Modularity of the Software Industry: A Model for the Use of Standards and Alternative Coordination Mechanisms

Heiko Hahn, University of the Federal Armed Forces Munich, Germany
Klaus Turowski, University of Augsburg, Germany

ABSTRACT

The purpose of this paper is to develop a research model for the driver and inhibitors to a modular software industry. A modular software industry is characterized by specialized software component producers instead of highly integrated firms that offer monolithic software systems. The research model we present is a synthesis of different and sometimes competing research traditions (e.g., system theory, transaction cost theory, and the resource-based view). Besides the theoretical foundation, the research model also incorporates insights from the analysis of 11 in-depth interviews that were conducted with experts from the automotive, IT, finance, and insurance industries.

Keywords: business components; component based software engineering; component markets; standardization

INTRODUCTION

Products often consist of different parts; that is, they are product systems consisting of different components rather than monolithic products. The parts are, to a varying degree, independent and interchangeable, and the products might be produced by one integrated firm or by different firms. In the latter case, the components can be purchased prefabricated on the market or from a supplier with whom the customer has an intensive business relationship and who produces the parts according to the special requirements of the customer. Before a market for components can evolve, however, it is necessary that a product architecture with documented interfaces exists that allows the flexible combination of components and the adaptation of the systems to different needs of the customer. For example, in the case of business application systems, such a market would allow a company that needs new software for stock keeping to buy a suitable software component from the component market and further integrate it into its business application system with little effort.

Both the idea of software systems made up of components and markets for software components have been discussed since the late 1960s (McIlroy, 1976). However, despite the advantage of component-based software systems and the success of component-based systems in other industries (e.g., the platform strategies in the automotive industry), markets for software components hardly can be found (Dietzsch & Esswein, 2001).
The remainder of this paper is as follows. After the discussion of modularity and business components, we develop a conceptual model that represents the different factors that influence interfirm market modularity. The paper focuses on the interfirm market modularity in the software industry.

MODULAR DESIGN AND THE PRINCIPLE OF INFORMATION HIDING

The concept of modular design is central to system theory. Simon characterizes nearly decomposable (i.e., modular systems) as follows: “(1) in a nearly decomposable system the short run behaviour of each of the component subsystems is approximately independent of the short-run behaviour of other components; (2) in the long run the behavior of any one of the components depends in only an aggregate way on the behaviour of the other components” (Simon, 1962, p. 476).

For technical systems, the influence that different components have on each other is limited and mediated by interfaces that allow abstracting from the lower-level implementation. The interface is the device that mediates the interaction between the different components and prevents a direct communication. Parnas (1972) introduced the principle of information hiding that emphasizes the importance of a non-direct communication and of a black box approach. According to the principle of information hiding, the design decision should be led by the goal to make the different parts of a software program as independently changeable as possible, so that the system can accommodate changes easily. Information hiding provides a criterion for the decoupling of the different systems: The interface level that connects the different parts of which the system consists should be invariant against changes at the implementation level. To reach this goal, the system has to be separated into parts that are stable (i.e., interface level) and those that are likely to change (i.e., implementation level). The implementation of the modules should be hidden from the other parts of the system as much as possible (i.e., it should be a black box to other components to prevent a designer of a module from creating interdependencies by referring to specific aspects of the implementation of other modules that are likely to change. The economic value of this principle has been elaborated and tested, based on simulations by Baldwin and Clark (2000), who introduced so-called modular operators that can be deduced from a modular architecture. Inter alias, they analyzed the value of the possibility of splitting up a system into modules, to substitute different modules for another, or to augment the functionality.

Besides the decoupling of the different subsystems, there is also a second important property in the theory of nearly decomposable systems. The structure of these systems is hierarchical, which means that the components are part of other components that are themselves part of larger components (Simon, 1962). For software systems, components will be part of frameworks that can be considered as high-level components (Szyperski, 2002; Turowski, 2000). A module, therefore, is not only interchangeable and can be substituted for a module that fulfills the same interface specification, but it is also part of a modular architecture that is created by the horizontal (i.e., same level of abstraction) and vertical composition of components (i.e., composition inside the black box consisting of lower-level components).

As a conclusion, modularity can be defined as the degree to which a system consists of only loosely coupled parts that can be separated and independently exchanged (Schilling, 2000).

INTERFIRM MARKET MODULARITY AND BUSINESS COMPONENTS

The dependent variable of our model (Figure 1) is interfirm market modularity. It can be defined first as the degree to which the components of different firms are interchangeable and second by the degree to which the coordination is market-based. Interfirm market modularity is also a measure for the division of labor.
Grid Data Handling


[www.igi-global.com/chapter/grid-data-handling/75035?camid=4v1a](http://www.igi-global.com/chapter/grid-data-handling/75035?camid=4v1a)