Working papers in
Information Systems

GENERIFICATION AS ECOLOGY

Petter Nielsen

WP 1/2016

Information Systems Group
Department of Informatics
University of Oslo
Gaustadalléen 23b
P.O.Box 1080 Blindern
N-0316 Oslo
Norway
http://www.mn.uio.no/ifi/english/research/groups/is/

Generificaiton as Ecology

Petter Nielsen
Dept. of informatics
University of Oslo
P.O. Box 1080 Blindern
0316 Oslo
Norway
pnielsen@ifi.uio.no

Abstract:

Making software packages generic and implementing them in organizations is a key activity in systems development. This paper reports from a study of the implementation of a web shop software package in a multinational Telecom company. The aim of the paper is to improve our understanding and conceptualization of such processes. To reach this aim, we use an ecological lens to extend the common perspective on generification and implementation of large scale and corporate-wide software systems in the existing literature. Through the discussion, we show the strengths of this perspective on generification as ecology as bringing focus to the open and multi-levelled nature of generification, the blurred distinction between generification and implementation processes as well as the co-evolving nature of parallel software package implementation projects. This paper is an argument to break up the restricted project focus in information systems research and appreciate the non-lonely nature of organisational implementation projects.

Suggested bibliographic references: Nielsen, P. (2016). Generification as Ecology. Working Paper 1/2016. Retrieved from University of Oslo, Information Systems Working Papers website: http://www.mn.uio.no/ifi/english/research/groups/is/publications/working-papers-in-information-systems

1. Introduction

The challenge of inflexible legacy systems, the cost and scarcity of internal IT resource and the promise of significant cost-savings motivate organisations to use software packages. Developed by software vendors, software packages are not particularly tailored for one organisation, but sensibly designed to serve a range of different organisations and contexts. By reaching a generic nature, through what Pollock et al. (2007) have termed generification work, a significant portion of the investment and maintenance costs of the software becomes shared among many. To achieve their generic nature, software packages must be based on a future-oriented architecture and capture requirements from a wide audience of organisations over time.

Although vendors strive to support all organisational requirements, organisations investing in software packages will usually discover that needed functionality is lacking in package. Thus, software packages are usually customised and extended to meet local and often idiosyncratic requirements as a part of its organizational implementation (in this paper we are using implementation as a short form for implementing software in an organization). The choice to invest in a software package by organizations therefore requires careful considerations not only with regard to the choice of vendor and package, but also whether the benefits and the costs of implementing software packages in organizations outweigh in-house developed or commissioned software.

Multinational companies are inclined to see a software package as having the potential not only to reduce costs but also as a tool to standardise technology and practices across the organisation. Implementing software across different national contexts is at the same time challenging as it usually involves idiosyncratic and local specificities, related to practices, capabilities, software systems, strategies and market particularities. Still, software vendors are able to make generic software packages, and multinational companies are able to implement them as global standards.

Based on a case study of web shops in a multinational telecom company, this paper analyses and discusses the processes involved and the strategy applied when implementing a software package. This discussion adds to and attempts to contribute to the body of research on the standardisation and implementation of large scale and complex information systems in global organisations (e.g. Ciborra et al., 2000). In this body of research, software is seen and understood not as independent code, but in an intricate interplay with multiple levels of organisations, architectures, technology, and people. The case discussed is therefore not simply about defining a software package with optimal functionality and implementing standardised software cost-efficiently. It is so much more than just that; it is about an intricate web of projects and processes of digitalisation, integration, transformation, architecting, diverse local practices, governance and organisational politics. In concurring with the work of Pollock et al. (2007) on generification and generification work, this paper emphasises the need to focus on how general solutions are made generic, and not only on how difficult it is to make the general specific.

The existing body of research on the implementation of software packages in global organizations (standardisation) and the development of standardized software packages (generification) has focused on the conflict between local and global needs and strategies for incorporating local requirements into global standards. In this paper, it argued that both the standardization and the generification literature at the same time have had a too narrow perspective on standardization and generification. It appears as processes of generification is closed and confined within vendor driven projects and that organisations implement software packages one at the time. The empirical focus on generification has been on software vendors and their strategies. The organisations implementing the software packages are treated only as participants involved in requirement gathering activities and the specification of functionality done by the software vendors. Research on the implementation of software packages in global organizations are not taking seriously into account that individual projects "… only constitutes one of many different projects, activities, ventures, undertakings, problems, issues, decisions, and solutions that gradually pass through the history of its organizational context." (Engwall, 2003, p. 805). This "lonely" project perspective is common in the studies of information systems projects in general, and packaged software projects in particular (Elbanna 2010).

Despite a number of studies of the implementation of software in global organisations and generification strategies and processes, we still have limited knowledge about how software packages are implemented in multinational companies due to a limited project focus. In this paper, we present as study from the point of view of a team implementing a web shop software package in a multinational company. The aim of the research is to understand and better conceptualize this process. At the core of the findings from this research, we emphasize the open, multi-level nature of generification as performed by vendors but also by the implementing organisation as well as the parallel nature of software package implementation projects. We argue that research to a larger degree should open its prevailing limited project focus and discuss the generification of software packages and the implementation of software packages in multinational companies as generification as ecology. Through the analysis, we illustrate the relevance of such a perspective. The main contribution of the paper lies in empirically describing a different generification and implementation story, and filling a gap in the literature on implementing large scale and complex information systems in global organisations by conceptualizing the implementation of software packages as processes unfolding in parallel with other implementation processes and generification as a process involving the implementing organization beyond being a source of requirements.

The rest of the paper is designed as follows. In the following section, related research on software package implementation and ecology is discussed. In section three, the qualitative research approach used in this paper is described and discussed. In section four, a case study of the implementation of web shops in a multinational telecom company is presented, followed by a discussion of this process and its dimensions of generification work in section five.

2. Related Research: From Independent Software Projects to Ecosystems

2.1 Implementation of Software Packages

While information systems are commonly seen as having the potential to solve a wide range of organizational problems, the literature on how information system implementation projects may be challenged is extensive (for example Hirschheim and Newman, 1991; Swanson, 1988; Walsham, 1993). The more particular challenges of and organizational strategies for implementing software packages have also received substantial attention in the literature. For example, arguing for the potential in packages over custom developed systems, Lucas et al. (1988) early introduced an implementation model based on challenges related to package-organisation misalignment and vendor dependency. Soh and Sia (2004) further explored the potential misalignment between the context of implementation and the context where the software package is embedded. Information systems must be aligned with institutional structures, whether imposed by the nation-state or professions, or voluntarily acquired based on experiences, management decisions or preferences. Software packages are also commonly promoted as reflecting best practices. This has however been questioned – how best practice can be context-free and readily acquired, stored and transferred (Yeow and Sia, 2008). Complicating this further is the complexity of organisations, which typically consists of sub-units with their own particular needs. Other researchers have focused more on the process nature of implementing software packages. Larsen and Myers (1999) questioned package-driven business process re-engineering and how success only can be evaluated over time because software will have implications on the organisation more at large. Pozzebon and Pinsonneault (2005) draw the attention towards the role of internal members (those being involved in configuration) and how external consultants may play a crucial role in the implementation of configurable packages. Further, Hong and Kim (2002) illustrate how packaged software has challenges related to uncertainty in acquisition and hidden cost in implementation.

2.2 Organizational Complexity

Software solutions are increasingly interdependent, large scale and assembled from a variety of components not under the control of a single governing body, i.e., they are complex (e.g., Hanseth & Lyytinen, 2010). They are no longer developed and managed as stand-alone and monolithic, but are built and implemented as parts of a larger whole. While systems developers may use conventional systems development methods to develop software components, aligning with and becoming a part of the larger “network” is a different task. A task not only about decomposing complicated requirements and deriving optimal technical designs, but grappling with the complexity that emerges from the heterogeneity of other systems to relate to, other parallel processes and projects, players involved, and the lack of overall control. This complexity poses a challenge to existing theoretical and methodological perspectives in software engineering in general (Sommerville et al., 2012). There is a need to capture and work with the development and implementation of software as based on complex coalitions, emerging and developing over time. The plethora of actors of different kinds involved, the dimension of time and the introduction of software packages as a process of negotiations, must be taken into account (Monteiro et al., 2013). This complexity is not (yet) well reflected in the literature. For example, there is a vast literature on implementation of Enterprise Resource Planning (ERP) systems, reviewed by Aloini et al. (2007). Studying the risks involved in large ERP projects, they summarize a range of important aspects to be considered in implementation projects: Technology, market, financial, operational, organizational and business. They also describe different phases of the ERP life-cycle, from concept (selection), implementation (deployment and integration) to post-implementation (evolution). But at the same time, this literature is not taking into account scale and complexity – software packages are described as packages in terms of being software (only), provided by a single vendor and implemented for one, dedicated purpose.

2.3 Standardisation and Generification

Due to the components' heterogeneity and dispersed nature and networked nature, standards are essential building blocks in large-scale and complex information systems (Lyytinen & King, 2006; Hanseth, 2000 and Ciborra et al., 2000). These systems, such as for example enterprise resource planning (ERP) and customer relations management (CRM) systems, which can span multiple organisations and multiple countries, rely on standards to be the glue that holds them together. At the same time, as standards diffuse, more radical changes become difficult centrally to implement and disseminate because the users are distributed and are not under central control, and standards in turn become cumulatively more change resistant (Egyedi, 2002; Hanseth et al., 1996). For that reason, standards are more than neutral technical specifications. A key challenge highlighted in the literature on global standards is that the spread of standards across various settings also requires sensitivity to local contexts by allowing local and situated practices to continue. Universal standards introduced top-down will challenge existing practices with the risk of breaking them down or that the new standard will be rejected. Standardised software comes with risks and potentially substantial costs, making it challenging, if possible at all, to "transport" standards between contexts (e.g., Hanseth & Braa, 2001; Berg, 1999; Walsham, 2001). Making software relevant in one local context may also make the same software irrelevant in others (standardization), software will get entangled with other software (embeddedness) (Monteiro et al., 2013) and there is an inherent mismatch between visions of global and standardized practices and the reality of a wide variety of micro-practices throughout a global organization (Hepsø et. al 2009).

Pollock et al. (2007) has pointed to this standardisation literature as being overly focused on the misalliance between standards and local practices and the work necessary to make global standards work locally. They emphasise the need to also take into account the less investigated and understood processes of making software work across contexts. Despite all difficulties, it is hard but still possible to use the same systems and software packages in different settings. They have thus argued for shifting "… the debate from understanding how technologies are made to work within particular settings to how they are built to work across a diverse range of organisational contexts" (Pollock et al., 2007, p. 254). They label this process of making the generic out of the specific generification work, work that enables vendors of software packages to treat different local sites as the same, and develop software packages travelling across context efficiently. At the centre of generification work is the trade-off between the particular and the generic in terms of e.g. prioritising requirements according to their relevance across the customer base and striking a balance between functional diversity and the requirements for local customisation.

The standardisation and the generification literature relate to the complexity perspective. However, both strands have a focus on standardisation and generification as processes strategically driven centrally by one focal actor and in a rather unidirectional top-down fashion. Either, the standard is portrayed as propagated from a headquarters in a multinational company to local operations (commonly failing or with limited success), or a vendor is releasing a software package to certain customers. Requirements will be gathered from the local operations or the customers, but the standard and the software package is made centrally. In that sense, the discussion only captures standardisation and generification as processes under the control of a single governing body, as linear and as decoupled from the implementation of the software. Further, the implementation process is portrayed as lonely, as only one software package is implemented in an organization at a time.

2.4 A Software Ecosystem Perspective

In this paper, we argue for taking an ecological perspective on generification. The concept of ecology is borrowed from biology and used by information systems researchers to describe networks of diverse actors influencing each other's and being mutually dependent within a specific (eco)system. Both the standardization and generification literature is based on a rather broad and encompassing socio-technical perspective, taking into account not only the software but also the social in terms of the role of different actors, local practices and institution. In this paper, we argue that these discourses still miss out on key dimensions of generification and the implementation of software in multinational companies, namely the multi-level nature of generification and the simultaneous unfolding of software package implementation projects. We will show that this is a crucial part of the software ecology and key to understand the unfolding of the project described in the case study in this paper.

In their review the software ecosystem research, Manikas and Hansen (2013) point at the early definitions of software ecosystems as collections of interconnected software products. Relating to the idea of relations between software systems, Lungu et al. (2010) apply a project oriented perspective and argues that software systems are not developed by projects totally independent of other software projects. Rather, projects unfold at the same time and are influenced by and influencing other projects. Lungu et al. argue for an ecosystem perspective to illuminate these interdependencies and therefore define software ecosystems as: "… a collection of software projects which are developed and evolve together in the same environment." (p. 265). Manikas and Hansen (ibid) suggest combining these and other definitions into the following definition: " … a set of actors on top of a common technological platform that results in a number of software solutions or services." (p. 1297). This definition moves beyond the pure technical definition of Lungu et al. (2010), by introducing the social dimension of actors. At the same time, it considers particularly the rather recent phenomena of software platforms, upon which different actors can undertake their software development.

Motivated by the prevalence of customizable and configurable products, Dittrich (2014) argues for the importance of better understanding how the development and evolution of software products differs from project-based development of software systems. Based on several case studies, she identifies key differences between project-based and what she terms software eco-systems, related to the interaction of different actors, the technical design and the organization of software development. With a particular focus on software platforms, Dittrich questions the project, and its closure in scope, effort and time, as an adequate point departure. Software platforms involve software development beyond making the platform. Not only will platforms be maintained, but they also open up for continuous innovation and customization to meet local needs. Dittrich further identifies three key features of product software development within ecosystems: 1) Parts of the design and innovation is deferred to other actors closer to the use context; 2) multiple layers of actors customizing and configuring software is involved; 3) and development is driven by different factors such as bug fixes, the need for new features and technical re-engineering.

In this paper, we are not considering software platforms in particular, but organizational contexts where generification is unfolding on multiple levels and several different actors are implementing different software systems at the same time. While independent in the sense of being based on different approaches, strategies, timelines, and managed and run by different people, the projects also influence each other. Their interdependency spans different technical as well as the social levels in ways which cannot be fully anticipated, what we can term co-evolution (Jansen and Nielsen 2006). Together, these co-evolving processes make up an ecosystem involving multiple and different actors (developers, managers, users etc.) and software packages and generification on multiple levels (by vendors and organisations implementing the package).

To summarize, in this paper we will discuss the implementation of a software package in a multinational company. We are extending the existing research on generification and standardisation by applying a broader ecology perspective on implementation and generification by taking into account the multi-level nature of generification and the co-evolution of multiple implementation projects.

3. Research Method

The empirical setting of this research is the multinational telecommunication company Telco. With its corporate headquarters located in Europe, Telco is one of the larger mobile operators, with 150 million customers, 30.000 employees and, at the time of the study, operations in 11 markets across Europe and Asia. It has a leading position in mobile, Internet and TV services in its home market, while its international presence is primarily based on mobile operations. The particular focus in this paper is an initiative by the Industrialisation Unit (IU) in the corporate headquarters in Telco to renew the local, independent and not standardised web shops in the various Telco operations. This initiative was grounded on activities dating back to 2009, leading up to an attempt to implement a standardised web shop software package during 2012. The initiative turned out to be overly challenging and ultimately failed to reach its primary aims, at least in terms of establishing a globally standardised web shop. These challenges and the complexity of the process triggered the interest of the author to conduct an analysis of the initiative, and what the author saw as a process of increasing openness and interconnection between processes and projects.

The author was employed in the R&D unit of Telco, organised as a part of the IU from 2006 to 2012. Engaged as a resource from the R&D unit in the business development programme promoting the web shop initiative, the author was instrumental in driving parts of the standardisation process from the IU side and participated throughout the process. The author was further involved in concrete local web shop projects in markets in Europe as well in Asia. The author also participated in an assessment of the status of web shops and eCommerce in general in all of the operations. Throughout these activities, in addition to other forums and meetings, the author was a part of the ongoing discussions related to web shops on a variety of levels: in the IU, in the operations and in Telco in general. Finally, the author was also engaged in active discussions with the potential vendors of web shops’ software packages.

The primary aim of the researcher was not to tease out some “objective” truth about the initiative, but rather, through interpreting the different perspectives of the various people involved, to provide a deeper understanding of the process and to conceptualise this related to standardisation and generification. The researcher thus followed an interpretative kind (Walsham, 1993; 1995) of exploratory case study approach (Yin 2003). A key source for this case study was the experiences of the author. Through his efforts and participation in the implementation process and other related processes, the author developed a deep understanding of what was going on. Because the case study was planned after the initiative was “terminated”, the author was not a participating observer, but rather started out as a pure participant and ended up doing interviews with the other participants post hoc and as a reflecting practitioner. As a participant, the author was deeply involved and interacted in meetings globally and locally, discussion at the IU, discussions with operations, discussions with vendors, and discussions with other units within the IU (sourcing and technology in particular). This research approach will necessarily come with a certain bias. In particular, the researcher was an advocate for standardising the web shops and had the vantage point of the IU throughout the process. This will necessarily be reflected in the choice of data collected and the way they are interpreted. For example, the perspectives and concerns of the software vendors and the Telco operations are less prominent and only based on the researchers interpretations. Their actions are also interpreted in the light of an ultimate goal of standardisation. At the same time, it is the IU vantage point and perspective which triggered and is driving this paper’s argument for conceptualising generification as ecology. This paper is attempts to establish and show the relevance of this perspective by discussing a case of a process driven from and by the corporate headquarters. Establishing whether this conceptualisation also make sense in other cases and from the vantage point of a software package vendor or the local operations of a multinational company will require more research.

The fundament of the case study is the author’s recollection of the project and central events. Based on this, a first skeleton of the case was written out. Already at this point in time, the two central themes of this paper was identified and pursued further. One key issue was the way in which the web shop project interacted with and how it was influenced by other parallel software implementation projects. The other was the strategy behind and the attempts towards developing a “local generification” process on top of the software package. Guided by these two fundamental issues, several interviews were conducted to understand what was going on and documents were analysed to see how strategies were reflected.

All the web shop project participants from the IU were interviewed. The motivation for only using these “internal” sources was partly because of ease of access. Also, the focus was on the approach of IU in this particular initiative. An alternative would have been to broaden the scope and interview the “whole” network of actors involved or certain slices of it more in depth. This was not possible within the constraints of time and resources. The influence and strategy of the “external” actors are still represented based on the interaction experienced by the author and the interviewees. In addition to the interviews, a range of presentations and other document were collected. A particular focus when analysing the documents was how the web shop project was related to other software implementation projects. The data sources are summarised in table 1 below.

Interviews	#6 February 2013 Project Manager, Programme Manager, Project Participants from distribution team (2), architecture group in the IU and sourcing group in the IU Total 7 hours, recorded and transcribed (total 54.000 words)
Documents	#32 May 2009 – January 2013 24 presentations, 5 Minutes of Meetings, 2 Preparation documents and 1 discussion document Presented to project steering committee, the IU management meetings, global management meetings and global management boards

Table 1 Data sources

The interviews did not follow any written interview guide. They were all fully recorded and transcribed. The interviewees were asked to broadly describe the unfolding of the project, the strategy applied, how this project was different compared to other projects they had participated in, and key challenges and issues. The interviewees were also asked to reflect on what they and Telco could learn from the project. Notes were taken during the interviews. After the interviews, based on the previously identified key issues of parallel implementation projects and the multiple levels of generification, interesting insights were identified from the notes and structured in a separate document. Further, during the transcription of the interviews, key events, challenges, and approaches were highlighted in the text. Subsequently, this text and its highlights were revisited and reflected upon by the author. Also, the transcriptions were searched to build a clear timeline of the project. Finally, the notes and the highlighting in the interviews were seen together, offering key themes to focus on in the description of the case as well as the key issues addressed in the discussion. The contents of the documents were also used to confirm and better understand the strategy and approach of the IU during the project, as well as to document the unfolding of the project.

4. Case: Standardising the Web Shops in Telco

In this section, a case study of the implementation of a standardised web shop in Telco is described. We start out by describing the multinational telecom operator Telco, before we go into the details of the implementation project.

Telco is a multinational telecom operator, with a decentralised organisation composed of largely independent operations in 11 markets in Europe and Asia. While certain organizational functions are placed at the corporate headquarters, the operations are by and large governed locally. This offers the operations the flexibility to develop their organisations according to their local markets. This organisational setup is, at least from a historical point of view, closely related to the very different nature of the countries Telco operates in. For example, the markets are different in maturity when it comes to the mobile networks in operation (second, third and fourth generation of GSM) and the market sizes range from 600,000 to 1.2 billion inhabitants (and potential customers). As a side-effect of this setup, Telco has as many IT stacks as operations, and there is only a minimum of coordination and standardisation across. Therefore, one important area for improving cost-efficiency identified by the Telco corporate headquarters is to strengthen the coordination of the IT function. Their ambition is to leverage on scale and volume and the unit responsible for this process is the industrialisation unit (IU).

4.1 The eCommerce Team and the Web Shop Initiative

IU has a dedicated eCommerce team working with web shops. Web shops in Telco have historically been treated as a fringe channel for sales and service, with the well-established and traditional retail shops being in focus. As a result, the web shops have been disconnected from other channels. One consequence of this is that customers buying a phone in a Telco web shop cannot pick it up in a retail shop, and the retail shop will not be aware if the customer has been active in the web shop.

Telco has not previously coordinated and standardised their web shops globally. This has led to there not only being 11 different web shops in terms of software, but also 11 different web shop strategies and organisations along several dimensions. First, some web shops are based partly on software packages and partly on internal software development efforts, while others are bespoke software. Second, some have fully automated solutions managing complete sale processes, while others are more or less manual. Third, some web shops are operated internally while others have components run by external partners. Fourth, some web shops are partially integrated with back-end support systems while others are run as independent channels. Fifth, some web shops are focused only on selling mobile phones while others focus on top-ups of prepaid subscriptions. This illustrates how the different operations have aligned their web shop with other internal systems, partners in their local markets, as well as according to the market they are operating in.

From the very beginning, the eCommerce team appreciated web shops as something more than just selling products and services online. A web shop must deliver powerful and flexible capabilities. These capabilities include linking the web shop and back-office systems to reuse customer and product information and streamline sales and service processes, and the flexibility to launch new products and campaigns in the web shop swiftly. Relevant back office systems include customer relation management, Enterprise Resource Planning and supply chain management.

The dialogue between those responsible for the local web shops in Telco and the eCommerce team, and outputs from assessments made early in 2011, indicated that the web shops in Telco performed far below expectations. This was in terms of the sales and service functionality they offer to customers and the functionality and flexibility they offer those administrating the sales and service activities in the web shops. There was a common understanding that customers are not likely to have a particularly positive experience using the different web shops of Telco. At the same time, those administering the channel struggle to be agile enough to meet rapidly changing market demands and have to rely heavily on scarce IT resources and cumbersome manual processes to make things work.

During summer and fall of 2011, the eCommerce team launched a Request for Information (RFI) project on web shops. The objective of the RFI was to get more information about different web shop software packages and their vendors, as well as address the feasibility and the potential cost-efficiency of a common and standardised solution in all Telco operations. In the longer term, the aim of the RFI was also to lay the ground for a future Request for Quotations (RFQ) project culminating in the procurement and implementation of a standard web shop software package. A range of mature software packages from different vendors were identified through the RFI, offered by large software companies (like IBM and Oracle) and niche web shop software companies (like Hybris and ElasticPath). While being different in focus, size etc., they all offered more or less the same web shops from a functional point of view. Some of them also offered a telecom accelerator—a template/blueprint for what mobile operators typically require as extra functionality to be able to use the package and thus reduce the time to implement it. The RFI project concluded by identifying 5 potential web shop vendors who seemed capable of delivering the functionality, the user experience and required cost-efficiency for Telco. However, none of the vendors had substantial experience delivering a standard web shop throughout a multinational telecom company. Related to this, the RFI project concluded with a set of key issues to be examined further. The heterogeneity of the back-office systems in the different operations was identified as the most prominent challenge, raising issues of integration costs as well as the possibility and real potential of standardisation. Another conclusion was that there was a need to work with the operations to persuade them to use standard functionality. This strategy had to take into account that some mature operations would possibly feel a standardised software package was inferior compared to what they already had in certain areas where more immature operations would consider it far more advanced than necessary. In terms of business case, the RFI project also identified that market size and online transaction volumes varied significantly from operation to operation. A possible need for a differentiated business model and/or more than one standard to cover basic and advanced web shops according to market potential was identified. Finally, the functional role of the web shop was also questioned: It can, for example, master customer and product data, or just link up with other systems offering this functionality.

4.2 Local and Global Requirements

Through meetings and discussions with the operations, a broad range of requirements was identified. In addition to the standard web shops functionality, a range of key areas was pointed out with regard to reducing manual work (automation), independence from IT while making changes (flexibility), and supporting customers as they move between different channels (cross-channel). These requirements were results of the operations' experience with their own web shops, as well as their knowledge of other, better performing ones. In addition to these functional requirements and the importance of integration, the operations also pointed to other non-functional areas of importance. First, they were concerned about being able to use local system integrators for the implementation and maintenance of a standard web shop. They argued that using a global system integrator would require a lot of resources in terms of bringing them up to speed on the local IT stack. At the same time, a concern was raised that local presence is a requirement in order to secure flexibility and appropriate maintenance. Second, it was also a concern that with increasing transaction volumes in the web shop, the web shop must be scalable. The requirement of uptime will increase accordingly and failover mechanisms (both if the web shop and the back-end systems are failing) become critical. Third, the investment costs of the web shop had to be aligned with the transaction volume in the local market so as to secure a viable business model. Fourth, while some of the immature operations were focused on offering a simple web shop solution for basic phones, the more mature markets required the standard to offer a more advanced web shop (e.g. with Apps for smartphones).

The eCommerce team also identified several areas where local and global requirements were at least potentially in conflict. First, a central issue is the amount of integration work to be done due to the heterogeneity of the existing stack of back-office systems. If 80 per cent of the costs of a web shop are related to customisation and integration work, standardising the last 20 per cent will not result in substantial cost savings. The ability to standardise on integration and functionality across the different operations, with a “Telco layer”, was accordingly identified as critical. Second, autonomy was identified as a challenge related to operating a global software package over time. Should the web shops be operated with implementations in all operations, regionally or centrally, and how should change requests and maintenance be performed? Moving processes and decisions away from the local operations will reduce costs globally, but also reduce local flexibility and autonomy. Fourth, the way existing web shops were integrated also seemed challenging. Some of the operations have outsourced parts of the web shop value chain, and parts of the web shop were offered by a third party. While swapping the software to a standardised one did not seem so challenging, the third parties also provided functions in the value chain, such as stock management, distribution and servicing of mobile phones. This setup is much harder to change.

To deal with both local and global requirements at the same time, the eCommerce team chose a strategy to involve and work closely with the different operations as well as the other units in the IU. While acknowledging that not all requirements could be met, nor all conflicts between local and global interests solved entirely, staying close to the other involved entities in the process was seen to be important to developing and keeping a common understanding of the promises of and challenges related to a global web shop.

4.3 Relating to other Initiatives: Heterogeneity and Complexity

The web shop project did not unfold in a vacuum, but in parallel with and in interaction with a range of other local projects and initiatives by IU. Here, a few of them are briefly described to illustrate the context.

A large parallel initiative was run by the IU to identify ways to collocate server parks as well as maintenance and operations staff globally in Telco. Moving towards a shared service organisation, the strategy of Telco was to leverage group synergies, economy of scale and competence, substantial cost reduction through off-shoring, quality improvement through standardisation and reduction of local IT system costs. The eCommerce team saw the success of a standardised web shop for Telco in terms of cost-saving potential related to initial investments, but also related to maintenance costs and governance and thus dependent on a shared service centre. However, the shared service centre project was an ongoing project and it could at the time not offer the eCommerce project inputs on how the web shop could be supported, and specify the involved cost elements and potential savings using the future shared centre. The lack of an operative shared service centre not only introduced risk in the business model of the web shop project, but also mandated the initiative to request a software package that could be run locally in the short term, and equally important in regionally or globally service centres in the longer term.

The IT unit of the IU was also running a Telco technology architecture project. Their plan was to introduce a standardised architecture and back-office stack throughout Telco on the basis of several design principles ranging from service-oriented architecture (SOA), open interfaces, fine-grained application components to master data management. Through a significant transformation, the ambition was to introduce SOA in the different operations, but also a total back-office system renewal. Telco did not have a global IT architecture and the local architectures of its operations were not optimal, typically struggling with legacy systems and accumulating complexity. From the architecture project's point of view, the web shop project was an interesting case since they recognised this channel as a central component in the overall architecture, not at least in a digital future. The requirements an efficient web shop poses for integration with back-office systems were at the same a very strong argument for a renewed and common architecture across Telco. Therefore, during 2012, the architecture project got themselves deeply involved in the web shop project, using it somewhat as a pilot. While the architectural principles championed by the architecture project were reflected in the web shop project, the ongoing nature of the architecture discussions took considerable time and resources. More importantly, the architecture project did not have a fixed functional architecture to support the web shop initiative, but only one in the making. Even more importantly, it turned out, was the fact that the ambitions of the architecture project failed to get enough backing in Telco. Spending significant time with potential vendors during 2011 and 2012 with the aim of identifying a partner and running a proof-of-concept with one of the operations, the architecture project did not get adequate funding from the IU to continue during the autumn of 2012. The web shop initiative therefore had to continue without any clear idea of the future situation in terms of a common architecture.

One of the Asian operators also opted for a local web solution in 2011. The eCommerce team could not stop this process, but tried to be pragmatic by pushing the operation to choose a software package from one of their short listed vendors, and in particular persuade them not to choose to develop the web shop from scratch or together with a local partner. The management in the operation agreed that their initiative should be aligned with Telco requirements and they somewhat volunteered as being a pilot. At the same time, they chose a software package and a vendor in a setup that became problematic from a global point of view. First, the selected software package, when compared to other packages, required the most customisation. It was open and flexible, but very little came out-of-the box. One explanation for this choice was cost, since this solution was far cheaper than other options in terms of licences. Second, the contract was not with the software vendor but a small, local integration and implementation partner. This partner agreed to make any customisation for a very low fee, being truly eager to work with this operation. While these decisions can be justified from a local perspective, they were problematic from a global point of view. In particular, the choice of a local implementation partner that had no footprint outside the country made their contractual relationship non-reusable. At the same time, the operation claimed the solution was to be a pilot for the global IU initiative, even if it was not possible to use the same contract across Telco.

Another of the Asian operations also decided on a web shop autonomously during the fall of 2012. Their recently hired web shop manager got tough targets and to deliver on them he had to improve the existing web shop quickly. He found his key challenge as the lack of available technical resources for implementing, operating and maintaining a web shop in the local operation. From his point of view, taking part in the IU initiative would take too long and the local market did not generate the volume of transactions he believed would justify a larger investment. Based on previous experiences running web shops as a software-as-a-service model, the manager quickly adopted a web based web shop developed and hosted by a third party. The lack of a standardised architecture and service centre made this decision easy for him.

As a final example, during 2011 and into 2012, a digital web shop project was launched by a different Telco Group Unit dealing with digital services. This initiative primarily focused on digital content for smartphones, and not the traditional operator products and services (mobile phones and subscriptions). In early 2012, Telco signed a group-wide agreement with a vendor to deliver a standardised web shop for digital services. While not initially being coordinated, the eCommerce team and the digital services team tried to coordinate their efforts to create synergies and to avoid any overlaps in functionality. This coordination was in particular triggered by the fact that the operations were confused and did not understand the differences between the digital content web shop and the web shop. As a consequence of the coordination activities, the web shop project removed sales of digital content from its scope beyond requiring the web shop to be able to be integrated with a digital storefront. This did however backfire as the digital web shop project was put on hold due to lack of actual buy-in from the operations later in 2012. While initially challenging the web shop project as a “competitor”, the digital web shop project ended up challenging it with an unattended question about how to deal with digital content.

To summarise, numerous global and local initiatives influenced and partly overlapped the web shop implementation projects as well as other initiatives. For the local operations and the other units in IU, the choice of renewing the web shop is not only about going for a software package or not. The operations and IU are also considering different operational models locally, and their choice of approach is influenced by issues including transaction volumes, business cases, and internal capacity for implementation, operation and maintenance, as well as considerations related to integration, cross-channel, and flexibility and controlling the value chain.

4.4 Planning and Running the Web Shop RFQ

In September 2012, the Request for Quotations (RFQ) was launched and five web shop vendors were invited to participate. The RFQ documentation requested the potential vendors to describe their web shop software and indicate compliance to 446 requirements. In addition to indicating compliance, vendors were asked to indicate whether compliance was generic or would require customisation. Further, the vendors were requested to give price models for a licensed software package and a software-as-a-service solution. The eCommerce team analysed each of the responses and returned to each of the vendors to clarify certain issues. After this, the vendors presented their solution as well as how their solution would handle a set of use cases for the eCommerce team as well as representatives from the operations.

The evaluation of the different solutions was based on three core areas: degree of customisation necessary, requirement scores, and showstoppers. To be used as a standard, the solution was to be described as composed of the generic software package (out-of-the-box), a suggested generic “Telco layer” on top of this for all the operations in Telco, as well as what needs to be customised particularly for each operation. With the development of a generic “Telco layer” on the top of a software package, Telco particular functionality and integration points could be developed only once and reused in the 11 operations. The ratio of functional requirements across these different layers was used as evaluation criteria, where the target was to have as much functionality in the core product, alternatively in the "Telco layer", as possible. The second area was the scoring according to the 446 requirements. The third area of showstoppers consisted of key critical areas considered important for the success of the web shop. As more soft and qualitative criteria, this concerned compliancy to certain requirements related to multi-channel support, integration approach, and the possibility to be run from a shared service centre.

4.5 Terminating the RFQ Process

In December 2012, the project concluded by shortlisting 3 vendors and was ready to start preparing these vendors for commercial negotiations. However, just before starting this process, it became clear that the project would not get a governance model and a clear coordinating role related to a common "Telco layer". The support for this centrally and from the operations was too weak. Consequently, in practice, there was no decision that web shops and any standardisation activities were to be the responsibility of the IU. In this situation, the eCommerce team did not see any value in pursuing the initiation of a negotiating process with the vendors. Without a standard and some sort of central governance, they did not see any merits of pursuing the matter any further and so the project was terminated.

This journey somewhat ended up as a closed circle for the eCommerce team. From identifying the challenges of the independent web shops, establishing requirements in collaboration with the operations, culminating in the formalisation of and preparation for a standard. But, finally then, coming back to square one with independent web shops only and just the continued role of supporting individual operations. The consequences of this way of termination, and what can be called the failure of this project, can be debated. In hindsight, it is pertinent to also reflect upon what could have been done different. The project participants from the IU and the operations had a steep learning curve, establishing a common understanding on what web shop software package can deliver and what it takes to make and accept one common standard. At the same time, the project did not have the required backing from the global level. Another important factor was the failure of parallel projects in delivering e.g. a common architecture, an operative service centre and a web shop for digital content. With these other components in place or in a more mature stage, the web shop project may have ended differently.

5. Discussion

The success and failure of standardising software packages in global organizations have been attributed to different factors, ranging from the features of generification processes, the flexibility of software to the nature of institutionalized local practices. The key argument and contribution of this paper lies in its focus on the implementation of a software package in global organisations as unfolding in a dense and complex environment of multiple projects. What we have seen is that the generification space is an ecosystem composed of multiple parallel implementation projects and multiple levels of generification activities. These loosely interconnected initiatives and projects are not necessarily well coordinated while at the same time being dependent on another for their success.

5.1 Parallel Implementation Projects

Generification work has been described as focusing on two types of requirements. First, it has captured the fine balance of choosing what functionality to include in the software package and what needs to be custom made at the user site (Pollock et al., 2007). By aggregating functional requirements from a range of customers, software vendors adjust and refine their software packages to accommodate what is common and most important for the most important customers. Second, as long as the software package will interact with other information systems, the generic nature of the software also relates to how easily it can be integrated in a larger IT portfolio (Johannessen & Ellingsen, 2009).

In the case of Telco, the web shop project was tightly related to other, parallel projects. For example, the total cost of software packages includes licence and customisation, but also its operation. Operating and hosting the web shops in only one or a few (e.g., regional) sites will save costs. At the same time, the availability of resources to implement and handle a local instance of the software package varies across Telco operations. For some of the operations, there was a lack of resources dedicated for the web shop. Under these circumstances, the software package is irrelevant if it is not backed with additional resources. One approach to this already available in the market is web shops offered as software-as-a-service. While limited on flexibility, with this model the service provider is fully responsible for the operation. Furthermore, the resource situation may change and the willingness and ability to operate locally, regionally and globally may change over time. Thus, the generic nature of the web shop is also related to the operational models supported. In the case of Telco, the ongoing establishment of a shared service centre confused the web shop projects by creating uncertainty about how it would influence the business model as well as whether it would be established in time. Another example is the need for a web shop to be integrated with back-end systems to obtain the required user experience and support for those administering the web shop. For the IU, the web shop had to work with a variety of local architectures. There was no global architecture, and the existing local ones were far from optimal and some of the operations lacked what others had as core components in the architecture, such as a product database. While a global architecture was in the making, this was also an ongoing project. This architecture initiative showed interest for the web shop project and offered input to the architecture of the web shops and its relation to other software components in Telco. In doing so, it delayed the web shop project slightly. More important, when the architecture project was terminated, the web shop project was left with uncoordinated architectures in the 11 operations. The web shop project did not come up with any straight forward solution to this challenge, but approached it by suggesting two different solutions to serve mature and immature operations and their architectures. A requirement related to the software-as-a-service solution serving immature operations lacking important components in the architecture (for example product catalogue) was also that it should be possible to swap to a software package more tightly integrated with the back end system when the operations become ready in the future.

To summarise, software attributes in terms for functionality and integration support were prominent parts of the standardisation and implementation of the web shop software package. However, defining and implementing the web shop software package did not unfold alone, but significantly influenced by parallel projects such as the implementation of a shared service centre, renewal of the architecture and the web shop for digital content. The argument here is not to attribute the successfulness of the web shop project to these parallel, but co-evolving projects. Rather, we have shown that ignoring them would leave us with less explanatory power and a weaker understanding of the implementation project.

This is reflected in our case as the influence of the parallel projects such as those related to digital content, architecture and shared service centre.

5.2 Multiple levels of generification

In the case of Telco, generification work was not an activity performed only by a software vendor in response to organisational requirements, but proceeding in layers also involving Telco in their implementation the software package. When Telco evaluated the different software packages, it became evident that they all required significant work to deliver the needed front-end functionality and back-end integration. While the software packages appeared generic, they were also deemed not generic enough as a standard. They required too much additional implementation work to achieve the necessary economy of scale of a standard. To deal with this, Telco suggested a strategy where they would develop their own generic "Telco layer" on top of the software package. With a "Telco layer", time-to-market was expected to be shorter, the need for customisation for each operation less and the business case for a standard more viable. This approach is interesting in several ways. The introduction of a second layer of generification comes with another level of trade-offs between local flexibility and global standards. While Telco accepts the limits in the generic nature of the software packages, the question about how much the "Telco layer" should cover functionally and integration-wise remains. From the global Telco perspective, the "Telco layer" should cover as much as possible. For the less mature operations, a turnkey web shop would be preferred. For the mature operations, this is more complicated as the functionality already may exist in other software components. Thus, "internal generification" for Telco was about dealing with and prioritising diverging needs of the operations much in the same way as the software package vendors are dealing with their customers. The economic rationale behind standardising as much as possible may have deep implications on flexibility and the functional architecture of the local operations. At the same time, a "minimum" standard that fits all will lose its generic nature if it necessitates too much particularisation work. This two-layered model and the involved trade-offs between the generic and the particular illustrate a situation with multiple layers of generification work going on.

If the operation of a standardised web shop is not made right, it is likely to fragment as a standard over time because the operations will develop their own functionality at the cost of the "Telco layer" and the software package. The "Telco layer" adds complexity to this, since it is not always trivial to decide where new functionality should be implemented—locally, in the "Telco layer" or in the software package. Thus, for Telco and its operations, it is not only the software but also the internal process where new requirements are handled which has to be adequate across different contexts, and thus generic. Generification work in this context is about governing the balance between the particular, the Telco specific and the software package in a way that makes the web shop relevant operationally, and for Telco as a standard over time. Another level of complexity in this setting was the approach to centralisation in the implementation (software) and the operation of the web shops (operational model). For example, the operation of the web shop can be central, while the common software code base can be minimal. Or the code base can be completely standardised, while the web shops can be operated locally and independent. One particular approach to this was the use a software-as-a-service solution. The choice to take this approach was based on accepting the very limited possibilities for customisation. The merit of this approach was that the software code does not travel at all.

It comes as a package where the functionality is set and the maintenance and operation is taken care of by the external service provider.

The generification argument of Pollock et al. (2007) is based on changing the focus from how difficult it is to implement standardized software in global organizations with different local needs, to how it is actually possible to make global standards. They describe how generification can be based on strategies of discrimination and focus on development of the functionality which makes most value for the most valuable customers. While at the same time offering a good enough product for those not so valuable. This is a strategy which is working. In this paper we have described generification as a process unfolding on many levels, and not confined to organizations developing software packages, but also global organizations making their own, second level, generic package. In many ways, this global organization is facing the same challenges when it comes to give priority to different users and requirements as described by Pollock et al. (ibid). In addition, this second level must also relate to a more fixed customer base – they have to serve all organizational units. It is also less clear who the "most valuable customer" is. In the case of Telco, this was challenging because those business units being ready to implement was most immature and operating in the most immature markets. They therefore also had the weakest business case to invest in a new web shop.

To summarize, a perspective on generification as unfolding on multiple levels makes sense to understand the web shop project in Telco. While the software vendors already had made their generic packages, Telco pursued a similar process to make the package generic enough to become a standard. At the same time, their relation to the operations was of a different kind compared to a typical vendor – customer relationship. With this perspective we blur the distinction between the generification and implementation of software packages. And we have shown in the case of Telco that generification is not a project by the software vendors where they finalize the software package, but it unfolds on different levels and it will continue also after the implementation of the software package.

6. Concluding Remarks

Telco is a case of a large multinational telecom company, with small governance structures that leave each of its operations to operate mostly as they see fit (with resulting heterogeneity in strategy, organisation, architecture and IT portfolio). The pursuit of the standardisation of web shops, an area considered to be on the fringe by Telco and its operations, is then perhaps an extreme case. In a more "perfect" world, the different operations would have been more centrally controlled and standardised and web shops considered a more important channel. But still, Telco will continue to implement different software packages at the same time and in parallel. They will co-evolve and at certain point in time will intersect and influence each other for good or for bad. As a multinational company, Telco will also need to pursue ways to achieve the right standardised and generic nature of the software package implemented. The particular strategy described in this paper to pursue a local generification process on top of the already generified software package will also be relevant.

The empirical research and conceptualisations related to the implementation of large scale and corporate-wide software systems by Ciborra, Hanseth and associates distinguishes itself by dominating the information systems field and by taking, as well as extending, an ensemble view (Orlikowski and Iacono 2001) of information technology by taking into account what they term information infrastructures' distinctive properties (socio-technical, networked, distributed, etc.). This conceptualisation also explicitly draws upon Actor-Network Theory (ANT) (e.g. Monteiro 2000) and pictures the developments of information infrastructure as an evolutionary process which is intrinsically linked to the interplay between human and technical components. Interestingly, where this literature on the implementation of large scale and complex information systems is based on a socio-technical perspective and explicitly arguing for taking into account heterogeneity and complexity, they describe the context of implementation projects as more or less exempt of other implementation projects. This paper makes the important argument that the implementation process cannot fully be understood if detached from other parallel implementation projects. Implementation projects are not likely to lonely, and their faith is likely to be influenced, if not dictated by other co-evolving implementation projects. In addition, this paper argues for taking into account that generification work unfolds on multiple levels and in a much more blurred and distributed fashion compared to what has been discussed before by Pollock et al. 2007. In this paper we have shown the strength of a perspective on generification as ecology when it breaks up the common perspectives on generification as confined within a vendor project, the blurred borders between generification and implementation, and the co-evolving nature of implementation projects.

References

Aloini, D., Dulmin, R., and Mininno, V. 2007. "Risk management in ERP project introduction: Review of the literature," Information & Management (44), pp. 547-567.

Berg, M. 1999. "Patient care information systems and health care work: A sociotechnical approach," International Journal of Medical Informatics (55), pp. 87-101

Ciborra, C. et al. 2000. From Control to Drift: the Dynamics of Corporate Information Infrastructures, Oxford: Oxford University Press.

Dittrich, Y. 2014. "Software Engineering Beyond the Project – Sustaining Software Ecosystems", Information and Software Technology, (56), pp. 1436-1456

Egyedi, T. 2002. "Standards Enhance System Flexibility? Mapping Compatibility Strategies onto Flexibility Objectives," in Proceedings of EASST 2002 conference, Responsibility under Uncertainty: Science, Technology and Accountability. N. Brown, U. Felt and N. G. e. al. (eds). University of York. July 31 - August 3.

Elbanna, A. 2010. "Rethinking IS Project Boundaries in Practice: A Multiple-projects Perspective", Journal of Strategic Information Systems, (19), pp. 39-51.

Engwall, M. 2003. "No Project is an Island: Linking Projects to History and Context", Research Policy, (32), pp. 789–808

Hanseth, O. and Braa, K. 2001. "Hunting for the treasure at the end of the rainbow. Standardizing corporate IT infrastructure," Computer Supported Cooperative Work, (10:3-4), pp. 261-292.

Hanseth, O., Monteiro, E. and Hatling, M. 1996. "Developing Information Infrastructure: the Tension between Standardisation and Flexibility," Science, Technology and Human Values (11:4), pp. 407-426.

Hanseth, O., Jacucci, E., Grisot, M., and Aanestad, M. 2006. "Reflexive Standardization: Side Effects and Complexity in Standard Making," MIS Quarterly, (30:2), pp. 563–581.

Hanseth, O., Lyytinen, K. 2010. “Design theory for dynamic complexity in information infrastructures: the case of building internet,” Journal of Information Technology (25), pp. 1-19.

Hepsø, V., Monteiro, E., and Rolland K. 2009. “Ecologies of e-Infrastructures”, Journal of the Association of Information Systems, (10), pp. 430-446

Hirschheim, R., and Newman, M. 1991. “Symbolism and Information Systems Development: Myth, Metaphor and Magic,” Information Systems Research (2:1), pp. 29-62.

Johannessen, L. and Ellingsen, G. 2009. “Integration and Generification—Agile Software Development in the Healthcare Market,” Computer Supported Cooperative Work (18), pp. 607–634.

Jansen, A. and Nielsen, P. 2005. “Theorizing Convergence: Co- Evolution of Information Infrastructures”, Scandinavian Journal of Information Systems, (17:1), pp. 67-100

Larsen, M. and Myers, M. 1999. “When success turns into failure: a package-drive business process re-engineering project in the financial services industry,” Journal of Strategic Information Systems (8), pp. 395-417

Lucas, C. H., Walton, E. J., and Ginzberg, M. J. 1988. “Implementing Packaged Software,” MIS Quarterly (12:4), pp. 537-549.

Lungu, M., Lanza, M., Gîrba, T., and Robbes, R. 2010. “The Small Project Observatory: Visualizing software ecosystems”, Science of Computer Programming, (75), pp. 264-275

Lyytinen, K. and King, J. 2006. “Standard Making: A Critical Research Frontier for Information Systems Research,” MIS Quarterly (30), pp. 405-411.

Manikas, K. and Hansen K. 2013. “Software Ecosystems – a Systematic Review”, The Journal of Systems and Software, (86), pp. 1294-1306

Monteiro, E., Pollock, N., Hanseth, O., and Williams, R. (2013). “From Artefacts to Infrastructures,” Computer Supported Cooperative Work (22), pp. 575–607.

Monteiro, E. 2000. "Actor-Network Theory and Information Infrastructure." Pp. 71-83 in From Control to Drift: the Dynamics of Corporate Information Infrastructures, edited by C. Ciborra. Oxford: Oxford University Press.

Orlikowski, W., and Iacono, S. 2001. "Research Commentary: Desperately Seeking the "IT" in IT Research - a Call to Theorizing the IT Artifact." Information Systems Research, (12:2), pp. 121-134.

Pollock, N., Williams, R., and D’Adderio, L. 2007. “Global Software and Its Provenance: Generification Work in the Production of Organizational Software Packages,” Social Studies of Science (37), pp. 254-280.

Pozzebon, M., and Pinsonneault, A. 2005. “Global-local Negotiations for Implementing Configurable Packages: The power of Initial Organizational Decisions,” Journal of Strategic Information Systems (14), pp. 121-145.

Soh, C., and Sia, S. K. 2004. “An Institutional Perspective on Sources of ERP Package-Organization Misalignment,” Journal of Strategic Information Systems (13), pp. 375-397.

Sommerville, I. et al. 2012. “Large-scale complex IT systems,” Communications of the ACM (55:7), pp. 71-77.

Swanson, E. B. 1988. Information System Implementation: Bridging the Gap Between Design and Utilization, Homewood, IL: Irwin

Walsham, G. 1993. Interpreting Information Systems in Organizations, Chichester, UK: Wiley.

Walsham, G. 2001. Making a World of Difference: IT in a Global Context. Chichester: Wiley.

Walsham, G. 1995. “Interpretive Case Studies in IS Research: Nature and Method,” European Journal of Information Systems (4:2), pp. 74-81.

Yin, R. 2003. Case Study Research: Designs and Methods. Thousand Oaks, California: SAGE Publications.

Yeow, A. and Sia, S. 2008. “Negotiating ‘‘best practices’’ in package software implementation,” Information and Organization (18), pp. 1-28.