Normalized Systems

Prof. dr. Herwig Mannaert
Prof. dr. Jan Verelst
Department of Management Information Systems
University of Antwerp
{herwig.mannaert,jan.verelst}@uantwerpen.be

The term Normalized Systems refers to a theoretical framework for the development of highly evolvable information systems described in the book "Normalized Systems: Re-creating Information Technology Based on Laws for Software Evolvability", Koppa, 2009. This text contains a brief summary of the framework.

Introduction

In the first decade of the 21st century, organizations are operating in hypercompetitive environments, constantly monitoring their environment for new business opportunities and striving for customer satisfaction by delivering products and services of unprecedented quality. This has resulted in enterprises that are faced with challenges such as *increasing complexity* and *increasing change*. These challenges require high evolvability, agility or flexibility of the organization and its information systems. However, current methodologies for the development of information systems struggle to meet this demand for flexibility.

Part 1: Principles

The theoretical framework of Normalized Systems is based on an analysis of the modular structure of software architectures of information systems. This modular structure consists of so-called constructs in programming languages such as procedures, functions, classes, services and most recently, aspects. The framework consists firstly of 4 principles, which indicate when so-called *combinatorial effects* occur in a modular structure. A combinatorial effect exists when the size of the impact of a change to a software architecture is dependent on the size of the information system. Combinatorial effects therefore represent a particularly harmful kind of coupling in the software architecture, as such effects cause information systems to become increasingly difficult to maintain during their life cycle, until they cannot be maintained any more in a cost effective way and ultimately have to be replaced by an information system with similar functionality. In this sense, the principles explain the Law of Increasing Complexity as formulated by Lehman (see, e.g., http://en.wikipedia.org/wiki/Lehman's_laws_of_software_evolution). In other words, combinatorial effects explain *why current software architectures are inherently limited in their flexibility*. Furthermore, they also have highly a negative effect on other quality factors such as reusability.

Part 2: Elements

The goal is therefore to build information systems free from combinatorial effects. To that end, the theoretical framework contains 5 elements, with which the basic functionality of virtually all information systems can be built. These elements are:

- a *data element* for storing data,
- an *action element* for the execution of a calculation or algorithm,
- a *workflow element* for the execution of sequences of action elements,
- a *connector element* for input and output functionality, and
- a *trigger element* for time- or status-based execution of action elements.

An application, then, can be built consisting of *N* instances of these elements. These instances are parameterized copies of the 5 elements, and can therefore be built using a kind of code generation, which we call 'element expansion'. As it can be proven that these elements do not contain combinatorial effects, it is also guaranteed that the resulting application is free from combinatorial effects, and therefore guaranteed to be more evolvable and reusable than current information systems.

The Normalized Systems theoretical framework applies to any system consisting of modular structures and is therefore completely independent of specific programming languages and the use of specific packages or frameworks. Hence, the elements can in principle be built in any combination of technologies (including programming languages, packages and frameworks). The essence of the framework remains that certain 'errors against evolvability', i.e. combinatorial effects, systematically need to be removed from modular structures which results in '*evolvable modularity',* irrespective of the particular language or framework being used.

Applications and Research

Mission-critical Normalized Systems have been built for a wide variety of applications to illustrate the genericity of the theory. Additionally, PhD-research on Normalized Systems is performed at the department of Management Information Systems of the University of Antwerp. This research aims at, for example, advancing the theoretical framework at the software architecture level, as well as the application of the Normalized Systems principles to business processes and organizations. The goal is to strive for '*evolvable modular'* organizations, which can satisfy the ever growing demand for flexibility.

More information

Available at: http://en.wikipedia.org/wiki/Normalized_Systems