COMMONALITY VERSUS VARIABILITY
The Contradictory Nature of Enterprise Systems

Stig Nordheim
Agder University College, P.O Box 422, N-4604 Kristiansand, Norway

Keywords: Commonality, Variability, Enterprise Systems, Dialectics.

Abstract: This position paper argues that there is a major contradiction inherent in Enterprise Systems. The evidence for this contradiction is seen in the meta level concepts of commonality and variability that characterize Enterprise Systems. The inherent contradiction between commonality and variability is discussed in the light of Enterprise Systems' literature, and the contradiction is elaborated. Then an Enterprise Systems' vendor perspective on commonality and variability is presented, based on interviews. A challenge of this inherent contradiction is its synthesis, finding the right balance between commonality and variability.

1 INTRODUCTION

Enterprise Systems (ES) are commercial software packages that enable the integration of transaction-oriented data and business processes throughout an organization (Markus and Tanis, 2000). Examples of ES include: Enterprise Resource Planning, Enterprise Content Management, Supply-Chain Management, Customer Relations Management and Data Warehousing.

This paper argues that there is a major contradiction inherent in ES. A contradiction is a relationship between two opposite aspects of a phenomenon. The evidence for asserting that there is an inherent contradiction in ES is presented by means of the meta level concepts of commonality and variability that characterize ES. In the following, the meta level concepts of commonality and variability are discussed in the light of the ES literature, and then the concept of contradiction is presented. A system level contradiction that is asserted to be inherent in ES is then presented, and the contradiction is illustrated by interview data from an ES vendor.

2 COMMONALITY AND VARIABILITY

Enterprise systems (ES) are configurable systems, based on the concepts of "commonality" and "variability" as discussed by Leishman (1999). These concepts are related to fit and alterability. The concepts are based on understanding the commonality and variability (c/v) across industries, geographies, customers, and systems.

The general meaning of commonality is "possession of common (i.e. widespread or general) features or attributes" (Merriam-Webster, 1996). In software engineering a commonality is seen as "an assumption held uniformly across a given set of objects (S)" (Coplien et al., 1998), and commonality is represented by the software code. Variability is generally understood as "subject to variation or changes" (Merriam-Webster, 1996). Variability in the context of software engineering is seen as "an assumption true of only some elements of S" (Coplien et al., 1998.), and is implemented in different ways. In this context, variabilities are "bound" by placing specific limits on each of the variabilities. One example of bounded variability may be a range of legal values for a parameter (Coplien et al., 1998).

By providing variability in the ES package the vendor anticipates that the customer has a certain need for flexibility. The scope of the variability is typically predefined and with a set of options. Parameter configuration is one example of predefined variability, and black-box customization of frameworks is another. This concept of a predefined variability fits with the software engineering perspective on variability (Coplien et al., 1998), that variability is bounded.

Based on a review of different types of systems, Leishman (1999) identifies a number of generic mechanisms available for implementing variability.

Nordheim S. (2006).
COMMONALITY VERSUS VARIABILITY - The Contradictory Nature of Enterprise Systems.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 572-575
DOI: 10.5220/0002469505720575
Copyright © SciTePress

Variability here applies both to extensions of the system and integration with other applications. This variability may be implemented by mechanisms such as configuration, subclassing and inheritance, specialization of patterns, multiple versions of components, instantiation of abstract classes, parameters, templates, DDLs, customer exits, import/export mechanisms, adapters and connectors, install scripts, registry, property sheets and customizers (Leishman, 1999).

Another set of variability mechanisms is presented by Brehm et al. (2001), who present a typology of Enterprise Resource Planning (ERP) adaptation in practice. These ERP variability mechanisms include configuration, bolt-ons for implementation of third-party packages, screen masks, workflow programming, user exits for programming additional software code in an open interface, ERP programming using the programming language provided by the vendor, interface development, and in rare instances some package code modification.

A special case of variability is found with Open Source software, where the commonality itself, the source code, also becomes subject to variability. But even in this case some vendors have a kernel of commonality, e.g. the enterprise content management vendor eZ Systems (http://ez.no/).

3 CONTRADICTIONS

To explain the contradictory nature of commonality and variability, we need to clarify the concepts related to contradictions. A contradiction can be seen as a relation between two opposite aspects of a phenomenon. One aspect in a contradiction (also called a contradistinction) cannot be fully understood without considering the other aspect. The two aspects of a contradiction are called thesis and antithesis, where antithesis is the negation of the thesis. So the two aspects (or contradistinctions) are intrinsically related, yet opposite and distinct from one another (Mathiassen and Nielsen, 1989; 1990). The combination of thesis and antithesis may lead to a synthesis, which may be different from both thesis and antithesis.

A thesis (A) may be challenged by an antithesis (Not-A), and the resolution of the conflict becomes a synthesis (which is Not Not-A). By its very nature, the synthesis is a novel construction that departs from both the thesis and the antithesis. Over time, this synthesis may become a new thesis as the dialectical process continues (Van de Ven and Poole, 1995).

Analysing contradictions is also referred to as dialectics, where the dialectical tension is the opposition between two interacting forces or elements. The main point in dialectical reflection is to find the principal contradiction of a process in order to understand a situation (Bjerknes, 1992).

According to Dahlbom and Mathiassen (1993), contradictions can in some cases be seen as trade-offs: “When working with computers … we are typically faced with what is traditionally called trade-offs. From a dialectical perspective, these trade-offs are manifestations of contradictions inherently related to the use and development of computer systems” (p63). According to Israel (1979) dialectics contributes to the production of knowledge, by an increased understanding of a phenomenon. For the purpose of this paper, we note one dialectic to be of particular interest here, namely the age-old contradiction between stability and change (Lewis, 2000).

4 THE INHERENT CONTRADICTION OF ENTERPRISE SYSTEMS

The mechanisms for commonality and variability built into Enterprise Systems (ES), may also be seen as representing a contradiction (cf. Figure 1). Viewed as a thesis, commonality assumes that the system properties involved should not and cannot be changed. A rationale for this is that the system is based on best practices, and therefore it should not be changed. This argument is used for ERP systems. An ES following this thesis to the extreme, would simply be installed with no configuration whatsoever, as it would only consist of commonality. A pure thesis is not a feasible solution. Viewed as an antithesis, variability assumes that the system properties involved can and probably should be changed. A rationale for this is that the organizational contexts where the system is installed, may be different, and therefore the system needs to have its properties set or changed. An ES following this antithesis to the extreme, would only consist of variability and have no commonality at all. So a pure thesis is not a feasible solution either.

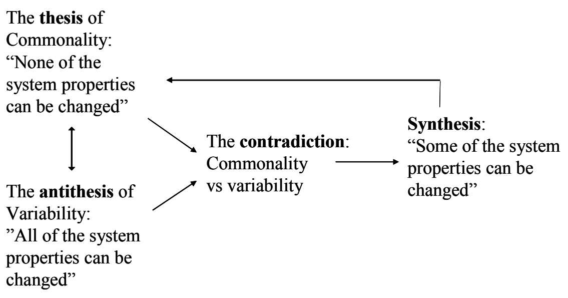
The **thesis** of Commonality: "None of the system properties can be changed"
The **antithesis** of Variability: "All of the system properties can be changed"
The **contradiction**: Commonality vs variability
Synthesis: "Some of the system properties can be changed"

Figure 1: The inherent contradiction of Enterprise Systems.

The contradiction itself is about the fixed versus the variable, or stability versus change (Lewis, 2000). Commonality versus variability may therefore be seen as an inherent dialectic within ES, expressing the contradictory nature of ES. These design trade-offs are indeed manifestations of contradictions inherently related to the development of the systems (cf. Dahlbom and Mathiassen, 1993). Because of its inherent commonality/variability architecture, an ES may therefore be seen as the embodiment of a contradiction that is fundamental to the nature of ES. This contradiction is viewed at a system level and from a vendor perspective.

The synthesis for most ES vendors implies finding an optimal balance between commonality and variability, i.e. between stability and change. This balance means that a subset of the properties of the system may be changed. Different vendors will reach very different decisions about the right balance between commonality and variability (Davenport, 1998).

A puzzling phenomenon is that some ES vendors provide powerful customization tools, while emphasizing the importance of limited configuration. So the vendor's attitude to stability and change may even be contradictory.

5 A VENDOR PERSPECTIVE ON THE CONTRADICTION OF ENTERPRISE SYSTEMS

To illustrate the inherent contradiction presented above, the results of interviews with two representatives from an ES vendor are presented in the following.

The case studied is eZ Systems (http://ez.no/), an Open Source enterprise content management (ECM) vendor with 30-40 thousand downloads per month. One of the informants interviewed has a split role between management of customer projects and programming, and the other informant is responsible for all customer projects in the company.

The idea behind eZ Systems is flexible solutions rather than "off-the-shelf" software. So they emphasize the antithesis of Fig 1. Their customers are extremely heterogeneous. As this is an Open Source software, most customers download and implement it without any contact with the vendor.

An interesting point is that even with this Open Source enterprise system, there is commonality. The commonality is in this case a kernel that is developed and controlled exclusively by eZ Systems.

There are two types of variability provided by eZ Systems, configuration and programming. Configuration is referred to as a supported variability, and programming is referred to as an unsupported variability. In addition to configuration, supported variability also consists of modifying templates, e.g. HTML templates. Unsupported variability consists of interface programming, usually based on existing libraries. So a considerable variability is achieved by plug-ins into the kernel, and as the system evolves, plug-ins are extended without affecting the kernel. Consultancy on variability ranges from adapting templates for small businesses, to complex integration with legacy systems in large enterprises.

The line of demarcation between commonality and variability is by eZ Systems perceived as an interesting design trade-off. So from this vendor's perspective, the contradictory nature of the ES is recognized, and an optimal synthesis is seen as a design challenge. As pointed out by one of the informants, if there is too much commonality compared to variability, the customer is being locked up due to lack of configuration options. But if there is too much variability compared to commonality, the customer will be confused due to the lack of standard functionality. According to the informants at eZ Systems, an implementation based on as much commonality as possible means better maintainability, higher quality and a cheaper solution.

The vendor's goal is in this case to have as much configuration as possible, and as little programming as possible. The ideal is to empower non-programmers to establish complex solutions by simple configuration. A goal is to cover 95% of a customer's needs by configuration. This can therefore be seen as eZ Systems' view of an ideal synthesis (c.f. Fig 1). The vendor also tries to influence the customer's requirements on to what is configurable. Configuration constitutes the typical effort, together with modifying HTML templates. Functional adaptation and user interfaces are usually adapted through configuration. In a typical project, programming may constitute 10% of the work, whereas 90% of the effort is configuration and template adaptation. When programming occurs, typical programming tasks include: integration with back-end systems, including the integration of user administration and access rights. Special processes for document approval within the implementing organization may also require some programming. There is very little programming in a typical project, but the extent of programming varies enormously. Larger projects are characterized by some programming, typically modifying scripts to handle import/export. In some larger projects there may also be some programming to create new functionality, based on existing libraries.

6 CONCLUSIONS

Most of the practice literature referred in this paper is from an ERP context, since ERP systems is a prime example of Enterprise Systems. ERP systems constituted only one of the six example systems that the vendor literature (Leishman, 1999) was based on. By examining another type of ES in the case of eZ Systems, it was found that the categorizations of commonality and variability were similar to those of ERP. The ECM vendor eZ Systems followed categorizations similar to ERP systems, and both are ES.

We argue that Enterprise systems by nature are contradictory. They embody an inherent contradiction at the system level. This is evident from the concepts of commonality and variability. The vendor challenge as seen in the case presented, consists of finding an optimal synthesis. The synthesis is a good balance between commonality and variability. As part of this, an apparent vendor goal is to provide sufficient variability by means of configuration.

For ES customers the inherent contradiction of these systems also may give rise to another question: What are the possible implications of this contradiction for organizations that are implementing ES?

REFERENCES

Brehm, L., Heinzl, A., Markus, M.L., Tailoring ERP Systems: A Spectrum of Choices and their Implications. *Proceedings of the 34th Hawaii International Conference on Systems Sciences*, Hawaii, 2001.

Coplien, J., Hoffman, D., and Weiss, D., Commonality and Variability in Software Engineering. *IEEE Software*, 1998. 15(6): p. 37-45.

Dahlbom, B. and L. Mathiassen (1993). *Computers in Context : The philosophy and practice of systems design*. Cambridge, Mass., NCC Blackwell.

Davenport, T. H. (1998). "Putting the Enterprise into the Enterprise System." *Harvard Business Review* 76(4): 122-131.

http://ez.no/

Israel, J. (1979). *The Language of Dialectics and the Dialectics of Language.* Copenhagen, Munksgaard.

Lewis, M. W. (2000). "Exploring Paradox: Toward A More Comprehensive Guide." *Academy of Management Review* 25(4): 760-776.

Leishman, D.A., Solution Customization. *IBM Systems Journal*, 38,1 (1999) 76-97.

Markus, M. L. and C. Tanis (2000). The Enterprise System Experience - From Adoption to Success. In *Framing the domains of IT management : projecting the future through the past*. R. W. Zmud (ed.) Cincinnati, Ohio, Pinnaflex Education Resources: 173-207.

Mathiassen, L. and P. A. Nielsen (1989). "Soft Systems and Hard Contradictions - Approaching the Reality of Informations Systems in Organizations." *Journal of Applied Systems Analysis* 16: 75-88.

Mathiassen, L. and P. A. Nielsen (1990). Surfacing Organizational Competence. Soft Systems and Hard Contradictions. In *Organizational Competence in System Development: a Scandinavian Contribution*. G. Bjerknes (ed.) Lund, Studentlitteratur.

Merriam-Webster's collegiate dictionary. 10th ed. 1996: Merriam-Webster, Springfield.

Van de Ven, A. H. and M. S. Poole (1995). "Explaining development and change in organizations." *Academy of Management Review* 20(3): 510-540.