INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

# PATTERNS OF PRODUCT DEVELOPMENT INTERACTIONS

Steven D. Eppinger and Vesa Salminen

*Keywords: process modeling, product architecture, design teams, design structure matrix, complexity management*

## 1 Introduction

Development of complex products and large systems is a highly interactive social process involving hundreds of people designing thousands of interrelated components and making millions of coupled decisions. Nevertheless, we have created methods to study the development process, identify its underlying structures, and critique its operation.

In this article, we introduce three views of product development complexity: a process view, a product view, and an organization view. We are able to learn about the complex social phenomenon of product development by studying the patterns of interaction across the decomposed elements within each view. We also compare the alignment of the interaction patterns, between the product, process, and organization domains. We then propose metrics of product development complexity by studying and comparing these interaction patterns. Finally, we develop hypotheses regarding the patterns of product development interactions, which will be helpful to guide future research.

## 2 Methodology

In this research, we study product development situations by assessing the patterns of interactions within three domains and then compare the patterns across the domains.

### 2.1 Three Product Development Domains

To study product development, we believe that there are three relevant domains: product, process, and organization. In complex development situations, each of these three domains is decomposed in order to manage the complexity. We therefore begin our analysis by documenting the decomposition of each of the three domains:

- Product: A complex product or large system is decomposed into sub-systems, and these in turn may be further decomposed into sub-assemblies and/or components.
- Process: A full development process is decomposed into phases or sub-processes, and these in turn may be further decomposed into tasks, activities, and work units.
- Organization: A large development organization is decomposed into teams, and these in turn may be further decomposed into working groups and individual assignments.

## 2.2 Patterns of Interactions

Once we have documented the decomposition, we then document the patterns of interaction between the decomposed elements. It is interesting to do so within each domain (Figure 1):

- Product: The architecture of the product is defined not only by the decomposition of the complete product into elemental components, but also by the interactions between these components. The interactions may include well-specified interfaces and undesired or incidental interactions. System architecture design principles [1, 2] suggest ways to plan architectures with minimal interactions across sub-systems, maximizing the density of interactions within. Documentation of complete patterns of system architecture interactions has been accomplished using matrix-based methods [3]. Analysis of such patterns may be used to suggest clusters forming effective product modules.
- Process: The product development process is generally a complex procedure involving information exchange across the many tasks in order to execute the work. Various network-based methods have been used to map and study development processes [4, 5, 6, 7]. Analysis of product development processes allows us to study product development efficiency and to suggest process improvements.
- Organization: The organization structure determines who works with whom and who reports to whom. However, in development organizations we are particularly interested to study the communication patterns of the people conducting the technical development work. This follows from well established methods used to study communication networks in R&D organizations [8] and can be used to assess whether necessary interactions are taking place within the organization.

## 2.3 Comparison Across Pattern Types

We believe that the three types of patterns should be strongly related (arrows in Figure 1). After all, the development organization is executing the development process, which is implementing the product architecture. When we can compare the map of interactions in one domain to another, we hope to be able to answer questions such as:

- Does the organization properly execute the development process?
- Is the development process effectively implementing the product architecture?
- Are the architecture interactions driving the organizational communications?

But the comparison across such different types of data can be problematic. We have found that where there exists a one-to-one mapping from one domain to another, a direct comparison becomes straightforward. For example, if there is a single development task assigned to each individual team member, then a direct comparison between process and organization is possible. Similarly, where there is a single team assigned to each subsystem, we may directly compare the interactions within the organization to the interactions within the product architecture [9].

In practice, a perfect one-to-one mapping rarely exists in real and dynamic engineering design environments. Many industrial product development situations involve scarce or shared resources, multi-tasking, outsourcing, and dynamic or uncertain development demands, all of which make the analysis difficult. Utilizing a many-to-one or a many-to-many mapping from one domain to another yields a model of potential interactions, not

simply expected ones. Since this reduces the predictability of the model, we prefer to conduct the analysis in situations with simpler structures.

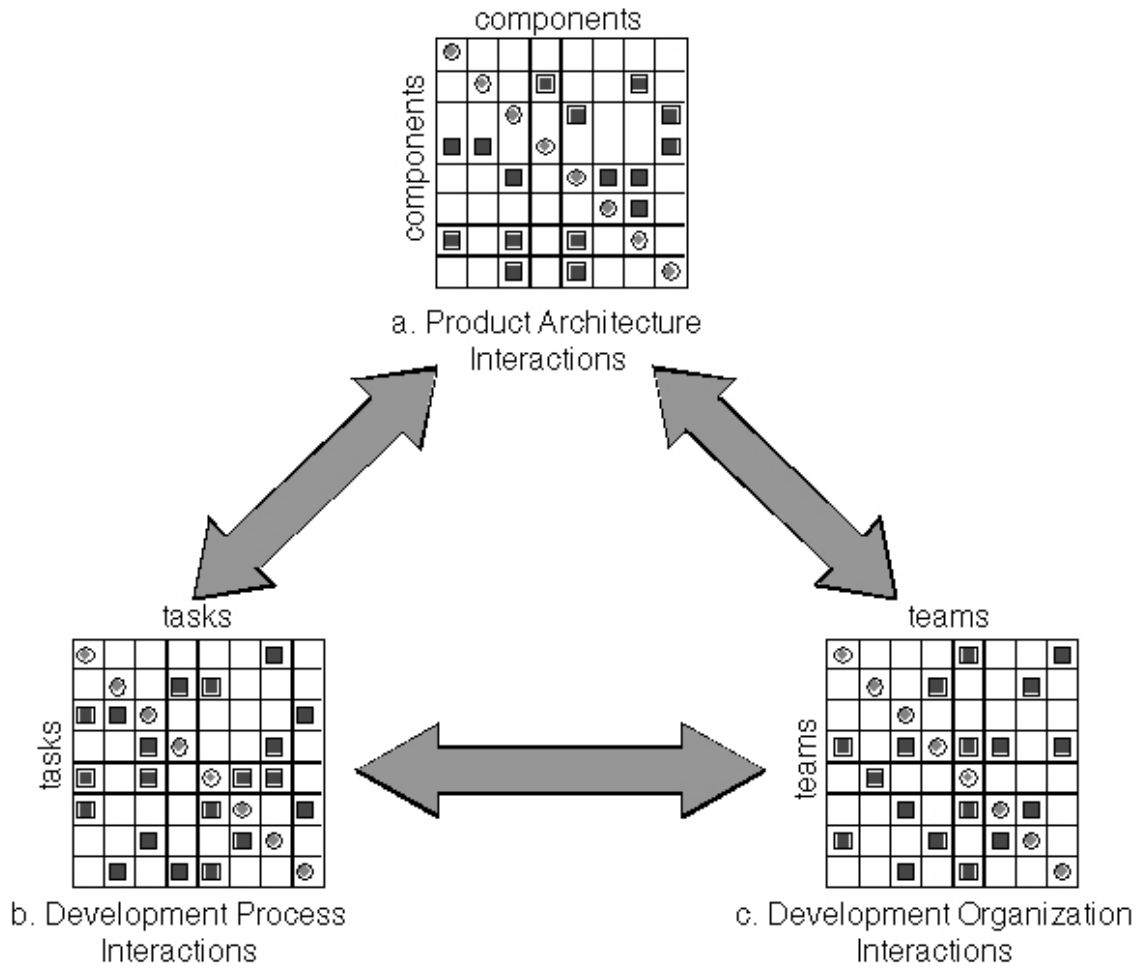


Figure 1. Three domains of product development interactions: product, process, and organization.

# 3 Industrial Examples

While case studies of this type take a great deal of effort, we have been able to make progress by utilizing graduate student internship projects at several companies. We are not able to describe these projects in detail here, but publications are available documenting the results of each application.

Figure 2 provides industrial examples showing the patterns of interactions in the three individual views, product architecture (Figure 2a), development process (Figure 2b), and development organization (Figure 2c). Figure 2 also shows examples of comparing across two of these three domains at a time (Figures 2d, 2e, and 2f). We have not yet attempted to compare all three views together for a single industrial example.

## 3.1 Product Architecture Example

Figure 2a shows a model representing the decomposition of a climate control system into 16 components and documenting the product architecture as interactions between the components. 34 interactions were identified along technical dimensions of spatial, energy, materials, and information [3]. Clusters of these interactions identify groups of highly

interrelated components, thus suggesting modules for development, production, and/or potential outsourcing.

## 3.2 Development Process Example

Figure 2b shows a matrix illustrating the procedure followed by an automobile manufacturer to determine the feasible layout of the engine compartment based on a digital mock-up using CAD solid models [10, p.349]. Interactions in this type of model represent flows of information and data between the tasks. The highly iterative sets of tasks become apparent through analysis of these process data and such a model is useful for process reengineering.

## 3.3 Development Organization Example

Figure 2c shows the decomposition of the organization used to develop a new automobile engine. The organization involved 22 teams, each with responsibility for a major component or subsystem. The matrix depicts the interactions across the 22 teams in terms of the frequency of their required technical communications. The clustering of the teams suggests an efficient arrangement of five system-engineering team assignments [11, 12].

## 3.4 Comparing Product Architecture to Organization

Figure 2d shows a comparison of the interfaces specifying the product architecture with the communications inside the development organization for a jet engine. In this case, there was a single product development team responsible for the development of each of the 54 components. This study not only confirmed the ability of design interfaces to predict technical communication, but also revealed several reasons why development professionals do not communicate even when their components interact, and further reasons why teams do interact while their components do not share a direct interface [9]. This research also identified differences in the behavior of teams designing modular components from that of teams designing distributed components [13].

## 3.5 Comparing Development Process to Organization

Figure 2e shows a comparison of the product development process to its development organization. In this study of electronics hardware components, there was not a one-to-one mapping of development tasks to individuals in the organization. While this hindered the comparison, it was still possible to show that the process model predicts technical communications in the organization much better than earlier models based on geographical layout of the personnel [14].

## 3.6 Comparing Product Architecture to Process

The comparison of the product development process to the product architecture of an elevator system is a case study still in progress. This example allows us to study the differences between the nominal and actual development processes and how these changes arise from the particular architecture chosen for the product studied.

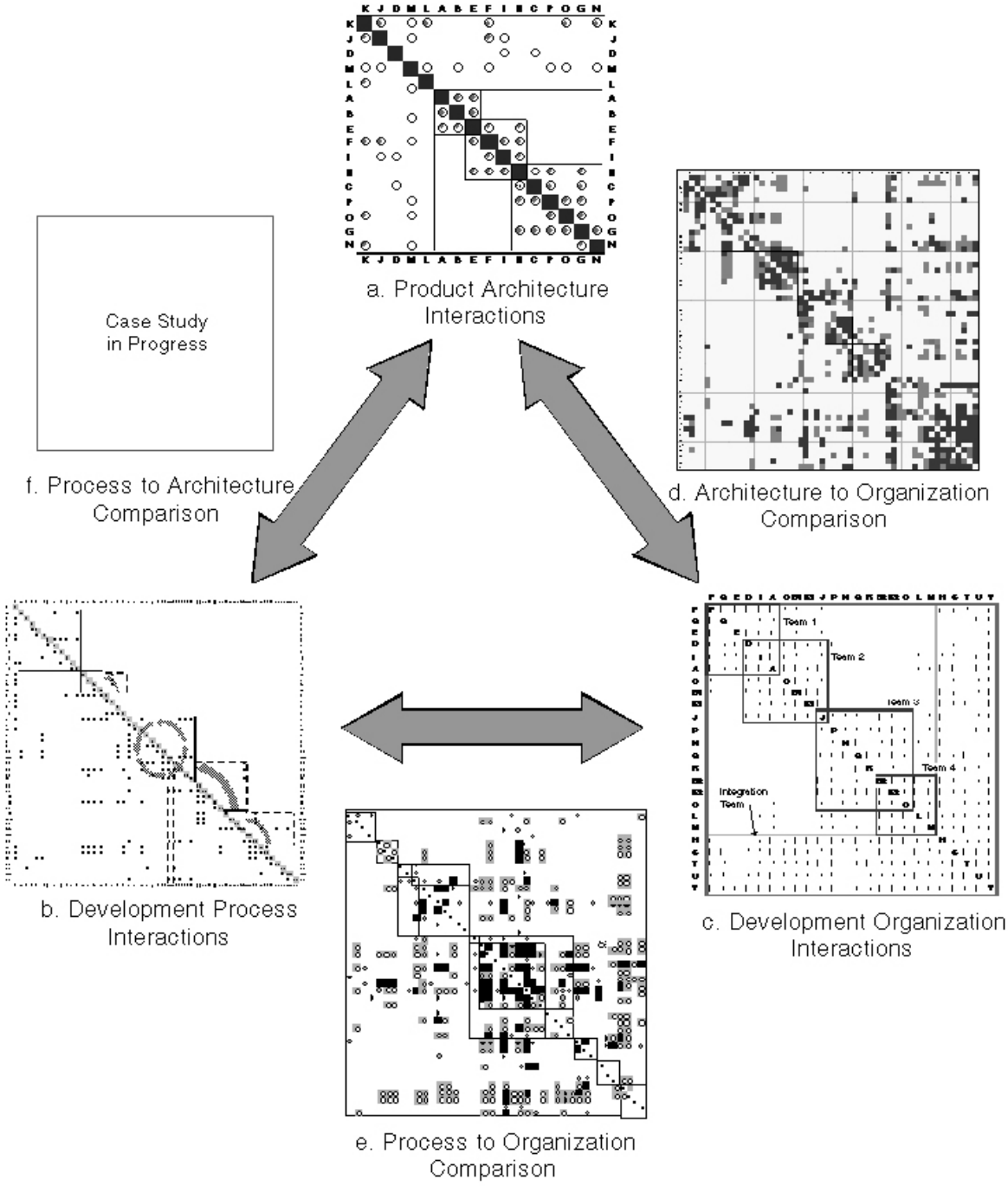


Figure 2. Examples showing matrix-based mapping of interactions in the product architecture (a), product development process (b), and development organization (c). We have also found it possible to compare such models across these domains (d, e, and f).

# 4 Discussion

## 4.1 Impact on Industrial Practice

The methods summarized in this paper have proven useful for diagnosing and improving product development processes, product architectures, and development organizations. In particular, analysis in the three individual domains provides direct benefits:

- Product: Analysis of the product architecture suggests more effective module and sub-system boundaries, highlights critical interfaces, and identifies appropriate outsourcing opportunities.

- Process: Analysis of the product development process leads to streamlining and accelerating the process, reducing and focusing design iterations, identification of failure modes within the process, and replacement of chaotic information flows with more formal procedures where necessary.
- Organization: Analysis of the product development organization can yield more effective system team arrangements and formation of system engineering functions for better integration of the overall product or system.

The three possible comparison views require somewhat more work to build two independent models and interpret their patterns jointly. Still, we have found that these analyses serve to help diagnose cultural and dynamic causes of process-related and organizational failures to efficiently develop the selected product architecture.

## 4.2 Complexity Metrics

Meaningful measurement of complexity can serve to improve our understanding of and ability to work with complex systems. With the help of complexity metrics, it will be possible to track complexity changes over several product generations. We may also be able to benchmark one company's product or process complexity with respect to its competitors. We have not utilized complexity metrics in our research thus far, and this remains an interesting area for future exploration. However, we believe that useful complexity metrics will consider several factors:

- The number of decomposed elements (components, tasks, or teams in our three views)
- The number of interactions to me managed across the elements
- The uncertainty of the elements and their interfaces
- The patterns of the interactions across the elements (density, scatter, clustering, etc.)
- The alignment of the interaction patterns from one domain to another

## 4.3 Hypotheses for Future Research

In considering what we have learned through many case studies, we have formulated several hypotheses about the dynamics of product development interaction patterns. Future empirical research will involve additional case studies, data collection, and analysis specifically designed to test these hypotheses:

1. Maturity: One hypothesis we have is that the density of the known interactions within any particular view varies with maturity of the product architecture, experience of the organization, and skill in managing the process. Specifically, we hypothesize that at first there are quite few interactions known. Then with experience, more interactions become evident. Finally, a mature architecture has more focused and clustered interactions, with others eliminated or minimized in impact.
2. Learning: We expect to find that, of the three views, experience builds most quickly in the product architecture view for complex, engineered products. This is because engineers learn quickly about the product and its technology, even while the development process and organization remain informally structured.
3. Evolution: We believe that the pattern of interaction within each domain changes over time, not in a random or unplanned manner, but with respect to a reference model [15]. Such a model may be the should-be product development process, the perfect product architecture, or the ideal organization. We believe that the presence of a reference process or architecture will affect the changes in the interaction patterns over time.

4. Co-Evolution: We further hypothesize that the interaction patterns in the three domains change in coupled ways. The organization evolves to address deficiencies in its ability to implement the development process and product architecture. Furthermore, the product architecture and development process may change to compensate for shortcomings in the development organization.
5. Alignment: Finally, we expect to find that industrial firms in which the interaction patterns across the three domains are well aligned will outperform firms for which the patterns are not aligned.

# 5 Conclusion

This paper presents three important perspectives for studying product development: product architecture, product development process, and the development organization. Within each domain, we focus on the pattern of internal interactions. We analyze these patterns to learn about the particular product development situation and how to improve it. We are also able to compare patterns across the three domains to assess the effectiveness of the process and organization to develop the particular product. After using this approach to study several industrial situations, we have developed some hypotheses which may guide future research in this area.

**References**


[1] Alexander, C. *Notes on the Synthesis of Form*, Harvard University Press, Cambridge, MA, 1964.

[2] Rechtin, E. and Maier, M.W. The Art of Systems Architecting, CRC Press, Boca Raton, 1997.

[3] Pimmler, T.U. and Eppinger, S.D. "Integration Analysis of Product Decompositions", ASME Conference on Design Theory and Methodology, Minneapolis, MN, pp. 343-351, September 1994.

[4] Steward, D.V. "The Design Structure System: A Method for Managing the Design of Complex Systems", IEEE Transactions on Engineering Management. vol. EM-28, no 3, pp. 71-74, August 1981.

[5] Marca, D.A. and McGowan, C.L. SADT: Structured Analysis and Design Technique, McGraw Hill, New York, 1988.

[6] Warfield, J.N. A Science of Generic Design: Managing Complexity through Systems Design, Intersystems Publ., Salinas, CA, 1990.

[7] Eppinger, S.D., Whitney, D.E., Smith, R.P., and Gebala, D.A. "A Model-Based Method for Organizing Tasks in Product Development", Research in Engineering Design. vol. 6, no. 1, pp. 1-13, 1994.

[8] Allen, T.J. "Architecture and Communication among Product Development Engineers", MIT Sloan School of Management Working Paper, no. 3983, 1997.

[9] Sosa, M.E., Eppinger, S.D., and Rowles C.M. "Understanding the Effects of Product Architecture on Technical Communication in Product Development Organizations", MIT Sloan School of Management Working Paper, no. 4130, August 2000.

[10] Ulrich, K.T. and Eppinger, S.D. Product Design and Development, McGraw-Hill, New York, Second Edition, 2000.

[11] Eppinger, S.D. "A Planning Method for Integration of Large-Scale Engineering Systems", International Conference on Engineering Design, Tampere, Finland, August 1997, pp. 199-204.

[12] Eppinger, S.D. "Innovation at the Speed of Information", Harvard Business Review, vol. 79, no. 1, pp. 149-158, January 2001.

[13] Sosa, M.E., Eppinger, S.D., and Rowles C.M. "Designing Modular and Integrative Systems", ASME Conference on Design Theory and Methodology, Baltimore, MD, September 2000.

[14] Morelli, M.D., Eppinger, S.D., and Gulati, R.K. "Predicting Technical Communications in Product Development Organizations", IEEE Transactions on Engineering Management. vol. 42, no. 3, pp. 215-222, August 1995.

[15] Salminen V., and Pillai B. "Strategic Management of Adaptive, Distributed Product Development of a Mechatronic Product", International Conference on Machine Automation, Osaka, Japan, September 2000.

**Contact Address**

Prof. Steven D. Eppinger and Mr. Vesa Salminen
Massachusetts Institute of Technology
Center for Innovation in Product Development
Cambridge, MA 02142-1347 USA
Email: eppinger@mit.edu vesas@mit.edu