Tailorable Information Systems

D. S. Stamoulis
National and Kapodistrian University of Athens, Greece

D. Theotokis
University of Peloponnese, Greece

D. I. Martakos
National and Kapodistrian University of Athens, Greece

INTRODUCTION

The quest for the business fit of information systems is as old as the history of computers. In the past, computers used to lag behind the daily needs of the users because they lacked the capabilities of tackling the issues raised by the users. In the late 80's, the fourth generation of computers was a decisive step toward the acquisition of such system capabilities that allowed powerful software to be developed (O'Brien, 2002), rendering them able to tackle almost any business problem.

However, the previous decade has been marked by failures of big information systems (IS) projects, despite the sound accumulated knowledge on various information systems development methodologies and the advent of software engineering tools. The examples of IS that failed to deliver their premise are abundant, to name but a few: the U.S. IRS tax system modernization (Hughes & Cotterell, 2002); the London Ambulance Service's information system and the UK automated stock brokerage system TAURUS (Beynon-Davis, 1995); the automated train seats' reservation system SOCRATE in France (Eglizeau, Frey, & Newman, 1996). Of course, a variety of reasons could lead an IS project to a disaster, and the sources of the failure could stem from project management malpractice to poor software engineering techniques, to wrong leadership choices, and so forth. Apart from them, research has shown (e.g., Fitzgerald, 1990; Oei, Proper, & Falkenberg, 1994; Robertson, 1997) that any IS, no matter how "perfect" it can be built, is immediately being undermined for the forces of change. These forces of change inherently exist in any organization and have to do not only with the constant expansion or modification of the user requirements day-to-day, but also with the user's computer and information literacy. The latter means that the familiarization of the user with an IS creates new needs that IS maintenance comes to respond to. Maintenance is a corrective action, aiming at endowing the IS with improved functionality, in contrary to the irrational inference that a "finished" IS product is supposed to operate in the same business/organizational context meeting always to the same set of initial requirements (Bjerknes, Bratteteig, & Espeseth, 1991). No matter how precise and complete the elicitation of user requirements may be, the behaviour of an IS is constantly diverging from the evolving needs of the business/operational context (Stamoulis, Martakos, & Introna, 1998).

BACKGROUND

All methods and techniques used so far to produce flexible enough IS which can achieve business fit have all fallen victims of the fixed-point pseudo-theorem (Paul, 1993): "There exists some point in time when everyone involved in the system knows what they want and agrees with everyone else." This statement can never be verified in practice, and subsequently, describes a dead-end situation. User requirements are the result of the evolving business/organizational context of the IS, whereas the IS itself has been built to an exact specification which had been based on a snapshot of its business context's operation. Ideally, IS should behave like living organisms (Paul, 1993) that have the ability to adapt to changes of their operating environment. But, the software engineering technologies along with the way of thinking of the IS development discipline cannot escape the destiny of their determinism. This is where ateleology and tailorable IS (TIS) come into play. The former negates the mentality of constructing an IS as a product developed to meet a specific telos (purpose, definition) and the latter provides the theoretical framework for a new breed of truly flexible IS to appear. The questions of whether TIS are feasible and if software can even be tailorable are answered by means of the Atoma architecture, which has already been implemented.

MAIN THRUST OF THE ARTICLE

Tailorability is not another word for customization. IS flexibility has been based for years on how many parameters of the IS functionality can be configured by the end-user without taking any maintenance action. The tailorable IS thinking immediately focused on adaptivity. A program is adaptive when it changes its behaviour depending on its current contextual situation (Gouda & Herman, 1989). If everything in an IS may change, then tailorability affects the user interface in terms of the user dialogue, the screen layout, the help function, the icons and the functions themselves, as well as the functionality in terms of input, processing and output (Patel, Gardner, & Paul, 1995). Of course, this flexibility should be realized in real-time, without disrupting the IS operation, upon user's demand. If all of these features of an IS could change, then IS would really behave like a living system, and then it could be named as a tailorable IS (Patel et al., 1995) whose purpose is to "provide control over systems to a user". Looking at the problem at the macro level, an abstract systems architecture model has been proposed (Stamoulis, Patel, & Martakos, 1996), which describes the need for tailorability at three software layers: IT infrastructure (hardware, operating system, data sources); application logic (workflow, process logic, and facilities layers); and presentation logic layer. For each one of these layers, emerging software technologies that demonstrate tailorability traits or characteristics have been identified (Stamoulis, Kanellis, & Martakos, 2000).

In the old vein of thought, an observer was looking at an IS as an autonomous organ within a living organism's context, that is, its business environment, expecting from the IS to follow the needs and the commands of its context. From a different perspective, the user-IS system co-evolutes as a closed system (Stamoulis et al., 1998) towards a destination, a purpose, a telos that cannot be known in advance. Having used the self-organizing theory coming from the physical sciences to explain the theoretical foundations of TIS (ibid), the embrace of the ateleology ideas comes as a natural subsequence. At about the same time, research work (Baskerville, Travis, & Truex, 1992, 1996; Truex, Baskerville, & Klein, 1999) in the same area – but not starting from the need for true flexibility – started converging to the need for ateleological IS development, by means of a methodical systems development, where a methodical means is not obeying to well-structured techniques through which compliance with a specific goal can be achieved. In theory, the problem was well-defined. In practice, there was a missing link that would connect the tailorability concept with possible implementation routes. The definitive characteristic of a TIS – the key to achieve tailorability – is the deferred design decisions (Patel, 1999, 2003) mechanism.

Design decisions translate the user requirements into rigid specifications, which render an IS short-living by definition. Since an IS must constantly adapt to changing user requirements, design decisions must be deferred until run-time, when the ultimate judge, the user, determines the preferable IS behaviour and, consequently, fixes the appropriate design decisions for the IS to run. If ever these need to be changed, this IS behaviour change must occur at the user's control, at run-time, while the IS remains operating. To prove realistic, this new way of thinking had also to define other issues such a method of work for implementing TIS, a security and stability framework for TIS that constantly changes and so forth. As far as the former is concerned, the concept of an IS blueprint has been devised (Stamoulis, Theotokis, Martakos, & Gyftodimos, 2003). In line with the ateleology and a methodical IS development, TIS are initially constructed as blueprints with minimal features that correspond to a vague problem definition within a problem domain, as the user starts using the TIS, co-evolutes towards an end, a goal, a telos, and takes design decisions that implement a specific, well-defined IS behaviour, pertinent to each and every problem situation. To arrive at an IS blueprint, the Soft Systems Methodology (Checkland & Scholes, 1990) can be perfectly used, as explained in Theotokis (2003). As far as the latter is concerned, a model with separations of concerns in terms of security and change authorization domains has been proposed to guarantee for the stability of TIS (Farmakis, Kanellis, Martakos, & Stamoulis, 2000).

In the following, an innovative implementation of the deferred design decisions mechanism is been presented called the ATOMA architecture and new developments towards tailorability are being discussed. This approach is based on the concept of separation of concerns as proposed by Morch (2003) and Randall and Lycett (2003). Concern separation is key in realising deferred design decisions as it facilitates the notion of "injectable" behavioural adjustments in existing operational IS. As such, ATOMA was selected as representative of systems supporting concern separation and consequently behavioural variations.

The ATOMA model was developed in order to better model cross-cutting behavioural changes related to the evolution of the behavioural landscape of an information system, by alleviating the problem associated with traceability and monitoring, scattering and tangling and, at the same time, enable the incorporation of modelled behavioural changes into an existing system without affecting in any form the system's stability. An implementation of the model exists and is based on the Java programming language. The key idea behind ATOMA is to improve the effectiveness in realizing modifications, of the behavioural landscape of object, in a dynamic, transparent and natural

Related Content

Implementation of the ASP Organizing Vision: The Role of Participation and Trust
Rajiv Kishore (2003). *Business Strategies for Information Technology Management (pp. 160-167).*
www.irma-international.org/chapter/implementation-asp-organizing-vision/6110

A Hybrid Approach Based on Genetic Algorithm and Particle Swarm Optimization to Improve Neural Network Classification
Nabil M. Hewahiand Enas Abu Hamra (2017). *Journal of Information Technology Research (pp. 48-68).*
www.irma-international.org/article/a-hybrid-approach-based-on-genetic-algorithm-and-particle-swarm-optimization-to-improve-neural-network-classification/182712

Project Management Models in IT
Marly Monteiro de Carvalho, Fernando José Barbin Laurindoand Marcelo Schneck de Paula Pessoa (2005). *Encyclopedia of Information Science and Technology, First Edition (pp. 2353-2358).*
www.irma-international.org/chapter/project-management-models/14612

Differential Impacts of Social Presence on the Behavior Modeling Approach
Charlie C. Chen, Lorne Olfmanand Albert Harris (2008). *Information Communication Technologies: Concepts, Methodologies, Tools, and Applications (pp. 2398-2416).*
www.irma-international.org/chapter/differential-impacts-social-presence-behavior/22825

Business Technology Strategy for a: Specialty Chemicals Company
Anne Wilmsand Stephen J. Andriole (2010). *Journal of Information Technology Research (pp. 11-18).*
www.irma-international.org/article/business-technology-strategy/47214