ENGINEERING ENVIRONMENT AND AVIONICS VIRTUAL PROTOTYPING

Rudolf Volner[1]

Summary: Specialized system engineering tools, although capable of supporting specific tasks or functions, have limited use in most organizations and do not dearly fit into a broader project data architecture. Attempts by either tool users or developers to integrate applications outside of a bundled tool set have generally been limited to various forms of data import which creates recurring manual administrative effort and related configuration management problems by having data in multiple places. Collaborative engineering and virtual prototyping Is the application of advanced distributed modeling and simulation and engineering tools in an integrated environment to support technology development, system design, performance, cost, and producibility trade-off analyses throughout the entire product and system engineering life-cycle.

Key words: environment, avionics, modeling

INTRODUCTION

A virtual prototype allows design teams to walk through the system design to see how components have changed. The virtual prototype serves as a common frame of reference for designers, engineers, and managers. It allows the program manager to establish a level playing field for consistent comparisons among alternative concepts and designs. By means of virtual prototypes, the program or technical manager can work with the user to define requirements. Historically, program requirements are difficult to quantify and verbalize. Users are able to state what they do not want much easier than describing what they do want. A simulation model developed in parallel with the hardware or technology development allows the scientist, engineer, or end-user to refine system requirements early in the engineering process. All will participate in and adhere to the High Level Simulation Architecture initiatives being managed by the Modeling and Simulation:

- Simulation System (SS) - a distributed, object-oriented simulation architecture and system for training for the operational level of war (campaign and mission level simulation),
- Warfare Simulation (WS) - focused on Joint campaign analysis,
- Modeling and Simulation System (MSS) - distributed, object-oriented simulation architecture and system for the tactical level of war (engagement simulations).

Systems engineering tools largely support tasks that occur prior to design implementation and can be both technical and programmatic in content. These tools are numerous and address a wide variety of specialty tasks. However, they have somewhat limited usage relative to primary design tools. Although there is no common classification

[1] Prof. Ing. Rudolf Volner, Ph.D., Department of Air Transport, Institute of Transport, Faculty of Mechanical Engineering, VŠB -Technical university of Ostrava, E-mail: rudolf.volner@vsb.cz, Tel.: +420 596 99 1765

system for systems engineering tools, for the purposes herein they can be represented as follows:

- Functional and Architectural Analysis,
- Requirements and Verification,
- Risks and other Project Data,
- Simulation and Models.

The basic objective is to create an integrated systems engineering and project management environment that would enable users to easily move from one function to another without having to work with multiple, disconnected tools and manually integrate data between these tools. An integrated SE (*Systems Engineers*) environment is best viewed as part of a broader integrated engineering and program management environment. This perspective could be seen as supporting network-centric concepts. An expanded vision looks to an environment where different tools, including specialized applications, internally-developed applications, and common office files, could be integrated into a sort of plug-and-play environment. Using legacy integration approaches, this was impossible because interfaces had to be specifically built between individual instances of applications but using a different approach leveraging emerging technologies, it is feasible as I will address later.

1. LEVELS OF INTEGRATION

The type or method of integration can vary greatly. In order to establish a common frame of reference, the integration level must be clearly defined. Levels of application integration can generally be broken down into three distinct categories described as follows:

- Link or Reference,
- Data Association,
- Interface Integration.

2. AVIONICS VIRTUAL PROTOTYPING

Over time, the DAT (*Department of Air Transport*) has developed a robust set of laboratory facilities, computer hardware and software resources and other domain specific assets focused on supporting various individual mission objectives. To develop the next generation technologies required for integrated avionics systems, DAT needs to network and electronically integrate its existing facilities, resources, and assets to create an environment to support virtual prototyping.

Virtual prototypes and simulation models developed in parallel with the hardware or technology development allows scientists, engineers, or end-users to refine system requirements early in the technology and system development process.

This virtual prototype allows the user and developer to see the impact of design changes and trade studies using the model can be performed throughout development as an integral part of the systems engineering process.

3. DEFINING A COLLABORATIVE ENVIRONMENT FOR VIRTUAL PROTOTYPING

Collaborative Engineering and Virtual Prototyping is the application of advanced distributed modeling and simulation and engineering tools in an integrated environment to support technology development, system design, performance, cost, and producibility trade-off analyses throughout the entire product and system engineering life-cycle. Collaborative engineering enables all members of a DAT to continuously interact through electronic modeling and data interchange; increases insight into life-cycle concerns; permits earlier testing and experimentation through virtual test ranges; and accelerates physical production through process optimization using virtual factories. Collaborative Engineering simulations, with integral product and process models, will permit cietai1ed knowledge to be obtained earlier in the conceptual and preliminary design phases where it can have the most influence on life-cycle cost. More emphasis will be placed on the collaborative development of virtual prototypes of key technology products to demonstrate their military effectiveness and worth in an integrated systems/mission environment.

CVP (*Collaborative virtual prototyping*) can be implemented in many organizational structures. Traditional hierarchical workplaces, concurrent engineering environments, and work groups focused on rapid prototyping are a few examples. Implementation of a CVP system requires attention to the necessary enabling technologies and supporting infrastructure. A crucial part of a CVP system implementation is educating personnel on how CVP can meet customer, organizational, and individual goals as well as decrease time-to-market, lower life-cycle costs, and improve product quality.

4. BASIC ARCHITECTURAL REQUIREMENTS TO SUPPORT APPLICATION INTEGRATION

In order to integrate two or more applications using common legacy methodology, the applications must meet basic characteristics associated with multi-user applications. Most systems engineering applications (especially those in the modeling and simulation area) are not designed as multiuser applications in the first place. These constraints, associated with multi-user applications, can create some real problems.

- Data served from common file location,
- Secured multi-user access,
- Data level interface,
- Ability to support multiple simultaneous users.

Depending on the type of integration merited and the intent to re-use the connection between projects or applications, certain data architecture requirements may also apply as prerequisites to data integration:

- Common Data Elements,
- Common Data Structure,
- Integrated Security Structure,

- Ability to bypass User Interface (*Interface Integration only*).

In passing or sharing data between two or more specific applications, some sort of interface must exist to extract the data and make it readable to the receiving applications. There are several common ways of doing this that have relative advantages and disadvantages:

- Data table association,
- Application Program Interface (API),
- Static data import/export.

5. ENABLING TECHNOLOGIES FOR CEE ARCHITECTURE

- User System Interfaces - User system interfaces provide two basic functions: render output data, and accept input control data. Note that the term user can refer to either a human or an inanimate resource. The EE (Engineering Environment) will have many different user system interfaces. Some examples are:
  - Display System,
  - Control System,
  - Scientific/ Engineering Data System,
  - Imaging System,
  - Audio System,
  - Hardware Stimulation System,
- Applications - Applications are the resources needed by a user to perform a EE task. These resources can include models, simulations, COST software tools, hardware stimulators, hardware simulators, cost models, and other laboratory facilities. These applications can also include resources such as distributed white boards, video teleconferencing systems, multi-user virtual environments, or e-mail needed by users to collaborate with other users. Representative applications include:
  - Collaboration,
  - Domain Specific,
  - Product Model,
  - Process Model,
- Application Interfaces - Application interfaces connect specific applications to the EE. These interfaces allow the EE to manage and control the applications and provide a suite of tools to DAP engineers. Representative application interfaces include:
  - Hardware,
  - Software,
- Middleware - Middleware includes the software and tools that “glues” a set of applications together and provides a seamless EE to support user needs:
  - Application Support,
  - Context Management,
  - Object Management,
  - Communications Support,

    - User Interface Support,
    - Data Interfaces Management Support,
- Communications Infrastructure - The communications infrastructure provides the necessary functionality to send and receive data and information in a distributed processing environment:
    - Collaborative Intranet,
    - Virtual Simulation,
    - Constructive Simulation,
- Data Interfaces Management - Data interfaces management provides the services to manage all of the different types of data from the other EE architecture layers:
    - Access/Archive Management,
    - Manipulation Management,
    - Composition Management,

REFERENCES

(1) VOLNER, R. a kol. *Flight Planning Management*, Akademické nakladatelství CERM sro. Brno, 2007, ISBN 978-80-7204-496-2, 630 str.

(2) VOLNER, R. Based Analysis of Interaction Between Human Subject and Artificial System - Impacts of Driver Attention Failures on Transport Reliability and Safety, *Perner´s Contacts - Elektronický odborný časopis*, pp. 404-410, 3/2011, ročník 6, Univerzita Pardubice, ISSN 1801 – 674X, str. 410.

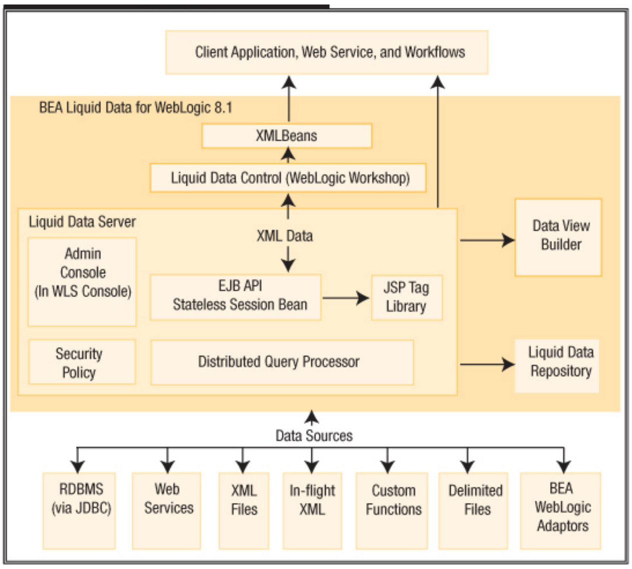
(3) VOLNER, R. Human Interaction System – Inteligence Network, *11th Biennial Baltic Electronics Conference, BEC 2008*, Tallinn, Estonia, October 2008, pp. 217 – 220, IEEE Catalog Number CFP08BEC-PRT, ISNN 1736 – 3705.

(4) VOLNER, R. Home Network and Human Interaction System, *Ninth International Conference on Enterprise Information Systems, Proceedings Human-Computer Interaction, ICEIS 2007*, Funchal, Portugal, June 2007, pp. 323-327, ISBN 978-972-8865-92-4.

(5) COHEN J. Human Robots in Myth and Science, *A.S. Barnes and Company*, South Brunswick and New York, 1966.

(6) VOLNER, R., Boreš, P. A Human Classification System for Biometric Parameters, *Electronics and Electrical Engineering N^o 6 (62)*, Kaunas University of Technology, Academy of Sciences of Lithuana, Vilnius Gedimas Technical University, Riga Technical University, Tallinn Technical University, Kaunas 2005, pp. 16-21, ISSN 1392-1215.

(7) VOLNER, R., SMRŽ, V. Wireless Air data systems, *Perner´s Contacts - Elektronický odborný časopis*, pp. 135 - 141, Srpen 2009, Univerzita Pardubice, ISNN 1801 – 674X.



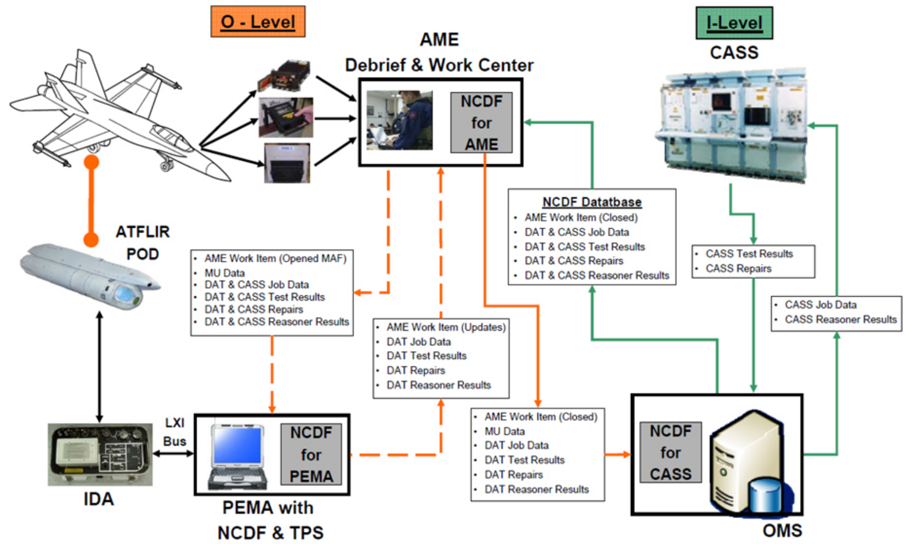


Fig. 1 - Basic Systems Engineering Data Architecture

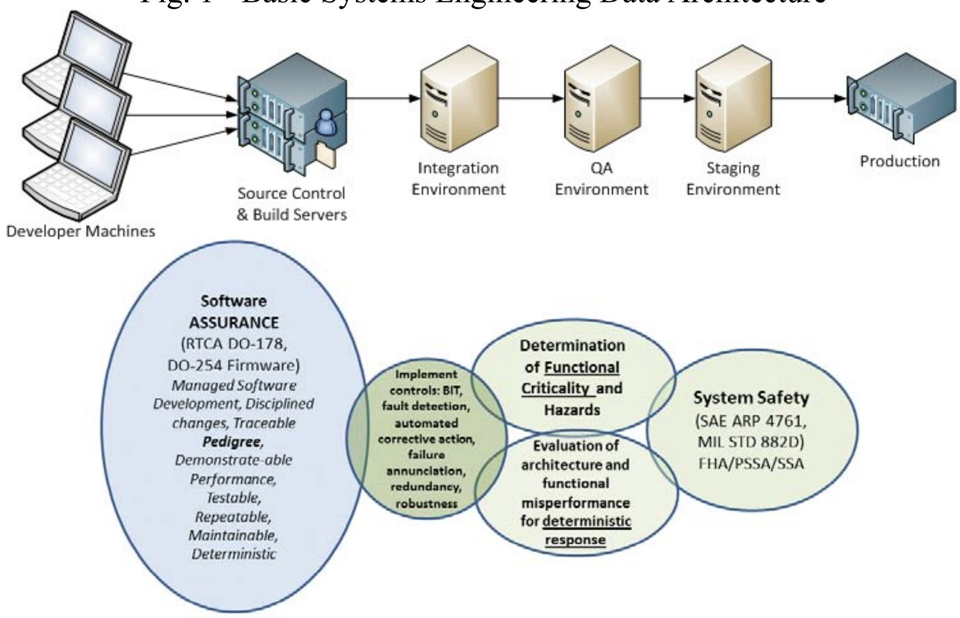


Fig. 2 - Systems Engineering Integrated Environment

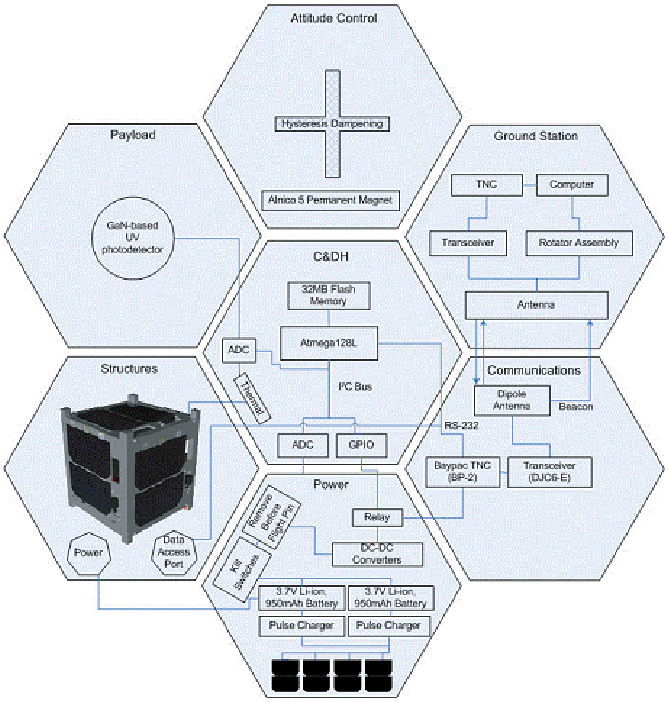


Fig. 3 - Avionics systems

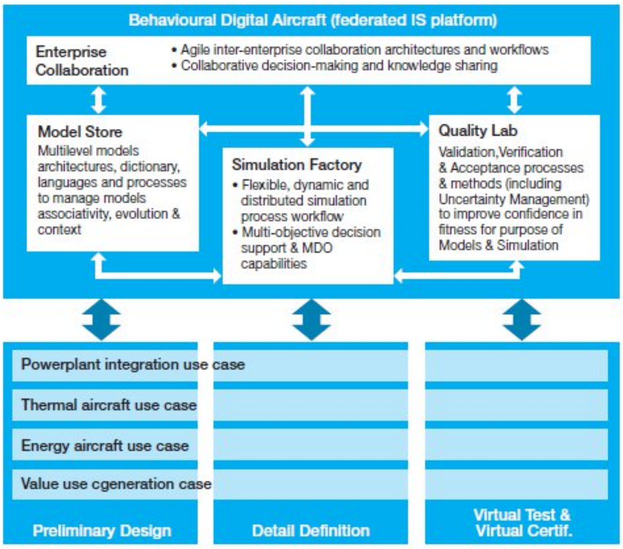


Fig. 4 -Avionics Virtual Prototyping

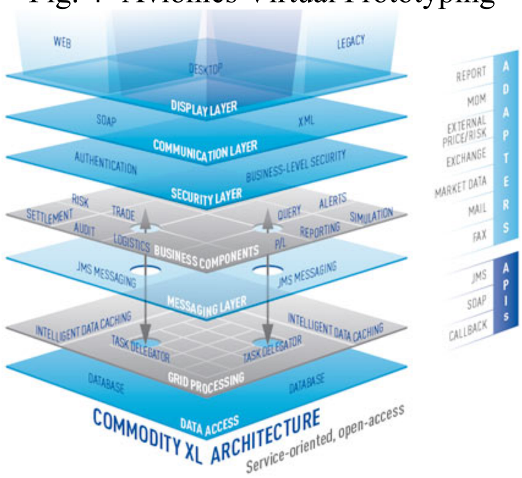


Fig. 5 - EE Technical Architecture