural deliverables. These deliverables are: Mission Description Documents, Logical Architecture Documents, and Physical Architecture Documents.

Mission Description Documents: A set of documents required to define the present and future Mission for the expected life-cycle of the system. They will contain the following documents for each time frame - a Mission Vision description, operational system concept document, including a security concept, and finally an operational requirements document.

Logical System Architecture Documents: A set of documents required to describe the Logical Architecture. They will consist of the system functional requirements, devoid of technology dependencies. These requirements will be documented in terms of a process model, a control model and an information model. The process and control models will be documented in terms of a system context diagram and flow diagrams, Processor Specifications (PSPECs) and Control Specifications (CSPECs), data dictionaries and associated graphics/text. The control model will also contain state transition diagrams. The information model may be addressed as Entity Relationship Diagrams, which combine data stores of all flow diagrams to depict the relationship between the data or an object diagram. Or, the information model may be implicitly addressed by consideration of coupling and cohesion, and may be embedded in the process and control models in the form of data stores, event-action diagrams and the data dictionary. In this case a trace matrix should be provided. Finally, a formal functional requirement document will be produced.

The Physical Architecture documents depict the hardware and software architectures: they are a set of documents required to define the candidate physical architecture. They will consist of: an architectural context diagram depicting the external physical data and control data flows between physical subsystems: an architecture physical subsystem specification which contains the elaborated and allocated requirements to

each of the physical subsystems; a list of any constraints against the architecture; and finally, an architecture dictionary.

SUPERSYSTEM ARCHITECTURE METHODOLOGY SUMMARY

New system development methodologies for large information and control systems are continuing to emerge. But for the present, we think the best methodologies are those which focus on developing the user's mission definition for the expected life-cycle of the system; a Logical Architecture to be evaluated against simulated mission scenarios and fine tuned to achieve the required functionality; and a Physical Architecture for the evaluation of alternative technologies in system hardware and software configurations. The best solution becomes the Candidate Architecture. The entire process is iterated through several cycles until the optimal physical architecture is obtained.

IVHS ARCHITECTURE METHODOLOGY

The National IVHS System will be an open-ended, open system architecture that can evolve along with technologies. An open-ended architecture will permit a variety of system configurations designed to match individual community's requirements across the nation. An open system permits interfaces with all other IVHS systems at higher, lower and lateral levels. The system's interfaces will be standardized to permit architectural coherence for users across the nation. The IVHS System must be agile enough to enable the system to thrive in an environment of change and to expand in scope as the community grows.

Our strategy for overseeing the IVHS (a supersystem) architecture development required a formal architecture development methodology. We had to formulate the supersystem architecture development methodology as described above. The contractual deliverables will provide the documentation necessary to ensure that the IVHS architecture development is technically complete and capable of fulfilling the IVHS mission description. Our strategy for the Further, it will require iterative systems developments with computer simulations of alternative, logical and physical architectures played against user's scenarios for the five, ten and twenty year scenarios.

These deliverables are the IVHS Mission Description Document, the IVHS Logical Architecture Document, and the IVHS Physical Architecture Documents.

IVHS Mission Description Documents: A set of documents required to define the IVHS Mission for five, ten or twenty year time frames. They will contain the following documents for each time frame - a Vision description, operational system concept document, including a security concept, and finally an operational requirements document.

IVHS Logical System Architecture Documents: A set of documents required to describe the IVHS Logical Architecture. They will contain the system functional requirements, devoid of technology dependencies. These requirements will be documented in terms of a process model, a control model and an information model. The process and control models will be documented in terms of system context and flow diagrams, Processor Specifications (PSPECs) and Control Specifications (CSPECs), data dictionaries and associated graphics/text. The control model will also contain state transition diagrams. The information model may be addressed as Entity Relationship Diagrams, which combine data stores of all flow diagrams to depict the relationship between the data or an object diagram. Or, the information model may be implicitly addressed by consideration of coupling and cohesion, and may be embedded in the process and control models in the form of data stores, event-action diagrams, and data dictionary. Finally, a formal functional requirement document will be produced.

The IVHS Physical Architecture documents depict the actual hardware and software architectures. They are set of documents required to define the candidate physical architecture. They will consist of: an architectural context diagram depicting the external physical data and control data flows between physical subsystems; an architecture physical subsystem specification which contains the elaborated and allocated requirements to each of the physical subsystems; a list of any constraints against the architecture; and finally an architecture dictionary.

CONCLUSION

We were successful in establishing a National IVHS Architecture Strategy that provides for an IVHS Architecture methodology that has been implemented for the IVHS Architecture Development Project.

GLOSSARY OF TERMS

The purpose of this glossary in this paper is to define the terminology used in this paper. The definition of terminology varies from one user to another. Therefore I have provided the definition of the terms as defined or indicated by references.

Conceptual design: The depiction of mission needs and one or more credible, feasible designs. (NASA)

1) A credible design must not depend on the occurrence of breakthroughs in the state of the art. While it may assume likely improvements in the state of the art, it is nonetheless riskier than one that does not. (NASA)

2) A feasible design is one that can be implemented as designed and can accomplish the system's goals within the constraints imposed by fiscal and operating environment. (NASA)

Deployment: Deployment of an IVHS architecture is a specified IVHS design configuration for an IVHS installation in a specific environment. (JPL)

IVHS Information and Control Architecture: An hierarchical depiction of a family of systems portrayed in the form of the system's: mission definition, logical architecture and physical architecture. (JPL)

Specific: Something peculiarly adapted to a purpose or use. (WEB)

1) A specific system design is the depiction of all physical components (hardware, firmware, software, people) and their interactions required to satisfy the system's mission, for deployment in a specific, operational environment. (NASA/JPL)

Supersystem: A system that is made up of systems. (WEB)

Supersystem Architecture: A supersystem architecture is a hierarchical depiction of the family of systems portrayed in the form of the system's: mission definition, logical architecture and physical architecture. These supersystem architectures are created to provide a non-specified depiction of a family of operationally related systems, each capable of autonomous functionality; but created synergistically as a supersystem to enable greater operational functionality. (JPL)

System: A set of interrelated components which interact with one another in an organized fashion toward a common purpose. (NASA)(IEEE)

1) The components of a system may be quite diverse, consisting of persons, organizations, procedures, software, equipment and/or facilities. (NASA)

System Architecture: The highest level and earliest depiction of a system, where the system elements are named and the interconnections between the system elements are identified. (NASA)

A non-specified depiction of a system in the form of the system's: mission definition, logical architecture and physical architecture. (JPL)

USDOT United States Department of Transportation, Washington, D.C.

BIBLIOGRAPHICAL REFERENCES:

(GAO 92) "STRATEGIC INFORMATION PLANNING Framework for Designing and Developing System Architectures," a staff study from the United States General Accounting Office (GAO) dated June 1992.

Analysis and Design of information Systems by J. Senn, New York, McGraw-Hill, Inc. 1984.

Essential Systems Analysis, by S. M. McMenamin and J. F. Palmer, Englewood Cliffs, NJ; Yourdon Press, 1984.

Structured Development for Real-Time *Systems* , Vol. 1, 2 and 3; by P. Ward and S. Mellor, New York: Yourdon Press, 1985.

Strategies for Real-Time System Specification D. J. Hatley and I. A. Pirbhai, New York: Dorset House, 1988.

(NASA) ***NASA Systems Engineering Handbook,*** September 1992.

United States Department Of Transportation IVHS Strategic Plan Report To Congress, December 18, 1992.

(WEB) *Webster's Ninth New Collegiate Dictionary,* MERRIAM-WEBSTER INC., Publishers, Springfield, Massachusetts, USA, 1988