

A PLANNING METHOD FOR INTEGRATION OF LARGE-SCALE ENGINEERING SYSTEMS

Steven D. Eppinger

Design Process Management, System Engineering, Design Structure Matrix

1 Introduction

Complex products and large engineering systems require massive efforts to integrate designs at various levels: product and production plans into components, components into sub-systems, sub-systems into systems, and systems into a quality product. We define the product development integration problem as the challenge to coordinate the engineering activities at each level in order to achieve an integrated system in the final product. We have developed a three-step method to plan solutions to such integration challenges. The first step documents the decomposition of the system into components. Second, the interactions between the components are identified using methods based on various types of interactions. Finally, the components are clustered into systems around the integration challenges defined by the network of interactions.

2 Decomposition and Integration

In order to develop a complex product or large engineering system, it is common practice to decompose the design problem into smaller sub-problems which can be handled more easily. If any of the sub-problems are still too complex, they may in turn be further decomposed. Development teams are assigned to each design problem which may represent a component or sub-system of the larger system.

One important level of integration takes place within each development team. This is the now common practice of concurrent engineering, in which a cross-functional team addresses the many design and production concerns simultaneously. However, to assure that the entire system works together, the many sub-system development teams must work together. This latter form of integration is often called system engineering. Figure 1 graphically depicts the relationship of problem decomposition and system integration.

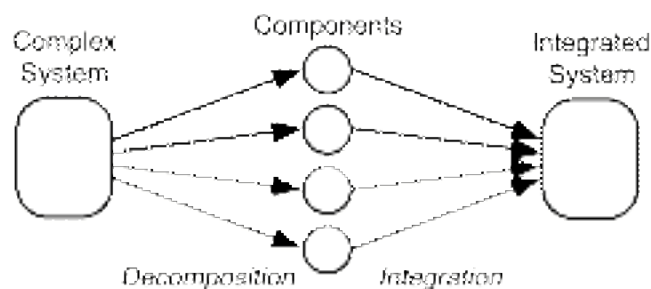


Figure 1. The decomposition and integration problems

The premise of this paper is that the complexity of system integration is related to the pattern of the relationships between the components. Indeed it is these interactions which system engineering must be planned to address. Several scholars have proposed that the network of task interdependency be considered in planning for system integration within development projects [Alexander 1964; Simon 1970; von Hippel 1990]. Engineering design researchers have recently been exploring methods for decomposition of design activities based on the interdependencies of the design parameters involved in each [Altus, et al. 1995; Kusiak and Wang 1993]. Many authors have recommended that problem decomposition be carried out so as to minimize interactions between the sub-problems. Unfortunately, this is rarely possible for large system decomposition because practitioners may be limited to historical, familiar approaches, or may be partially constrained to utilize existing sub-systems and components.

In this research, we assume that the designers have already developed such a decomposition. This can be accomplished for novel design problems using formal methods for functional decomposition [Pahl and Beitz 1984]. For more routine development problems, existing sub-system or component decompositions suffice. In the case of functional decomposition, many of the (fundamental) interactions between the components would be documented on the function diagram in the form of energy, materials, or signal interchanges between the sub-functions. Still other (incidental) interactions will arise due to the embodiment of the functions as physical components [Ulrich and Eppinger 1995]. In the case of more familiar design problems, the interactions between the sub-systems and components may be well known to the design teams and can be documented using methods explained below.

We use a mapping of the physical interactions between the components to represent the structure of the system integration problem. In a similar manner, we can alternatively use a mapping of the personal communications between the people who design the components to also capture the structure of the integration problem for the system as a whole. This paper extends our prior research in which we have shown how to model design procedures using the design structure matrix approach [Eppinger, et al. 1994], and in which we began to explore integration analysis of architectures and organizations [McCord and Eppinger 1993; Pimmler and Eppinger 1994].

3 Analysis Method

The integration analysis method is comprised of three steps:

Step 1. Record the problem decomposition.

Step 2. Document the interactions in a matrix.

Step 3. Analyze the matrix to identify the structure of the interactions.

To record the problem decomposition, we create a list of the components or of the teams responsible for designing the components. To document the interactions, we have had success with a scheme in which we ask the engineers directly how each of the components interact, using the four-attribute scale shown in Figure 2a. We have been able to model larger problems by surveying the organization regarding frequency of interaction between the people on the development teams, using the single-attribute scale shown in Figure 2b.

The interaction data are represented in graphical form using a matrix. The idea of using a matrix is derived from our earlier experience using the design structure matrix (DSM). However, this particular application features important differences not only in the type of DSM data we include, but also in the type of analysis we perform. Using the interaction scales given above, the interaction matrix becomes quite symmetrical, which means that the type of sequencing analysis common for DSM is not possible. Instead, we perform a type of clustering analysis.

The components are clustered into systems around the integration challenges defined by the network of interactions. We have explored many types of clustering algorithms [Romesburg 1984] and have found that typical algorithms assign elements into mutually exclusive groups. In the case of our integration analysis, we wish to allow overlapping clusters with cross

membership where appropriate. In the work presented here, we used heuristic clustering approaches implemented by Excel macros and C programs. We continue to develop improved clustering algorithms suitable for this problem.

	Detrimental	Undesired	Indifferent	Desired	Required
Spatial	-2	-1	0	+1	+2
Energy	-2	-1	0	+1	+2
Information	-2	-1	0	+1	+2
Material	-2	-1	0	+1	+2

Figure 2a. Scale used to represent component interactions

	Daily	Weekly	Monthly	None
Frequency of Interaction	●	●	•	

Figure 2b. Scale used to represent personal interactions

4 Applications and Results

We have developed and tested these methods in conjunction with projects in the automotive industry. Two examples will be presented here briefly.

4.1 Powertrain Design Example

In research with General Motors Powertrain Division, we studied the design of a new engine. (Additional details can be found in a working paper [McCord and Eppinger 1993].) The development project was decomposed into 22 major components (also some larger sub-systems) and a product development team (PDT) was assigned to each. Based on the cross-functional nature of each of the 22 PDTs, concurrent engineering was possible for each component. The concern of our analysis is how to assure integration of the components as a complete system (the entire engine). Using the survey method shown in Figure 2b, we assessed the necessary interactions across the PDTs, which are documented in Figure 3a.

The team leaders had a plan for the system-level integration of the development results. This involved assigning each of the PDTs to a system team according to the clustering scheme shown in Figure 3b. Inspection of this exclusive clustering suggests that many of the interactions across the teams would not be facilitated by the system team structure and that a more effective clustering may be found.

Our analysis suggests the alternative assignment of system teams shown in Figure 3c. By allowing the system teams to have cross membership, we achieve a structure which facilitates the more efficient exchange of information between the development teams. Note that it is the overall problem structure revealed by the data analysis which determines the assignments of PDTs to system teams. For example, the cylinder head and intake manifold PDTs (B and K) are assigned to three system teams (2, 3, and 4). This is due to their interactions with each of the other PDTs and to the overall pattern of interactions which defined the system teams.

Furthermore, there remain five PDTs which are tightly coupled to so many of the other PDTs, that we assigned these to an “integration team”. This is to acknowledge their special role as system integrators across the entire project, for which a more focused system team assignment may not be appropriate.

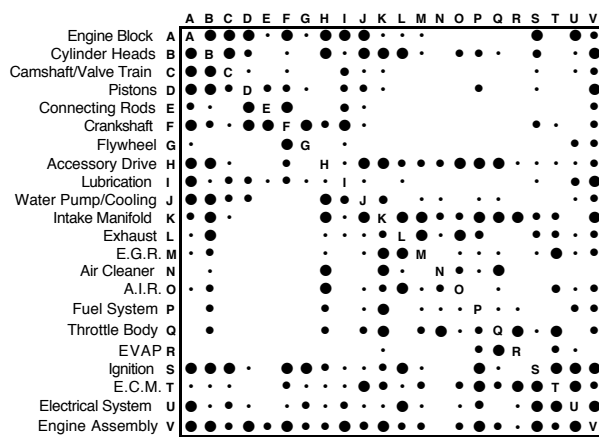


Figure 3a. Powertrain PDT interactions

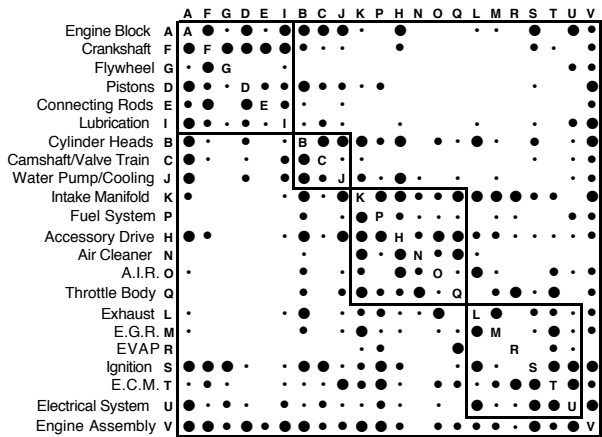


Figure 3b. Existing assignment of system teams

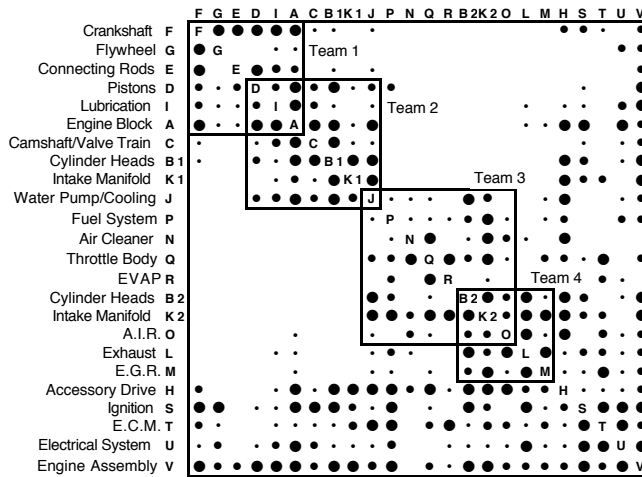


Figure 3c. Suggested system team structure

This example clearly illustrates the need for three distinct levels of integration within the project. First, there are design, production, and other concerns which must be integrated by each component-focused cross-functional team. Second, there are specific system-level performance issues to be addressed by the system engineering teams. Finally, there are system-wide performance issues which must be addressed by a system integration team with a broader purview than the smaller system engineering teams.

4.2 Climate Control System Example

As the powertrain application showed that the integration analysis method can assist engineering managers in assigning system team responsibilities, we looked next to test how the method applies to the analysis of product architectures. This example involves the design of an automotive climate control system at Ford Motor Company and is more fully documented in a paper with Pimmler [Pimmler and Eppinger 1994].

To conduct the integration analysis of the architecture of the climate control system, we documented the interactions between components using the scheme shown in Figure 2a, which was accomplished by interviewing the climate control engineers. Since there are four dimensions to the interaction scale (spatial, energy, information, and material), the clustering can be carried out on any of the four dimensions or on any combination thereof. We found that each dimension offered an important perspective, and that these could be combined in the final clustering display of the interaction matrix, shown in Figure 4.

This analysis revealed four physical chunks in the climate control system architecture. It suggests the opportunity for certain types of modular design and identifies the need for system engineering within three of the chunks and across all four chunks.

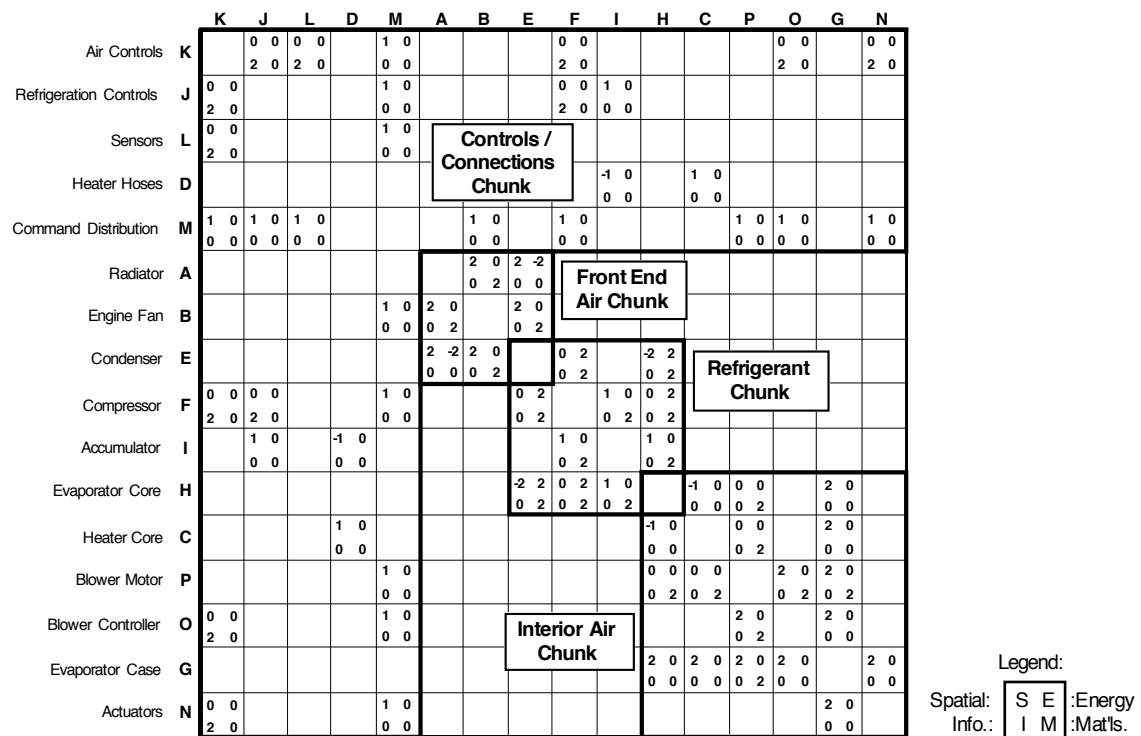


Figure 4. Integration analysis of the climate control system architecture

5 Discussion and Conclusion

In this stream of research, we have explored the analysis of large-scale engineering systems from the perspective of difficulty to integrate the separately designed components into working systems. We base the analysis upon data representing the functional interactions between each of the components or upon the personal interactions between the individuals or teams developing each of the components.

The proposed method is certainly limited in a number of ways. One important restriction is the use of engineering interviews or surveys as the source of the interaction data. It might be preferable to utilize detailed engineering models which can reveal the technical interactions; however, we have not studied problems with models rich enough to include the incidental interactions arising from the many subtle effects which must be handled in integration. Another restriction is the decomposition itself. Since our analysis begins with a problem decomposed into components, the very definition of the components gives rise to many of the functional interactions. If alternative decompositions were considered, the analysis could be enhanced.

Nevertheless, the integration analysis has very practical impact and teaches important lessons about both integration and decomposition. The resulting models allow for efficient, data-driven planning of system-integration team assignments and the critique of product architectures in terms of integration complexity. When clusters of highly interactive components comprise physical chunks, sub-systems, or modules within the product architecture, with few well-understood interactions between the chunks, this suggests a robust modular architecture with a corresponding efficient development process. Similarly, when clusters of highly interacting people form system teams to handle the interactions between their components, an efficient development organization is suggested.

This line of research suggests several directions for future work. In particular, we are investigating three areas: 1. exploration of the duality of the product architecture and the organizational architecture; 2. analysis of dense clusters of system-level interactions using a richer vector of attributes in problem space for each interaction; and 3. development of efficient and flexible clustering analysis algorithms.

References:

- Alexander, C., Notes on the Synthesis of Form, Harvard University Press, Cambridge, MA, 1964.
- Altus, S.S., Kroo, I.M., and Gage, P.J., "A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems", ASME Design Automation Conference, Boston, MA, September 1995.
- 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, 1994, pp. 1-13.
- Kusiak, A. and Wang, J., "Decomposition of the Design Process", Journal of Mechanical Design, vol. 115, December 1993, pp. 687-695.
- McCord, K.R. and Eppinger, S.D., "Managing the Integration Problem in Concurrent Engineering", MIT Sloan School of Management Working Paper, no. 3594, August 1993.
- Pahl, G. and Beitz, W., Engineering Design, Springer-Verlag, New York, 1984.
- Pimmler, T.U. and Eppinger, S.D., "Integration Analysis of Product Decompositions", ASME Conference on Design Theory and Methodology, Minneapolis, MN, September 1994, pp. 343-351.
- Romesburg, H.C., Cluster Analysis for Researchers, Krieger Publishing Co., Malabar, FL, 1984.
- Simon, H.A., The Sciences of the Artificial, MIT Press, Cambridge, Massachusetts, 1970.
- Ulrich, K.T. and Eppinger, S.D., Product Design and Development, McGraw Hill, New York, 1995.
- von Hippel, E., "Task Partitioning: An Innovation Process Variable", Research Policy, vol. 19, 1990, pp. 407-418.

Acknowledgments:

Funding for this research has been provided by the MIT Leaders for Manufacturing Program and the MIT International Center for Research on the Management of Technology, whose member companies have also provided access to the industrial examples presented here. The author is indebted to a team of graduate students who, over several years, have assisted in developing the concepts and examples presented here. Researchers wishing to explore this work in more depth may request additional references by contacting the author.

Contact Address:

Prof. Steven D. Eppinger
Massachusetts Institute of Technology
Sloan School of Management
Cambridge, MA 02142-1347
USA
tel.: +1-617-253-0468
email: eppinger@mit.edu