Proceedings of DETC'01
2001 ASME Computers and Information in Engineering Conference
September 9-12, 2001, Pittsburgh, Pennsylvania

DETC2001/CIE-EIM-15

AN ARCHITECTURE FOR VIRTUAL PROTOTYPING OF COMPLEX SYSTEMS

Bipin Chadha
Lockheed Martin ATL
1 Federal St, AE 2W, Camden, NJ 08102
bchadha@atl.lmco.com

John Welsh
Lockheed Martin ATL
1 Federal St, AE 2W, Camden, NJ 08102
john.j.welsh@atl.lmco.com

ABSTRACT

Virtual prototyping has emerged as a significant enabler for cost effective management of complex products over their lifecycle. The virtual prototypes play an important role through all life-cycle phases of products. Due to the breadth of impact, significant challenges arise in developing virtual prototypes. This paper describes the issues and tradeoffs that are important for implementing virtual prototypes. A detailed architecture is described that has been used to implement a successful virtual prototype for a complex DoD (Department of Defense) program. The paper discusses key properties of an architecture and strategies to enable its evolution over large timescales. The paper then describes the benefits of this approach.

INTRODUCTION

Many of the current industrial products are becoming increasingly complex. This rise in complexity is a natural outgrowth of the competitive environment. The rise in complexity is along many dimensions ranging from product complexity to process complexity to the complexity of the supply chains that design and manufacture these products.

This rise in complexity comes with increasing lifecycle costs, making many of the current DoD systems increasingly unaffordable. A key strategy to address lifecycle costs is through virtual prototyping during the entire lifecycle of a product, especially during early design stages. Approaches to address cost include intelligent application of new design and manufacturing technologies, new materials, leverage of modern computing and communication approaches, and analysis/optimization of process for individual processes and the total life cycle process. Decisions which affect cost in all life cycle phases should be addressed to the maximum extent feasible during the front end design phases, at a point when perturbations in design or strategy are practical. Decisions made at this phase of the process have the greatest effect on life-cycle cost – hence expansion of the set of life cycle phases and associated trade studies issues to be addressed at this level has the most dramatic effect on overall cost optimization.

Virtual prototypes need to support a comprehensive set of analyses that will be performed on the product; hence, all aspects of product data and behavior need to be represented. Building virtual prototypes of complex systems being designed by a multi-organizational team requires new architectural concepts and redesigned processes. Implementation of these new architectures is complex and leveraging commercial technologies is necessary to achieve feasible solutions. One must carefully consider the state of the current commercial technologies and frameworks as well as the organizational and cultural aspects of organizations that use these systems. Some of the key architectural principles and concepts are described in [Chadha 2000].

ISSUES AND TRADEOFFS

This section addresses some of the key issues and tradeoffs in implementing a virtual prototyping environment (VPE).

Integration Across Domains/Disciplines

As the granularity of information being managed increases, the issues of multidisciplinary interactions and overlaps become prominent. The practice of decomposing the problem along discipline lines is beginning to cause problems as the complexity of artifacts being designed is increasing and the assumptions of linearity no longer hold. This causes significant problems during integration and test phases where the pieces need to come together. It is common practice in analysis to "integrate" analysis by passing data from one tool to another. Again this approach breaks down when interactions across disciplines are tightly coupled and non-linear. Models, tools, and analysis techniques are needed that are multi-disciplinary in nature and account for non-linear interactions. One significant problem is maintaining consistency of information across disciplines as each discipline recreates the same underlying information to suit their needs. For example, it is very common to find multiple copies of the product structure information between various disciplines or teams. The current approach of creating a skeleton product structure representation and configuration controlling it only provides a partial solution. Solutions that seek integrated product structure solutions often do not account for process dynamics since they typically do not take advantage of interface concepts– e.g., an interface can act as an insulator by mitigating change propagation and facilitating autonomous changes that do not violate the interface.

Scalability, Modularity, and Robustness

Existing information system architectures are not very robust, in the sense that small changes in the software can cause significant disruption even when these changes have been carefully planned and implemented. This also contributes to the lack of modularity (and vice versa). It is difficult to add new modules or remove older ones without significant impact. These factors affect the scalability of these systems. There is some improvement through the use of component-based architectures, but the results are not spectacular. As a contrast there are many examples of non-engineered, highly complex systems that exhibit scalability, modularity, and robustness, i.e., biological, economic, and social systems [Coveney 95]. These systems exhibit these properties by utilizing bottom-up approaches of evolution and decentralization. Information systems need to leverage these principles to achieve similar results. One such example is the Internet architecture that has proven to be one of the most scaleable information systems in existence.

Engineering Analysis and Formal Change Control

Current engineering information management (EIM) solutions are either based on traditional PDM approaches that stress formal change and configuration control or are based on simulation integration approaches based on data translation and exchange. Traditional PDM based approaches are not suitable for engineering analysis and design exploration as they impose rigid controls and provide little support for exploratory analysis. The simulation integration approaches allow design exploration at the expense of information management and change tracking, and cannot ensure data and design integrity. This makes it difficult to recreate analysis results and ensure validity of results. Approaches that combine the two capabilities are required for effective virtual prototyping.

Commonality and Standards

Most information management approaches ignore qualitative factors that become important as commonality approaches 100 percent. Existing architectures only account for technical aspects and ignore cultural and organizational aspects. For example, standards should be evaluated in context to determine whether they are applicable, mature, and cost effective for the problem at hand. Applying a standard just because it is a standard is likely to produce less than desirable results. Interestingly, the study of complex systems shows that the greater the diversity of a complex adaptive system, the more fit it is [Kauffman 1995]. From a strategic perspective, diversity (presence of more than one option or more than one approach) is a key defensive mechanism for a non-linear system in a volatile environment [Kelly 1998]. This implies that standardization can lead to vulnerability (for example, wide spread use and acceptance of Microsoft products makes everyone more vulnerable to virus attacks). The critical issue is not commonalities in architecture but understanding the trade-offs to determine the right amount of commonality for a given domain in a given context.

Another debate that typically rages on is whether business needs drive technology or does the technology drive emerging business needs. Since these two are interdependent they cannot evolve in isolation. The two need to co-evolve simultaneously in today's dynamic competitive landscape.

Disproportionate focus on Geometry, ECO Automation, and Design Build Phase

Industry focus in the area of product information management has been largely in the areas of managing geometry, automation of the ECO (Engineering Change Orders) processes and the transition of design into manufacturing. Focus on geometry comes largely due to CAD tools, and the increased visibility and marketing appeal of geometry. Focus on ECO's and design build phase stems from the fact that most problems surface in these two areas. As is typically the case, symptoms always get more attention when causal factors are hidden or are separated from effects in space or time. It is well known that early conceptual design impacts over 80% of the lifecycle costs, yet receives little attention. Similarly it has been documented that defect-fixing processes usually receive more attention than the defect preventing processes [Senge 1994, Womack 1990]. This situation is systemic and can result in a vicious cycle where defect-fixing processes become more and more important and take away resources from defect preventing processes. This leads to more defects that further drives need for more defect-fixing. During investment decisions, the noisiest issue wins. Models that help prioritize and phase EIM implementation initiatives are critical to help guide management investment decisions.

Lack of visibility of non-technical issues

It is fairly common for system builders to concentrate on technical issues and technology [Chadha 1995]. Non-technical issues however heavily influence real systems. These issues tend to be qualitative in nature and are often implicit. Tools and methodologies for addressing non-technical issues and enabling their tradeoffs are not widely known or practiced. Technically sound concepts and systems can fail or become impractical when non-technical issues are ignored. Non-technical issues are typically ignored in analysis and come into play through intuitions of experienced designers and architects. Best or successful solutions are almost never the most technically elegant ones.

Implicit Assumptions of Linearity and Coupling

It is common to assume linearity and lack of coupling across disciplines or across systems of a product [Deng 2000]. These assumptions allow us to ignore unnecessary complexity for simple problems and systems and therefore can be very effective. However ignoring non-linearity and coupling for complex systems can cause significant problems and can lead to systemic defects that are hard to predict and understand [Perrow 1984]. The insights gained through linear systems almost never apply to non-linear systems. These issues are overlooked because the assumptions of linearity and coupling are often made implicitly.

Ability to Model, Design, and Analyze Complex Systems

As the systems we build become more complex and coupled, better tools and methodologies are needed to model, design and analyze complex systems. Complex systems confront conflicting design constraints making optimization hard to achieve. Complex systems can be thought of as interconnected networks and small perturbations can move a stable system into a chaotic regime. It is also known that these networks exhibit maximum fitness at that transition point [Kauffman 1995, Coveney 1995, Holland 1998]. It is desirable to understand the properties of the transition points and tools to position and maintain these systems at the transition point. It is important to note that the transition point is not static. For complex systems, their fitness landscape changes as the system moves, essentially the system and its environment co-evolve. The complex systems typically never reach the optimal point; they are continuously evolving towards higher levels of fitness. Traditional optimization methodologies can therefore become ineffective.

KEY PROPERTIES

Properties or characteristics of the good architecture should be understood and incorporated in the design process where reasonable. Examples of high-value properties relative to architecture approaches include the following:

- Accessible—the architecture should enable easy access via standard web interfaces or thin-client software.
- Scalable—the architecture should handle a large number of concurrent users without severe degradation in performance.
- Extensible—the architecture should link with new domain tools, should allow formation of a federated system, and should support incorporation of new technologies as they become available.
- Interoperable—the core infrastructure should be able to communicate with different information systems (PDM) and machines of different platforms.
- Reusable—the core infrastructure should be reusable across multiple programs.
- Available—the architecture should support the creation of applications that are available as dictated by user requirements.
- Deployable—the architecture should support the creation of applications that are easily deployable across geographically and organizationally distributed teams.
- Affordable—the architecture should support the creation of applications that are affordable.
- Reliable—the architecture should support the creation of applications that are highly reliable.
- Maintainable—the architecture should support the creation of applications that are easily maintainable.

Design practices, technologies, and use of cots (versus custom implementation) need to be traded off during the architecture-development process, with consideration of each property as a measure of the relative quality of the architecture.

ARCHITECTURE EVOLUTION

The architecture has to support a highly complex product over a long life-cycle. This requires that the architecture must evolve through changes in requirements, technologies, business practices, organizations and their roles. The architecture will also need to be flexible, and robust, not requiring frequent changes to the underlying infrastructure. The architecture approach therefore calls for a managed and controlled evolution of a stable architecture, and a careful balance of the inherent conflict in these goals. This strategy can be summarized as:

- Decouple fast changing things from slow changing things
- Decouple Information Architecture from System Architecture
- Decouple Logical Architecture from Physical Architecture
- Use of pervasive standards in the development of information architecture to enable interoperability
- Use Component Based Architectures with Explicit Interfaces for easier replacement
- Use Layered architectures to isolate layers from changes, spread responsibility to accommodate change, as well as promote reuse
- Use a deliberate approach to identify and incorporate new technologies and components into the architecture as they mature
- Use approaches from complex adaptive systems [Kelly 1995, Clippinger 1999]:
  - -Allow for redundancy, some inefficiency, and diversity.

-No centralized control. Many interconnected independent parts.
-Iterative, incremental development.
-Architecture with many intermediate stable configurations.
-Use of pervasive communication standards.

ARCHITECTURE

This section describes the architecture for a virtual prototyping environment (VPE). The VPE is a Windchill® based solution that logically organizes product model information, and its associated behavior. This enables users in the extended enterprise to analyze the impacts of their decision on critical success factors over a product's lifecycle. The VPE brings together core capabilities from configuration management and systems engineering to create a system that enables organizations to significantly reduce their time to market and life-cycle costs. The VPE ties together many commercial systems engineering and engineering analysis tools with a configuration controlled product representation. VPE also provides the capabilities to integrate an organization's legacy tools and databases.

VPE exploits the fact that design decisions during conceptual and preliminary design impact 70-80% of the life-cycle cost of a product. VPE enables the enterprise users to evaluate the impacts of their design decisions and perform tradeoffs to create an optimal design. VPE enables the entire supply chain to be involved in this collaborative process, influencing the lifecycle cost when the design is still fluid, thus achieving significant reduction in costs and cycle time. VPE also builds in tools and mechanisms to ensure superior quality in the product being designed.

VPE supports a modular open architecture (Figure 1). VPE components can be implemented in an incremental phases to provide rapid results and benefits. VPE supports many standard protocols such as HTML, XML, CORBA, HLA, and Java RMI.

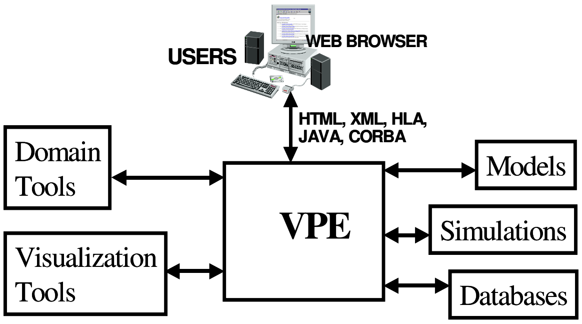

Figure 1. VPE Logical Architecture

A high level functional architecture is shown in Figure 2. Key components of the VPE are:

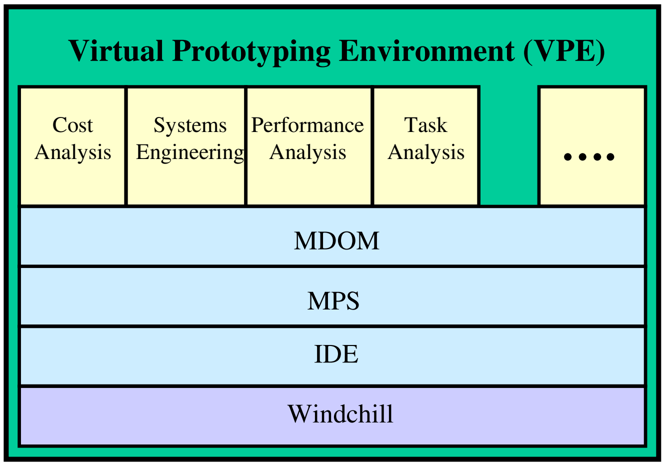

Figure 2. VPE Functional Architecture

- **Windchill**. VPE is built on PTC's Windchill product that provides the core PDM (Product Data Management) and CPC (Collaborative Product Commerce) functions using a scalable web based architecture.
- **IDE (Integrated Design Environment).** IDE is the collaborative design component of the VPE that provides a project based environment for complex products that need to be designed by distributed multi-company teams. IDE provides enhanced document management and workflow functions as well as collaboration services. IDE also provides product visualization services across the distributed enterprise.
- **MPS (Multidisciplinary Product Structure).** MPS enables multidisciplinary teams to collaborate on a product definition. MPS supports full configuration management and change management capabilities while capturing discipline specific attributes, business objects, and related documentation. MPS maintains traceability between requirements and the products that satisfy those requirements. MPS enables different disciplines to view product structure from their own perspective while maintaining overall integrity between different disciplines (Figure 3).

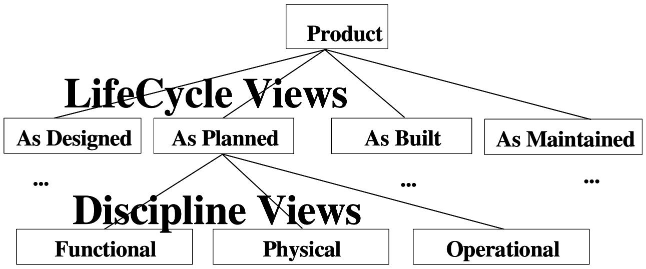

Figure 3. Multiple Views of Product Structure

- **MDOM (Multidisciplinary Design Optimization Manager).** MDOM enables sophisticated analysis and optimization capabilities by leveraging MPS and enabling a “sandbox” environment where engineers can experiment with different design approaches. MDOM allows users to create reusable design experiments that can be automated to search the design space to find optimal designs. MDOM integrates several proven optimization algorithms to enable users to quickly setup their design problems and search for effective solutions. MDOM provides mechanisms to track all analysis work and its results and it ties this information to the requirements as well as the design configurations (Figure 4).

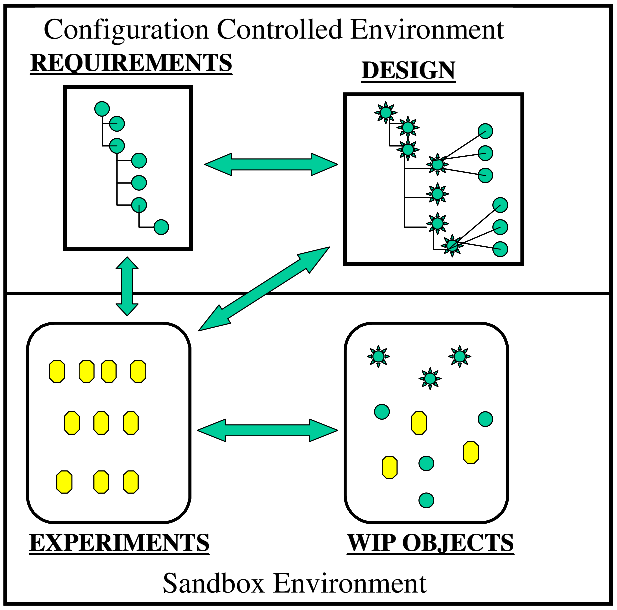

Figure 4. Analysis Experiments Structure

- **Application Threads.** Application threads are targeted towards key design processes. Some examples are cost analysis, performance analysis, systems engineering analysis, task analysis, etc. These threads encode engineering processes and their corresponding tools and databases to enable designers to perform analysis on configuration controlled designs or experimental designs. Application threads leverage MPS and MDOM capabilities to ensure all design and analysis work is traceable and configuration managed. These capabilities together provide true engineering design and change management support.

- **Interoperability.** VPE achieves interoperability through the use of pervasive industry standards such as HTML, XML, CORBA, HLA, and Java RMI for interaction among components and clients. Interoperability is also achieved via the use of commercial EAI frameworks (Enterprise Application Integration). These tools provide cost effective approaches to integration where mature standards are not available and provide convenient mechanisms to use standards where available. The VPE architecture leverages such tools where appropriate. Examples of such tools are PTC’s InfoEngine, CrossWorld’s CrossWorld, etc. The architecture must make judicious use of existing standards and EAI frameworks, and be prepared to incorporate new standards as these mature, gain market acceptance, and provide value to the business processes.

BENEFITS

VPE enables participants from multiple disciplines to collaborate on product development across the extended enterprise. Information and processes are streamlined, reducing the time associated in sharing knowledge among disciplines. VPE achieves significant efficiencies because it addresses engineering processes in conjunction with data administration processes typically addressed by PDM and CPC solutions.

- **“Common Source, Specific View” of product information**
  VPE removes the barriers between different disciplines to create a complete representation of the product. From any location, regardless of the source systems, users can access the subset of the product information necessary to make the best business decisions. This eliminates the need for specific disciplines to maintain their own little database of product information tailored to meet their needs and keeping it synchronized with the master product definition.

- **Consistency of Multi-Disciplinary information**
  Linking related information across the extended enterprise into a single product representation ensures its continual consistency. When all users have access to the most up-to-date product information, the latency

problems between different disciplines traditionally associated with copied and duplicated information are eliminated. This is done without creating monstrous objects that carry large amounts of information.

- **Consolidated and Simplified Configuration and Change Management** Goal of VPE is to significantly reduce the number of design changes during a product's lifecycle. VPE eliminates the design changes that result from poor communication or due to use of obsolete information. There however is still a role for change management due to change in the environment or customer needs. The efficiency and accuracy of communications improve when change processes within the extended enterprise are streamlined. Information can be automatically routed to the discipline specific users and systems affected by a proposed activity. The activity is targeted only to the users impacted. The result is that users immediately understand the changes required, and change information is synchronized across multiple business systems. Since all discipline views share the same underlying product representation change impacts become visible much more naturally.

- **Improved product and business agility** Because the VPE allows detailed technical information on products, suppliers, and partners to be modified rapidly, companies can take better advantage of technical innovations and new business opportunities. The MPS allows very early design tradeoffs to occur in an objective and configuration controlled environment, and allows such tradeoffs to occur throughout the lifecycle of the program. The resulting business and product agility can be a source of significant competitive advantage.

CONCLUSIONS

Virtual prototyping is a critical mechanism for cost-effective design of complex systems. New architectural concepts are emerging to support virtual prototyping of complex systems across company boundaries. When implementing virtual prototyping architectures, realize there are limits to benefits achieved through commonality across distributed implementations and that soft/qualitative factors play an important role in architectural trade-offs. This paper outlined some key architectural principles and concepts that improved the practicality of large-scale virtual prototyping efforts. This approach combines the capabilities of PDM systems that provide configuration control with a sandbox environment that allows engineers to explore design concepts and analyze the impacts. Many COTS tools and technologies are emerging to support implementation of these concepts. We have successfully implemented this environment for a major DoD program. Effectively leveraging these tools and technologies will enable design teams to manage the scope of the effort while maintaining cost and schedule constraints.

REFERENCES

1. Chadha, B., "A Model Driven Methodology for Business Process Engineering," ASME Engineering Database Symposium, 1995.
2. Chadha, B., Welsh, J., Architecture Concepts for Simulation-based Acquisition of Complex systems, Summer Computer Simulation Conference, July 2000.
3. Clippinger, J.H. , Order from the Bottom Up: Complex Adaptive Systems and Their Management, in The Biology of Business, Jossey-Bass Publishers, 1999.
4. Coveney, P. , Highfield, R., Frontiers of Complexity, Ballantine Books, 1995.
5. Deng, Y.M., et al., Constraint-based functional design verification for conceptual design, Computer Aided Design, 32, 2000.
6. Holland, J.H., Emergence: From Chaos to Order, Perseus Books, 1998.
7. Kaufmann, S., At Home in the Universe: The Search for the Laws of Self-Organization and Complexity, Oxford University Press, 1995.
8. Kelly, K., "New Rules for the New Economy", Viking, 1998.
9. Kelly, K., Out of Control: The New Biology of Machines, Social Systems, and the Economic World, May 1995.
10. Perrow, C., "Normal Accidents," Basic Books, 1984.
11. Senge, P.M., et al., "The Fifth Discipline Fieldbook," Currency Doubleday, 1994.
12. Waldrop, M. M., Complexity, Simon & Schuster, 1992.
13. Womack, J.P., Jones, D.T., Roos, D., "The Machine that Changed the World," MIT Press, 1990.