Identifying the Concept of Modularity in IS/IT Outsourcing Cases:
Some Empirical Evidence

Shahzada Benazeer
University of Antwerp, Belgium

Jan Verelst
University of Antwerp, Belgium

Philip Huysmans
University of Antwerp, Belgium

INTRODUCTION

Globalization of the world economy and digitalization has accelerated the advancements of information systems and information technology (IS/IT). IS/IT services are often outsourced to external partners for multiple reasons, but the main drivers are savings in cost, access to specialized expertise and technology, and focus on core competences. Outsourcing of IS/IT projects became a common practice among contemporary organizations in developed and in emerging economies. Literature suggests over 94% of '*Fortune 500*' companies are outsourcing at least one major business function (Modarress, Ansari, & Thies, 2014). Despite the prevalence and long experiences of CIO's in IS/IT project outsourcing, the failure of such projects is very common. The literature suggests that at least one in three projects was considered a failure and many projects were delayed, ran over budget, and were not able to meet their pre-defined targets (Delens, Peters, Verhoef, & Van Vlijmen, 2016; Jabangwe, Smite, & Hesbo, 2016; Schmidt, Zoller, & Rosenkranz, 2016; Wojewoda & Hastie, 2015). A pertinent question deals with how IS/IT project outsourcing failure may be addressed. So far, the literature includes many suggestions offered by both scholars and practitioners. Peterson and Carco (1998) suggested to streamline operations and 'fix the problem' before outsourcing IS/IT services. Various suggestions were introduced: the interested reader is referred to (1) Lambert, Emmelhainz, and Gardner (1999) who introduced their '*Partnership Model*'; (2) Greaver (1999) who formulated '*seven steps to successful outsourcing'*; (3) Logan (2000) who proposed two solutions in order to avoid failure in IS/IT project outsourcing. She suggests firstly, diagnosing the relationship from both sides of the contract and secondly, engaging agency theory to help design the types of contracts and relationships necessary to provide and support an environment of trust; (4) Lee (2001) who suggested knowledge sharing; (5) Rottman (2008) who elaborates on the importance of 'knowledge transfer'; (6) Harris, Herron and Iwanicki (2008) who stressed the importance of a high quality 'service level agreement' (SLA); (7) Karimi-Alaghehband and Rivard (2012) who proposed a model of IS/IT outsourcing success grounded in dynamic capabilities perspective; (8) Ishizaka & Blakiston, (2012) who proposed the *"18 C's model"* for a successful long-term outsourcing arrangement; and (9) Zheng and Abbott (2013) who argued that reconfiguration of organizational resources is vital to be successful in outsourcing. Despite the introduction of such remedies, the empirical

DOI: 10.4018/978-1-6684-7366-5.ch034

research referred to above continue to attest to the high failure rate of IS/IT project outsourcing. It seems that these remedies, if used, turned out to be partially successful at best. The remainder of this paper is structured as follows. The second section will describe in brief the literature review and the third section will describe the theoretical lens considered for this study, and at the end of this section, research questions are formulated. Afterwards, in the fourth section the adopted methodology will be discussed. In section five, case analysis, findings, and the reflections from the authors will be presented. In the sixth section, conclusions and in the seventh section future research directions will be discussed. The eighth section will present a glossary of the terminologies and concepts used in this chapter. The ninth section will suggest the readers some additional literature for further reading.

LITERATURE REVIEW

In the following some of the findings from a very extensive literature review on IS/IT outsourcing are described. Literature suggests that IS/IT project outsourcing is a complex maneuver (e.g., Aron, Clemons, & Reddi, 2005; Beulen & Ribbers, 2003; Cohen & Young, 2006; Hecker & Kohleick, 2006; Jacques, 2006; Nauman, Aziz, & Ishaq, 2009). Discussion in the introduction section highlights the high failure rate issue in IS/IT project outsourcing and remedies proposed in literature. The practitioners are also trying to address this issue by applying trial-and-error approach on a case to case basis. Interestingly, in the proposed remedies, none of them considered 'complexity' as a factor that might require attention in order to control the high failure rate. As literature suggests that the 'complexity' in IS/IT project outsourcing is a factor which requires due attention, hence digging further in literature is warranted to find out how the complexity issue is dealt with in other fields. Simon's article *"The Architecture of Complexity"* (1962) describes how one of the two watchmakers (Hora and Tampus) was dealing with a complex system consisting of about one thousand parts (i.e., assembling watches). Literature also describes how a huge complex system, such as Boeing 787 'Dreamliner', is manufactured in modules by more than hundred global suppliers (Tang & Zimmerman, 2009). Further insights gained from the literature about how Daimler-Benz managed complexity by decomposing the 'smart car' in seven modules assembled onsite by seven different suppliers (Van Hoek & Harrison, 2003). But compared to an automobile, Boeing '787 Dreamliner' was a much more complex product. On average, an automobile consists of 15-20 thousand parts whereas the '787 Dreamliner' consists of about 2.3 million parts.

Due to the volatile business environment, contemporary organizations are under immense pressure to achieve greater agility and flexibility in order to adapt to the ever-changing business environment. In this emerging volatile and ever-changing situation, contemporary organizations are splitting up their IS/IT systems, outsourcing agreements, and organizational structures into modules. When IS/IT systems are split into many modules, it offers greater agility and flexibility to customer organizations to decide which module/s will remain in-house and which module/s can be outsourced. Moreover, the customer organizations get the options to decide to outsource all the modules to a single vendor or to multiple vendors. Hence, the modular structure of IS/IT systems, on one hand, offers flexibility to outsource IS/IT services to multiple vendors at a competitive price, and on the other hand, if necessary, it offers flexibility to replace vendors or even back-source (reversibility) the IS/IT services because the theory suggests that in a truly modular system, modules can be used as *'black-box'* or in a 'plug *& play'-way* (Sako, 2005). In a modular environment, a customer organization may use the software as service (SaaS) or on-demand software. For instance, a study which analyzed 22,031 IS/IT contracts signed during a period of 20 years (1989 to 2009), found that highly modularized projects are more likely to be multi-

vendor outsourced (Bapna, Gupta, Ray, & Singh, 2013). Most importantly, the findings from the literature review suggest that in software development and/or business applications (a highly complex task), the use of modularity is prevalent. For instance, some modularity aspects such as 'interface', 'encapsulation' or 'information hiding', 'separation of concerns', and loose coupling, etc. are widely used in software development projects (Baldwin & Clark, 2000; Benazeer, 2018; Sanchez & Shibata, 2021). The IS/IT project outsourcing is often dealing with the software and/or business application developments. These relationships between complexity, modularity, and IS/IT project outsourcing and the insights from the literature guided the authors to think about the potential use of the concept of modularity as an ideal theoretical lens in the context of IS/IT project outsourcing.

THEORETICAL BACKGROUND AND RESEARCH QUESTIONS

In order to study the application of the modularity concept in the context of IS/IT outsourcing, a broad, exploratory investigation for phenomena in the context of IS/IT outsourcing has been conducted that can be interpreted as modular structures. These phenomena could be instances, examples or counter-examples of modularity in a wide-variety of aspects of IS/IT-outsourcing, both product- and process-oriented, both at the technical and/or at the non-technical (organizational) level. Product-oriented refers to the artefacts under production in the context of IS/IT-outsourcing, and could include software and software specifications, and modularity aspects such as coupling and interfaces. Process-oriented refers to the production process of artefacts, where IS/IT-outsourcing could be interpreted as modularity in terms of tasks shifting between two teams (resources) and organizations. The technical level refers to modular structures in software (from specifications to the programming code), whereas the non-technical (organizational) technical level refers to the possibility of interpreting IS/IT-outsourcing as organizational modularity in the sense that two organizations collaborate and communicate based on an SLA (interface). In addition to identifying instances, examples and counter-examples of modularity, an important task would be to obtain indications of their relevance or importance in terms of the IS/IT outsourcing project. This relevance or importance can be derived in multiple ways, including: first, instances/examples/counterexamples could be unimportant in the sense that they have little or no impact on the efficiency, effectivity, success or failure of the project. This kind of outcome from the case studies will nullify the assumptions that the concept of modularity may be relevant or important in the context of IS/IT project outsourcing. On the other hand, they could be linked to known issues or success factors in the project, which makes their relevance or importance more likely. Second, if instances/examples/counterexamples are related to design rules, design principles or theories regarding modularity, they could derive relevance from these theoretical foundations. This kind of outcome from the case studies will support the assumptions that the concept of modularity may be relevant or important in the context of IS/IT project outsourcing. For example, a known violation of a modularity design rule is likely to have, based on its theoretical grounding, an a priori negative impact on the modularity aspects of the products and processes that it is a part of. In this sense, the theoretical grounding establishes a certain measure of relevance or importance of the instance/example/counter-example. In order to pursue the abovementioned research goal, the research questions are formulated as follows:

RQ1: *Which instances (examples, counter-examples) of the use of modularity in the context of IS/IT outsourcing can be identified?*

RQ2: *How can the relevance and/or importance of these instances (examples, counter examples) for IS/IT outsourcing project be assessed?*

METHODOLOGY

This study is intended to get better insights of a new phenomenon using the lens of the concept of modularity. An interview-based descriptive, qualitative, multiple case study research approach has been adopted. As the phenomenon of IS/IT project outsourcing has rarely been studied using the lens of modularity, a descriptive case study approach is suitable. Literature review indicates that a descriptive case study is suitable if (1) the study is focused on the contemporary issues; (2) if it is a phenomenon where the boundaries between the phenomenon and context are not clearly defined (Yin, 2014). The present IS/IT outsourcing cases meet both conditions. A qualitative case study method is considered as the most suitable approach in getting answers for the research questions. Benbasat, Goldstein, and Mead (1987) describe three reasons why case study research is a viable information systems research strategy. First, the researcher can study information systems in a natural setting, learn about the state of the art, and generate theories from practice. Second, the case study method allows the researcher to answer 'how' and 'why' questions, that is to understand the nature and complexity of the processes taking place. Third, a case study approach is an appropriate way to research an area in which few previous studies have been carried out. Repeating an experiment more than once helps to strengthen the validity and reliability of research findings. Similarly, in a case study research, multiple case studies can strengthen the validity and reliability of research findings (Benbasat, Goldstein, & Mead, 1987; Yin, 2014). Before visiting the organizations, a thorough desk research was conducted to have some basic idea about the organizational structures and outsourcing practices. Through this approach, it became easier to understand the IS/IT architecture and the external partners who were offering IS/IT solutions to these organizations. The authors made several on-site visits to these organizations. Several sessions of open-ended, semi-structured and exploratory interviews were conducted. In addition to primary data collected from the interviews, some data were also collected from other sources (e.g., documentation, archival records, and media outlets).

ANALYSES OF THE CASES

In this section, two IS/IT project outsourcing cases are discussed, which have been analysed using primary data in order to investigate how modularity can be applied to IS/IT outsourcing. The first case focuses on one of the biggest service companies in its sector in Belgium, which is involved in a single-vendor IS/IT outsourcing project and the second case concerns a Belgian financial institution, which is involved in a multi-vendor IS/IT outsourcing project. In the following, both cases are discussed using the following steps in a systematic approach: first, the modular structure of the problem domain is made explicit. Most importantly, the identification of modules is addressed. Second, the relevant modularity aspects are selected. Third, the resulting modularity requirements are listed, and fourth, the absence of modularity characteristics is discussed in the context of violation or non-conformance of the modularity requirements. The analyses of cases have four sub-sections. First, the case in hand is introduced and then findings of the analysis is described. Next, analysis is done and then based on analysis and findings, a reflection sub-section describes some comments from the authors. Finally, a table summarizes the findings.

Case 1 Introduction: A Belgian Service Sector Organization

The selected case deals with a vendor organization referred to as *'Alpha'*, and a customer organization referred to as '*Omega*'. '*Alpha*' was regarded as a competent service provider. '*Omega*' was one of the biggest service companies in its sector in Belgium. The IS/IT outsourcing project involved managing and maintaining the entire IS/IT systems of '*Omega*'. The total number of employees of '*Omega*' was 'x', of which 'y' numbers were highly skilled employees who were managing and maintaining the IS/IT systems since long. However, recently it was decided to outsource the entire IS/IT systems to '*Alpha*'. The main motivation of the IS/IT outsourcing was cost reduction. As part of this outsourcing contract, almost all of the IS/IT headcounts were transferred from the '*Omega*' organization to the '*Alpha*' organization with job guarantees for a certain period. Those people were highly skilled IS/IT experts and were well paid due to their long experience. The contract period was of medium terms and at the time of the interviews a 2nd year was running. This IS/IT outsourcing project has been selected due to the special nature of this IS/IT outsourcing project deal. This IS/IT project outsourcing deal was about 'total outsourcing' which is defined by Lacity and Willcocks (2009, p. 5) as *"the decision to transfer the equivalent of more than 80% of the IT budget for IT assets, leases, staff, and management responsibility to an external IT provider"*.

(N.B. Due to confidentiality reasons and to maintain anonymity, fictitious names are used, and some information is masked).

Findings

Analyzing the case, it has been revealed that some decisions had been taken by the customer organization that might have led to the difficulties of the IS/IT outsourcing project. Concerning the service level agreement (SLA), the informant stated that:

"The contents of the deal (SLA) are determinant of whether the outsourcing goes well or not".

Even though the statement is a valid one, it does not facilitate an understanding of why (some) decisions were made, or why the problematic consequences occurred. In order to investigate in depth and to find the root causes, it is worthwhile to analyze the SLA from a modularity perspective. In analyzing the case, some flaws regarding the SLA were identified. For instance, incongruent with modularity, the SLA contained 'hidden dependencies'. In addition to the SLA, other violations of the principles of modularity were found as well. The analysis of the case, therefore, follows two recurring steps. First, to adequately identify a modular structure in a certain part of the case and then, requirements suggested by the concept of modularity for that structure, are described in a subsection *'Identifying the modular structure and requirements'*. Second, the description of the presented case illustrates how violations or non-conformance to the modularity requirements occurred under a subsection *'Assessing the modularity requirements'*. Obviously, any violation or non-conformance of modular design principles may, at least partially, contribute to the underperformance of IS/IT outsourcing initiatives.

Case 1 - Analysis #1: IS/IT Systems

De*pendency* is the degree to which a module relies on other modules in order to function and *coupling* is a measure of the dependencies between modules (Van der Linden, Mannaert, & De Bruyn, 2012).

i) Identifying the modular structure and requirements

From a modularity perspective, the system in scope is the IS/IT system of organization '*Omega'*, and within this modular structure, different IS/IT services are conceived as modules. This configuration is referred to as *modular structure MS1.1.* In this part of the analysis, the focus is on the coupling aspect of the concept of modularity. A good modular design should consist of 'low/loosely coupled' modules which are described by Simon (1962) as *'nearly decomposable systems'*. A low/loosely coupled modular system facilitates agility, flexibility, and evolvability in a changing environment (Sanchez & Mahoney, 1996). A non-agile and non-flexible highly/tightly coupled system inhibits change and therefore is a violation of the modularity requirements. Hence a loosely coupled modular system that facilitates agility and flexibility is referred to as *modularity requirement MR1.1*.

ii) Assessing the modularity requirements

The presence of undefined (hidden) inter-modular dependencies was one of the reasons that the informant during the interview labeled the IS/IT systems of '*Omega*' as '*spaghetti*', '*cobweb*' and '*usine à gaz*'. In one occasion the informant expressed the following:

"In our cobweb, everything works with chains. A task starts with one machine, processed by 2nd, 3rd, and will end in the 'n' machine. If a problem occurs in any one of these chains (machines), the entire process is blocked". The above excerpt draws a picture of tightly coupled systems. The informant further said that:

"Our (IS/IT) systems are not independent (loosely coupled) of one another, it is like a cobweb or 'usine à gaz'. While explaining the IS/IT outsourcing contract, the informant said that:

"It was a complete usine à gaz and at the technical level, it was almost impossible to split. There are too many connections which are dependent on one another".

These excerpts in the above paragraphs confirm that the IS/IT system of '*Omega*' was tightly coupled, hence it can be concluded that *modularity requirement MR1.1* mentioned in section (i) was not met.

Case 1 - Analysis #2: SLA

Modules should communicate with one another through interfaces (Langlois, 2002). An interface is a common boundary where direct contact between two modules occurs and where these two modules communicate with each other. The interface is a virtual or physical document where the rules of interaction among modules are exhaustively and unambiguously documented. The interface describes the inputs required by a module to perform its part of the functionality, and the output it will provide to its external environment (which includes other modules in the system). In the context of IS/IT project outsourcing, the SLA can be conceived as an interface between two modules, the vendor and the customer and this is supported by the literature. For instance, at the industry level, interfaces often consist of regulatory frameworks, rules, standards, and technical specifications that allow different players to connect (Jacobides, Knudsen, & Augier, 2006). Interfaces in services can include people, information,

and rules governing the flow of information (Voss & Hsuan, 2009). The importance of an SLA relating to the success of the IS/IT project is recognized and understood by the informant.

i) Identifying the modular structure and requirements

The outsourcing collaboration is the system in scope and within this modular structure, the organization '*Alpha*' and the organization '*Omega*' are conceived as modules. This configuration is referred to as *modular structure MS1.2.* The SLA serves as the interface connecting both organizations. To function adequately, the interaction between modules '*Alpha*' and '*Omega*' should be exhaustively and unambiguously documented in the interface. As far as the SLA is concerned, responsibilities of each module, rights of each module, and the relationships between modules are to be described in detail. In the context of IS/IT project outsourcing, the SLA essentially provides an interface between the vendor and the customer. Hence, all the interactions and settlements between modules '*Alpha*' and '*Omega*' should be conducted through the interface (SLA) and this requirement is referred to as *modularity requirement MR1.2*.

ii) Assessing the modularity requirements

As long as the highly skilled former employees of '*Omega*' were working for '*Alpha*', no major problems were reported. But '*Alpha*' started replacing those highly skilled people out and problems started to surface. Although it was stated in the SLA that the '*Omega*' would get similar services as it was used to get from the in-house team, the actual situation seems to be different. The following excerpts are highlighting the actual situation:

"It was stated (in the SLA) that we would get similar services". The informant further said that:

"Probably, there is something behind. Why they are not delivering, why? Are they not capable or is it something financially not interesting for them to deliver in time?".

The '*Omega*' team did not include several items in the SLA and as a result, they have to ask for extra services from the '*Alpha*' team for which the '*Alpha*' team charges them extra. As a result, the cost reduction motivation was overshadowed. An example can be given about the incomplete SLA from the following excerpt:

"We have to ask for extra things (services), it was not calculated in the predicted cost reduction".

The service delivery situation became so uncertain that the service managers from the '*Omega*' team had to travel regularly to the site of the '*Alpha*' organization in order to explain the priorities of '*Omega*' team, and to explain the '*Alpha*' team what they needed to do in order to deliver in time. At some point, it seems that the urgency and frustration triggered to ignore the SLA which is reflected in the following excerpt:

"Our service managers are physically traveling 2-3 times a week to the vendor in order to explain to them what the priorities are and what they need to do, jamais-vu". Later the informant added that:

"I don't think that the SLA is important right now, it just has to work".

Although the importance of a well-defined SLA is recognized by the informant, probably, this realization came too late. The above excerpts illustrate that the interface (SLA) was weak, vague, ambiguous, incomprehensive, inexplicit, and not well defined; therefore, the *modularity requirement MR1.2* mentioned in section (i) was not met.

Case 1 - Analysis #3: Change of Team

Change is inevitable within organizations and accommodating change poses a challenge. The following discussion is about highly skilled IS/IT experts who were transferred to the '*Alpha*' organization. In analysis #3, non-technical root causes of failure are dealt with. The team's composition can be interpreted and explained in terms of modular structures (Huysmans, De Bruyn, Benazeer, De Beuckelaer, De Haes, & Verelst, 2014). Furthermore, Terlouw (2011, p. viii) states that, *"modules can comprise humans and/or software systems"* and in addition, Dietz (2006, p. 81) proposed a method to identify modular actor role structures and thereby asserts that *"an enterprise is constituted by the activities of actor roles, which are elementary chunks of authority and responsibility, fulfilled by subjects"*. The main purpose of transferring IS/IT experts from '*Omega*' to '*Alpha*' was cost savings. The IS/IT experts had had long experience of working in blue-chip companies and as a result, they were very expensive people. '*Alpha*' replaced those highly skilled and expensive people by younger and less experienced people in order to reduce cost. '*Alpha*' succeeded in cost savings but failed to deliver the services, although it had been mentioned in the SLA that '*Omega'* would get at least the same level of services as it used to get by in-house experts.

i) Identifying the modular structure and requirements

From a modularity perspective, the system in scope is the group of highly skilled IS/IT experts that once belonged to the '*Omega*' organization, but which was transferred to the '*Alpha*' organization with job guarantees for a certain period. Within this modular structure, a highly skilled individual employee is also conceived as a module. This configuration is referred to as *modular structure MS1.3.* The focus of the analysis is on the '*substitution*' operator which is part of the modularity concept (i.e., a modular operator). In the following, some examples from modularity literature explain that substituting a module with another should guarantee improved or at least same functioning of the system. For instance, it has been stated that, *"substituting an older version of a module with the newer version should ameliorate the overall performance of the system"* (e.g., Huysmans, De Bruyn, Benazeer, De Beuckelaer, De Haes, & Verelst, 2014, p. 4418). Baldwin and Clark (2000, p. 262) suggest that *"The substitution operator allows a designer (or user) to swap one module of the system for a better version of the same module"*. Furthermore, Terlouw (2011, p. viii) asserts that *"the modular operators are the actions that may change existing structures in a well-defined way in order to enhance the efficiency of the system"*. The substitution modular operator can be applied successfully and relatively easily if all module versions adhere to the same interface and no undocumented or hidden inter-modular dependencies are present. If the interface is changed, or the dependencies of the modules are not made explicit, the application of the substitution modular operator is not without risk; one risk is that applying the substitution operator disrupts the working of the system and may trigger couplings and ripple effects. In a well-designed modular system, applying the modular operator '*substitution*' should not impact the existing structure negatively and this requirement is referred to as *modularity requirement MR1.3*.

ii) Assessing the modularity requirements

As mentioned, the conditions to successfully replace a module are 'same interface' and 'no unidentified or undocumented dependencies'. Replacing modules in a system with undocumented/hidden dependencies is a risky maneuver and success in IS/IT outsourcing may not be guaranteed. As it has been observed, the IS/IT system of '*Omega'* consisted of many undocumented dependencies and the knowledge about undocumented dependencies was inherent in the older versions of modules (former '*Omega*' employees). When the modules were replaced, the knowledge of the undocumented dependencies was also lost. As a result, substituting modules with newer versions was negatively affecting the efficiency of the project, which can be observed from the following excerpt:

"Now the circumstances have changed. Now when I contact 'Alpha' organization, I can't find my ex-colleagues anymore. Change of people triggers changing the circumstances. The level of knowledge and working practices of new incoming people are inferior comparing to my ex-colleagues". Furthermore, the informant said that:

"Our contract with 'Alpha' has just passed more than a year and most of our highly skilled colleagues were replaced by the younger and less experienced people".

Change of people not only caused delays in delivering services but in some cases, it was much more complex as the knowledge of the outgoing peoples was not retained. The following excerpt briefly explains the situation:

"Alpha took the entire spaghetti of 'Omega' intact and they do not have adequate knowledge about the legacy of 'Omega'. The team of 'Alpha' does not know how to decouple it as some parts of this legacy is recorded in specs and manuals, but some parts are 'shadow IT'. The problem becomes more complex as many authors of that shadow IT have left the organizations of 'Alpha' and 'Omega'. Some systems are still working but people don't know how they work".

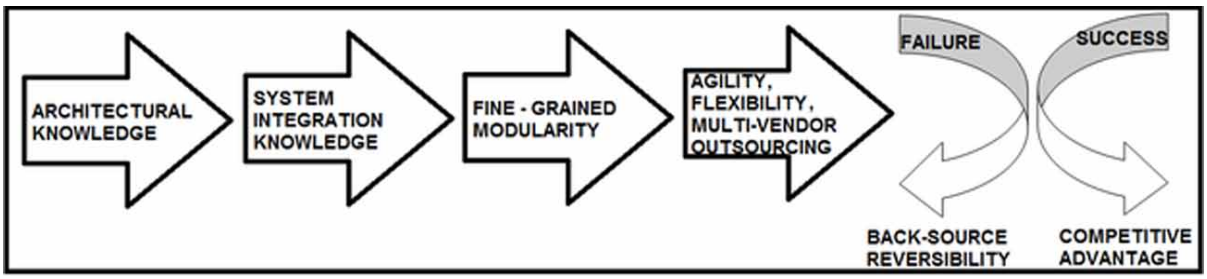
The above excerpt illustrates that applying the modular operator *'substitution'* resulted in problems. This led to delays in deliveries and the service managers of '*Omega*' have to visit the '*Alpha*' site in order to explain what to do and how to do. Therefore, in this situation and in the context of this case, it can be concluded that the *modularity requirement MR1.3* was not met.

Reflection

The third analysis of this case is about team composition, which is a non-technical issue but very often observed in the IS/IT outsourcing projects. Change of people or a team is a recurring event in many IS/IT outsourcing projects. There is a potential concern that the vendor organizations place their highly skilled people at the frontline during the negotiation or at the start of the project. Later, they can potentially replace those highly skilled people by less experienced and low skilled people. This could happen for two reasons: firstly, as those highly skilled people are few in numbers, so the vendor organization need to assign them again at the frontline to get another new project. Secondly, less experienced and low skilled people are cost effective. In this case, as a part of the outsourcing deal, almost all of the highly skilled people of customer organization *'Omega'* were transferred to the vendor organization *'Alpha'* with job

guarantees for a certain period. The literature argues that knowledge plays a key role in order to modularize a system as knowledge about interdependencies is crucial in designing such a system (Tee, 2009). For instance, in-depth knowledge about interdependencies among design parameters of different modules is a requirement in order to map the design structure matrix (DSM). But the requirement of specialized knowledge is significantly reduced once a modular system is perfectly designed and transformed as a *'black box'*. Nevertheless, even in a situation where everything is functioning well, customer organizations always need to keep some people possessing architectural and system integration knowledge in order to increase the ability to more precisely spell out contract terms and to effectively monitor and supervise vendors and at the same time increasing their relative bargaining power (Tiwana & Bush, 2007). Moreover, as the failure rate in IS/IT project outsourcing is significantly higher, it is indispensable to use the knowledge of skilled IS/IT experts to reverse the project smoothly and successfully (i.e., back-sourcing or switching vendors). The term 'reversible' or 'back-sourcing' denotes the possibility of a customer organization (in any unanticipated situation), to transfer back the operations in-house from the vendor (supplier). Sako (2005) argues that in a 'pure modular' case, interfaces are standardized and are reversible. Hence, truly modular outsourcing practice is conceived as being reversible and may potentially maximize growth, flexibility, and agility (Cohen & Young, 2006). The high failure rate in IS/IT project outsourcing and particularly the situation of this case is interesting in the context of Mannaert, Verelst & De Bruyn's (2016) assertion that most current modular systems in use are neither truly modular nor reversible because they exhibit only limited evolvability.

Figure 1. Possible role of knowledge in modularity and IS/IT project outsourcing.



The findings of this case support the point of view (figure 1) that it is indispensable for any contemporary organization to possess in-house, the architectural knowledge and the system integration knowledge of IS/IT systems in order to get competitive advantages from modularization and outsourcing. An organization with in-house knowledge and expertise is capable of designing the IS/IT systems in a modular way (i.e., respecting the rules prescribed by the concept of modularity). Once the entire IS/IT system is modularized in fine-grained modules (as smaller modules as possible), this gives greater agility and flexibility to the customer organization either to keep some or all the modules in-house or outsource to a single or multiple vendor. Moreover, if a vendor fails to deliver as promised, back-sourcing can be done with ease. Smooth, successful and painless reversibility is only possible when a customer organization possesses some skilled people with architectural and integration knowledge. Otherwise, the customer organization will have to face the same situation as *'Omega'* and will find itself to a certain degree in a *'vendor lock-in'* situation.

One of the main motivations of IS/IT project outsourcing is to concentrate more on the core competency and outsource the noncore activities. But by outsourcing the noncore activities, organizations

intend to forget those activities and anticipate that those noncore activities should work as '*black box*' or *'plug and play'* with minimum interventions. When a system is commoditized as a result of good modular architecture, a few skilled people are required to run or maintain it. Moreover, it is possible to avoid the risk of *'vendor lock-in'* when a system is commoditized and in addition, it gives greater agility and flexibility in choosing or switching vendors. Otherwise when the system is like '*usine à gaz*', or a *'white box'*, the organization becomes dependent on skilled peoples to run or maintain it. This implies the relationship underlying between the concept of modularity, IS/IT outsourcing, and knowledge. For instance, in this case, the main purpose of IS/IT project outsourcing was to reduce cost. But in order to provide services at low cost, the vendor *'Alpha'* needs to replace knowledgeable and expensive people who were inherited from the customer organization '*Omega*' (who has good knowledge about the system). As the system was not commoditized, remained as '*usine à gaz*', replacing those knowledgeable and expensive people was not possible. Since '*Alpha*' replaced those highly skilled people by under-qualified persons, everything went wrong. Following the reasoning from the modularity point of view, in this case, there was a contradiction in the initial setup. The goal is to cut cost, which is directly associated with commoditization, but this was never possible in the first place because the underline products/services were not commoditized. The following excerpts from the informant confirm this assertion:

"Although the purpose was cost reduction but at the end, it is becoming very expensive".

As many skilled people left the new organization ('*Alpha'*) after transferring from '*Omega*', the organization '*Omega*' became an empty company in terms of knowledge. In this emerging situation organization, '*Omega*' finds itself in a *'vendor lock-in'* scenario where reversibility or back-sourcing was no more possible. The following excerpts from the informant confirm this assessment: *"We cannot do a rollback (vendor lock-in)".*

Applying the concept of modularity in the context of IS/IT project outsourcing implies that in a good modular architecture the outsourcing contract should be reversible employing low efforts but, in this case, it was not possible as the organization '*Omega*' has lost the required knowledge to trigger back-sourcing. The findings of the analysis are summarized in the following table (table 1). In the first column of the table 1, 'modular structure' is described in capital letters and modules are described in italics. The second column describes the modularity aspect/s taken into consideration. The third column describes the requirement prescribed by the modularity literature and finally, fourth column illustrates the result based on the modularity conformance.

Case 2 Introduction: A Belgian Financial Institution

The second case of this study concerns a Belgian financial institution using multi-vendor outsourcing. *AB bank* focuses on private banking activities, implying that compared to traditional retail bankers, their customer base is smaller but wealthier. Further, the bank's activities include asset management and merchant banking services. Within the Belgian financial services industry, the organization can be considered as medium-sized in terms of the number of employees, number of clients, turnover, etc. While being a private bank in its core, the bank also welcomes investment clients with smaller budgets which can be served via an online investment portal. The portfolio management activities for bigger clients are offered through personal advice. Due to its relatively limited headcount consisting of 140 full-time employees in total, the IS/IT department of *'AB bank'* is rather small as well, consisting of ten full-time employees. The bank considers its IS/IT activities as operational and necessary but not as a strategic issue to obtain a

Table 1. Summary of the findings.

MODULAR STRUCTURE	MODULARITY ASPECTS	MODULARITY REQUIREMENTS	CONFORMANCE
Analysis # 1 **THE IS/IT SYSTEM** ***The IS/IT services of organization 'Omega' is conceived as a module.***	**Coupling**	**A loosely coupled modular system facilitates agility and flexibility.**	**Not met**
Analysis # 2 **OUTSOURCING COLLABORATION** ***Organization 'Alpha' and Organization 'Omega' are conceived as modules***	**Interface/SLA**	**Exhaustive, explicit, unambiguous, and well defined**	**Not met**
Analysis # 3 **HIGHLY SKILLED GROUP OF IS/IT PEOPLE** ***Highly skilled individual IS/IT employee conceived as a module.***	**Modular operator 'substitution'**	**The substitution operator can be applied successfully if modules have 'same interface' and do not have 'undefined or undocumented dependencies'**	**Not met**

competitive advantage. In that context, the modus operandi of multi-vendor outsourcing was chosen over the years. The main activity of the in-house IT team of the financial institution was therefore concerned with the integration of all outsourced activities as well as its general management (package selection, vendor negotiations, etc.). The informant is the head of the IS/IT department (CIO). While sketching the current situation of his department as well as the outlook for the future, the integration of the different (often externally acquired) applications was already indicated as a major concern. Literature suggests that 'systems integrating organizations' *know more than they make* (e.g., Brusoni, Prencipe, & Previtt, 2001). This means (as its illustrated in figure 1) firstly, that an organization should possess advanced levels of design, architectural and system integration knowledge in order to deal with multi-vendor IS/IT solutions and secondly, the IS/IT architecture should be sufficiently agile and flexible in order to 'mix and match' different IS/IT solutions offered by the multiple vendors. This part of the study intends to get some more insights using the lens of the concept of modularity, about how multi-vendor IS/IT outsourcing is managed by *'AB Bank'* and why the integration issue was so difficult to resolve. This case was a selective choice because the financial service sector in Europe is becoming highly digitalized, hence it is important to get some insights into this sector. Gewald and Dibbern (2005, p. 2) assert that *"One industry where digitization has dramatically altered the way in which business processes are carried out is the Banking Industry. Almost the entire portfolio of banking products is available in digital form and many services are now provided through the internet. The balance in a current account, an international payment, or the purchase of mutual funds is nowadays merely an electronic transaction which takes place in bits and bytes on a storage system within a corporate data center. Associated business processes like trade settlement or execution control are of an electronic nature as well"*.

(N.B. Due to confidentiality reasons and to maintain anonymity, fictitious names are used, and some information is masked).

Findings

Modularity literature suggests that modularity is inherently a recursive concept that can be applied at different levels (Baldwin & Clark, 2000). The analysis revealed two major levels at which modularity could be clearly applied to the case at hand. Therefore, issues relating to two levels are subsequently discussed at an inter-organizational level (the relationship between *'AB Bank'* and its IS/IT service providers) and at an intra-organizational level (the internal organization of *'AB Bank'*, such as the architecture of its different IS/IT applications and their integration). This is visually illustrated in figure 2. On the one hand, the figure depicts a general overview of the IS/IT system modules present within the case organization (the grey ovals indicate the internally developed and maintained applications). It has been observed that a large majority of the IS/IT applications (i.e., the white ovals) were outsourced to external parties. On the other hand, the figure shows that for these outsourced applications, a set of SLAs was agreed upon with a set of external IS/IT service providers. At the inter-organizational level, the findings reveal that the SLA was vague, ambiguous, incomprehensive, inexplicit, and not well defined. Ideally, an exhaustive SLA facilitates change and offers flexibility during the stipulated duration of the contract. At the intra-organizational level, it was interesting to note that the integration problem was not only technical or on a syntactical level, but equally semantic.

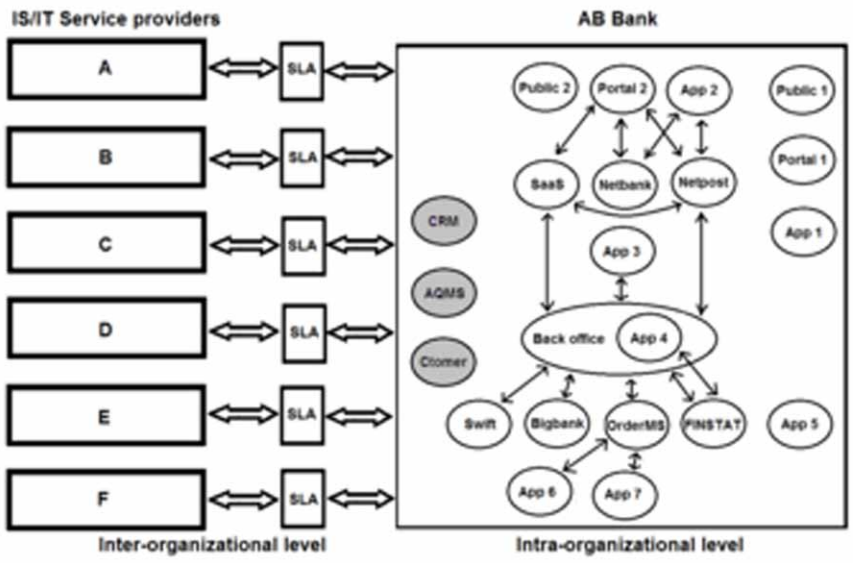
"Different systems use different concepts. The most difficult part is to match different concepts to each other. For me, the pain is in the interfacing part".

From the above excerpts, it becomes clear that the interfaces between the different applications within the IS/IT portfolio of *'AB Bank'* were often not exhaustive if they existed at all. However, the formulation of an analysis framework based on the modularity perspective could aid in detecting violations or non-conformance ex-ante. This case analysis adopted the same way as first case analysis of reporting for each of the identified modularity manifestations. The analysis of the case, therefore, follows two recurring steps. First, to adequately identify a modular structure in a certain part of the case and then, requirements suggested by the concept of modularity for that structure, are described in a subsection *'Identifying the modular structure and requirements'*. Second, the description of the presented case illustrates how violations or non-conformance to the modularity requirements occurred under a subsection *'Assessing the modularity requirements'*. Obviously, any violation or non-conformance of modular design principles may, at least partially, contribute to the underperformance of IS/IT outsourcing initiatives.

Case 2 - Analysis #1: The Inter-Organizational Level

In general, an IS/IT outsourcing deal concerns an agreement or contract between (mostly two) parties in which one party (the vendor) agrees to deliver certain services to another party (the customer). The outsourcing contracts (SLA's) need to be managed by good arrangements stipulating the roles and responsibilities of each of the involved actors as these deals are often highly complex and of crucial importance for both parties. First, it is clear that such SLA is crucial from a legal point of view. It has been explained earlier that many IS/IT outsourcing projects fail and may result in non-satisfactory relationships between the vendor and the customer which may sometimes even end up in a legal dispute. In such cases, obviously, the SLA serves as the starting point to analyze who has (not) fulfilled his or her responsibilities. However, the role of an SLA between two parties can be considered more broadly as well. For instance, based on literature it can be argued that it should include the set of people, information, and rules governing the

Figure 2. Modular structures identified: 'AB Bank' case.



flow of information between the parties (Voss & Hsuan, 2009). At the industry level, these interfaces often consist of regulatory frameworks, rules, standards, and technical specifications that allow different players to connect (Jacobides, Knudsen, & Augier, 2006). An interface is a common boundary where direct contact between two or more modules occurs and where these modules communicate with each other. The interface can be a virtual or physical document where the rules of interaction (dependencies, conditions) among modules are exhaustively and unambiguously documented. The interface describes the inputs required by a module to perform its part of the functionality, and the output it will provide to its external environment (which includes other modules in the system). In more general terms, the SLA can be regarded as the interface connecting two organizations which sets the rules between the vendor and the customer organization that governs the outsourcing relationship. SLA a crucial component of each and every IS/IT outsourcing engagement. Given the importance of an SLA in an IS/IT outsourcing context, questions were asked to the informant to get in-depth information about the SLAs adopted by the case organization during its outsourcing engagements. More specifically, questions were asked how these SLAs were established during the initiation of an outsourcing deal. Moreover, information was obtained regarding the extent to which the respondent believed that these SLAs were effective in providing a sufficient amount of guidance and coordination between the parties involved during the execution of the project. Questions were also asked about the evolution of such SLA's in case of long-term engagements: what if one of the parties wanted certain conditions in the contract to be adapted? (How) could this be done?

i) Identifying the modular structure and requirements

While focusing on the role of an SLA within an outsourcing project and considering an SLA as the interface between the two organizations involved. It has been mentioned above that an interface is a common boundary between modules which facilitates in managing the communication and interaction (input/output) between those modules. As the SLA should govern the rules and arrangements between parties, this clearly matches the definition of an interface. In that case, the IS/IT outsourcing collaboration

is the system in scope and the relevant modules are the vendor organization and the customer organization. This configuration is referred to as *modular structure MS2.1.*

Concerning *modular structure MS2.1*, as it has been explained earlier that a good interface between modules should be exhaustive and complete. Applied to the SLA, interpreted as the interface between the vendor and the customer, this implies that all services required by *'AB Bank'* from the vendor should be listed in the SLA (and other services than those embedded in the interface should not occur). This statement is referred to as *modularity requirement MR2.1.*

ii) Assessing the modularity requirements

It was remarkable to note that the SLA was mainly considered by the case organization as a 'necessary obligation' attached to the beginning of each IS/IT outsourcing initiative. The formulation of an IS/IT outsourcing contract was primarily regarded as a legal issue to be dealt with by the legal department and not to be consulted except in cases where judicial actions were required as it is stated by the informant:

"For me, a contract is something you make, you sign, and put it in a closet. You hope that you never need to look at it again. The moment you need that, it is because you have a problem".

Instead, the informant explained that *'AB Bank'* was counting - to a large extent - on the professionalism of the outsourcing partners and expected them to be reasonable. Stated otherwise, the 'legal' SLA was indeed realized via a formal contract whereas the actual or 'operational' SLA was mainly based on confidence and mutual trust, as the informant said about it:

"We have the contract; we have the SLAs but what was really missing are the operational things. Let's say, it's not about processes but more (about legal) procedures".

While in several situations such collaborations have succeeded due to personal contacts between people at the vendor and customer side, the respondent acknowledged this way of working had also clearly failed in numerous occasions, as the informant further added:

"(For instance) when we see a problem in a particular process and ask 'S' (pseudonym of a vendor) 'why is this problem not noticed by you?', the answer given was that they were not monitoring the process. We suggest the 'S' people that being a professional you should have monitored that problem. The 'S' people will reply that they have not been asked (by the SLA of 'AB Bank') to monitor that problem. [….] Both parties have done what is stated in the contract. Yes, but was that enough? No, probably not. It is not stated in the contract that we should make the design as to how the architecture (of the application) should look like and define it. But we expect them to operate the platform that runs the application".

From the above, it becomes clear that the description embedded within the SLA was by no means exhaustive. Ambiguous formulations were (often consciously) allowed within the contract as trust was considered to be the main driving force behind the collaboration. Therefore, it can be concluded that *modularity requirement MR2.1* was not met.

Case 2 - Analysis #2: Intra-Organization Level

At a fine-grained level, IS/IT project outsourcing concerns transferring certain responsibilities regarding a (set of) IS/IT applications from the client to the vendor. Clearly, within the client organization, those externally developed IS/IT applications should be integrated (both with internal systems and with systems from other vendors) so that they can collaborate with one another if required. As it has been aforementioned and illustrated (figure 2), *'AB Bank'* has adopted a multi-vendor outsourcing strategy encompassing several medium-sized applications and only a limited amount of applications developed and maintained by themselves. The arrows within the figure depict the most important interactions between the systems. Given the importance to *'AB Bank'* of managing this set of applications, during the interview, the conversation went in-depth on how the organization dealt with this particular configuration. More specifically, questions were asked about how the integration between these applications was established. Was this easy or problematic? Whose responsibility was this? And how was this taken into account during the different phases of the IS/IT outsourcing project (e.g., initiation, start-up, execution, etc.)?

i) Identifying the modular structure and requirements

The modular structure to study the communication and integration within the IS/IT systems of *'AB Bank'* can be easily identified. That is, each individual IS/IT application is a module per se. The informant discussed and elaborated the IS/IT application landscape within *'AB Bank'* (e.g., the different applications for the back office, front office, customer onboarding, etc.), with some applications being internally managed and some of them externally. As the organization did not distinguish subparts within each application, the application level is the lowest granularity level available when studying the integration issue in this case. When the informant was asked about the IS/IT systems of *'AB Bank'* which seems to be a configuration of the modular structure, the informant replied:

"Yes, it is like granularity. Indeed, we have different applications that work together. In the outsourcing part we have 'SaaS' type of outsourcing, we then have 'remote managed services'. 'ECM' is managing the hardware of operational stuff of 'X' machine (which runs the private banking system), and software is indeed managed and delivered by 'BLU'. The back office of the online portal is 'SaaS', and front services are running on 'managed services outsourcing'. The hardware is run by 'SR' but 'GL' developed the software".

Therefore, it is logical to consider the IS/IT application portfolio of *'AB Bank'* as the system, with every individual application being a module. This configuration is referred to as *modular structure MS2.2*.

Theoretical knowledge regarding modularity suggests that a well-designed modular system should have a clear and well-defined exhaustively documented interface. Based on this information, a set of design rules can be created which form boundary conditions with which the IS/IT applications have to comply (i.e., it describes a set of required inputs and outputs). Within these limitations, each IS/IT application can freely choose its specific implementation. Therefore, the existence of exhaustively documented inter-application interfaces referred to as *modularity requirement MR2.2*.

ii) Assessing the modularity requirements

During the interview and in-depth discussions about IS/IT applications and its integrations, the initial feeling was that the architecture looked rather complicated and questions came to the author's mind how integrations were managed. It was immediately noted by the informant that integration was an important IS/IT challenge within *'AB Bank'* as it was straightforward for him to enumerate a set of pertinent issues in this area:

"The integration is a challenge we have. We explored many issues in 'AB Bank' which are linked to the integration of applications. The integration challenges that we have are not really resolved".

It was an easy task for the informant to enumerate a set of examples of related issues. For instance, the informant stated that if a new customer is coming to open an account, the administrative employee needs to enter data manually in 7 different systems and that in some cases this number of systems can go up to 15. Or, if a customer likes to order a particular equity, the portfolio manager first has to look at the equity offers on a system 'A' and then needs to go to system 'B' to execute the order as no direct links between these two systems were established. Similarly, if a customer calls to *'AB Bank'* and asks to buy a certain amount of a particular stock, the portfolio manager will enter this request into system 'B'. System 'B' sends this request automatically to the broker's system (typically another Belgian bank). The request comes back to system 'B' confirming that the operation is executed at a particular price per share. The additional charges for this operation (e.g., commission, taxes, etc.) did not get incorporated in the invoice at that moment as it is not included within the interface between system 'B' and the external broker. In fact, the information about the additional charges is only known to *'AB Bank'* (and therefore its client) one day later. Stated otherwise, not all systems which can or should automatically interact were properly connected in the case of *'AB Bank'*. Furthermore, this did not even seem to be a real priority when asking about the process of vendor and application selection:

"When we select an application, the first things we look at are the functional requirements. Do they match with our business requirements? Then we look at the non-functional requirements. We look at things like, are we able to manage the operating systems, the database systems? But indeed, we don't look at the requirements in terms of what kind of interfaces do we want [….]".

Finally, it was interesting to note that the integration problem was not only technical or on a syntactical level, but equally semantic:

"Different systems use different concepts. The most difficult part is to match different concepts to each other. For me, the pain is in the interfacing part".

From the above excerpts, it becomes clear that the interfaces between the different applications within the IS/IT portfolio of *'AB Bank'* were often not exhaustive if they existed at all. Therefore, it can be concluded that *modularity requirement MR2.2* was not met.

Reflection

Aforementioned analysis of the case study was validated with the informant in the fourth phase of the interview. Questions were asked to informant whether present approach offered a useful way for him to look at some of the IS/IT challenges within *'AB Bank'* and if so, in what way. It was interesting to note

that the informant explicitly acknowledged that he indeed found the present perspective to be relevant and it was triggering him to think about certain things in a new and fresh way. In order to prevent possible bias and *"researcher pleasing"* behavior, the question was also asked for a more specific argumentation. Then, the informant, for instance, mentioned that he was not aware that their integration was a problematic issue to such large extent and that he was thinking about how he could incorporate ideas regarding design rules (which were currently absent) into his organization. The informant acknowledged that a necessary amount of trust combined with a more complete and operationally defined interface (SLA) was likely to improve their IS/IT project outsourcing collaborations:

"We didn't provide (ask this service) in the SLA and perhaps we should have thought about it in advance that the system should function as designed and that the response times are within appropriate limits. Perhaps we also put the SLA at a too high level. We should have gone into more detailed points. [. . . .] It is indeed more on operational levels that we didn't describe what we expected them to do. [. . . .] Ok, (now) I understand, if we would have this kind of detail description, we could easily challenge it and discuss it with the party to see where the differences are".

However, the actual realization of such contracts in practice did not seem straightforward in all cases. For example, listing all activities that should ever be done in an IS/IT outsourcing collaboration seems rather difficult as it is challenging to look ahead in this way:

"In an outsourcing contract, it is difficult to foresee what I need in six months or in one year. It is difficult to make it specific; therefore, it is also difficult to foresee it already in a contract".

In contrast, what might be realistic is to have an independent industry standard, in which the generally accepted best-practices for such an IS/IT outsourcing deals are listed. Contracts based on these standards could probably already partially mitigate this problem. The informant also agreed with the observation that the modular architecture resulting from the supply-based selection of packages (causing duplicate functionalities to arise) could be improved. It was indicated that probably, a more fine-grained modular approach was required to do this:

"Yes, indeed. Starting in fact from applications which are itself perhaps quite modular, but we don't get the advantage of modularity because we use it as a complete functional box and now, we are trying to cut pieces out of this complete box".

Finally, the informant acknowledged that following the principles of modularity concept, the integration of different IS/IT applications within *'AB Bank'* was far from optimal. *'AB Bank'* had the ambition to improve this situation in the future. However, also this was considered to be non-evident due to multiple reasons. For instance, being a small to a medium-sized financial institution with a limited number of customers, integration projects such as those related to the new customer registration process are very unlikely to obtain a sufficiently high priority. The informant additionally indicated that they also had some fundamental questions on how a good modular structure, in order to allow such integration, should be developed in the first place. While some basic and intuitive knowledge regarding modularity was present in the organization, the informant indicated that in such case it would be required for him and his organization to acquire more in-depth knowledge regarding modular systems and sound integration practices. The findings of the analysis are summarized in the following table (table 2). In the first column

of the table 2, 'modular structure' is described in capital letters and modules are described in italics. The second column describes the modularity aspect/s taken into consideration. The third column describes the requirement prescribed by the modularity literature and finally, fourth column illustrates the result based on the modularity conformance.

Table 2. Summary of the findings.

MODULAR STRUCTURE	MODULARITY ASPECTS	MODULARITY REQUIREMENTS	CONFORMANCE
Analysis # 1 **OUTSOURCING COLLABORATION** ***The vendor organization and the customer organizations are conceived as modules***	**SLA**	**Exhaustive, explicit, unambiguous, and well defined**	**Not met**
Analysis # 2 **IS/IT APPLICATION PORTFOLIO** ***Every individual application of IS/IT portfolio is conceived as modules***	**Interface**	**Exhaustive, explicit, unambiguous, and well defined applications and no overlapping**	**Not met**

CONCLUSION

This study provides interpretations of phenomena in IS/IT outsourcing projects based on the concept of modularity, as instances or examples of modular structures with indications of their importance based on links to theoretical frameworks, design principles or success/failure factors. These interpretations provide better insights to IS/IT project outsourcing researchers and can contribute to a richer understanding about the reasons why they are (not) successful. Answering the first research question, this study provides instances or examples of the role of modularity in the context of IS/IT outsourcing, at two levels; technical and organizational. Answering the second research question, these instances or examples can be linked to several theoretical frameworks, design principles and success/failure factors, providing further indications of their importance in IS/IT outsourcing projects. These instances or examples in the analyzed cases can be interpreted as violations of, or at least insufficient attention to, well-known design principles on modularity, thereby providing indications that they likely negatively impact the outcome of the project success by inducing undesired complexities. The global impression resulting from this study, is that in a highly digitalized environment, IS/IT outsourcing projects deal with several types of modular structures (technical, organizational). This implies that the aspects of these modular structures actually do play a role in IS/IT project outsourcing. The current cases have provided indications to this extent in the form of instances or examples. The analyzed cases indeed contain many violations against modularity (as well-known modularity design principles are not applied), or a lack of 'attention', which can be linked to theoretical frameworks, principles or success factors to illustrate their importance. In conclusion, the results of the analyses indicate that the concept of modularity seems to be certainly relevant in the context of IS/IT project outsourcing. This finding can be considered consistent with the literature on modularity in general and domain-independent frameworks such as Baldwin and Clark's modularity design rules (Baldwin & Clark, 2000), but this relevance seems to be underemphasized in most current research on IS/IT project outsourcing. Finally, the authors are not arguing that modularity

is the only or dominant factor determining the success of IS/IT outsourcing projects. Instead, the aim of this study was to explore the role of a factor that is in author's opinion often underexposed in the context of IS/IT project outsourcing, which in no way minimizes the role of other factors.

CONTRIBUTIONS AND FUTURE RESEARCH DIRECTIONS

This study provides interpretations of phenomena in IS/IT outsourcing projects based on the concept of modularity, as instances or examples of modular structures with indications of their importance based on links to theoretical frameworks, design principles or success/failure factors. These interpretations can contribute for the researchers to a richer understanding of IS/IT outsourcing projects, and the reasons why they are (not) successful. Aspects of this contribution include the wide range of areas where modularity can be applied and the variety of ways in which modularity influences IS/IT outsourcing projects (including organizational and technical), as well as the significant influence technical modularity issues seems to play, even though IS/IT project outsourcing is often considered from a non-technical, IT-management point of view. This study has provided indications of the role that modularity plays in IS/IT project outsourcing, in a wide range of domains, from organizational to technical, with implications on areas such as knowledge management and communications. In order to build on these indications and maximize the insights that can be gained from modularity in this context, the authors call for future research to provide more detail on the role of modularity and its potential to address the issues that are currently causing IS/IT outsourcing projects to fail. In other words, the authors call for the addition of modularity aspects as a complement to the current management-approaches to IS/IT project outsourcing, providing a combination of more management-oriented and more structure-oriented (i.e., modularity-oriented) approaches to provide a richer view of factors influencing the success of IS/IT outsourcing projects.

REFERENCES

Aron, R., Clemons, E. K., & Reddi, S. (2005). Just right outsourcing: Understanding and managing strategic risk. *Journal of Management Information Systems*, *22*(2), 37–35. doi:10.1080/07421222.2005.11045852

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: The power of modularity*. The MIT Press. doi:10.7551/mitpress/2366.001.0001

Bapna, R., Gupta, A., Ray, G., & Singh, S. (2013). *Specialization, integration, and multi-sourcing: A study of large IT outsourcing projects* [Paper presentation]. *International Conference on Information Systems. Reshaping society through information systems design*, Milan, Italy.

Benazeer, S. (2018). *On the feasibility of applying the concept of modularity in the context of IS/IT outsourcing* [Doctoral dissertation, University of Antwerp]. University of Antwerp Research Repository. https://repository.uantwerpen.be/docman/irua/199d4e/153326.pdf

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of Information Systems. *Management Information Systems Quarterly*, *11*(3), 369–386. doi:10.2307/248684

Beulen, E. J. J., & Ribbers, P. M. A. (2003). International examples of large-scale systems - theory and practice: A case study of managing IT outsourcing partnerships in Asia. *Communications of the AIS*, *11*, 357–376. doi:10.17705/1CAIS.01121

Brusoni, S., Prencipe, A., & Pavitt, K. (2001). Knowledge specialization, organizational coupling, and the boundaries of the firm: Why do firm know more than they make? *Administrative Science Quarterly*, *46*(4), 597–621. doi:10.2307/3094825

Cohen, L., & Young, A. (2006). *Multisourcing: Moving beyond outsourcing to achieve growth and agility*. Harvard Business School Press.

Delens, G. P. A. J., Peters, R. J., Verhoef, C., & Van Vlijmen, S. F. M. (2016). Lessons from Dutch IT-outsourcing success and failure. *Science of Computer Programming*, *130*(32), 37–68. doi:10.1016/j.scico.2016.04.001

Dietz, J. L. G. (2006). *Enterprise ontology: Theory and methodology*. Springer. doi:10.1007/3-540-33149-2

Gewald, H., & Dibbern, J. (2005). *The influential role of perceived risks versus perceived benefits in the acceptance of business process outsourcing: Empirical evidence from the German banking industry* (Working Paper Nr. 2005-9). J. W. Goethe University, E-Finance Lab.

Greaver, M. F. II. (1999). *Strategic outsourcing: A structured approach to outsourcing decisions and initiatives*. Amacom.

Harris, M. D. S., Herron, D., & Iwanicki, S. (2008). *The business value of IT: Managing risks, optimizing performance and measuring results*. CRC Press. doi:10.1201/9781420064759

Hecker, A., & Kohleick, H. (2006). Explaining outsourcing failure. *SSRN Electronic Journal*. Retrieved August 31, 2022, from https://ssrn.com/abstract=939411

Huysmans, P., De Bruyn, P., Benazeer, S., De Beuckelaer, A., De Haes, S., & Verelst, J. (2014). Understanding outsourcing risk factors based on modularity: The BSkyB case. *International Journal of IT/Business Alignment and Governance*, *5*(1), 50–66. doi:10.4018/ijitbag.2014010104

Ishizaka, A., & Blakiston, R. (2012). The 18C's model for a successful long-term outsourcing arrangement. *Industrial Marketing Management*, *41*(7), 1071–1080. doi:10.1016/j.indmarman.2012.02.006

Jabangwe, R., Smite, D., & Hesbo, E. (2016). Distributed software development in an offshore outsourcing project: A case study of source code evolution and quality. *Information and Software Technology*, *72*, 125–136. doi:10.1016/j.infsof.2015.12.005

Jacobides, M. G., Knudsen, T. R., & Augier, M. (2006). Benefiting from innovation: Value creation, value appropriation, and the role of industry architectures. *Research Policy*, *35*(8), 1200–1221. doi:10.1016/j.respol.2006.09.005

Jacques, V. (2006). *International outsourcing strategy and competitiveness: Study on current outsourcing trends: IT, business processes, contact centers*. Publibook.

Karimi-Alaghehband, F., & Rivard, S. (2012, December). *Information technology outsourcing success: A model of dynamic, operational, and learning capabilities* [Paper presentation]. *33rd International Conference on Information Systems (ICIS)*.

Lacity, M. C., & Willcocks, L. P. (2009). *The practice of outsourcing: From information systems to BPO and offshoring*. Palgrave Macmillan.

Lambert, D. M., Emmelhainz, M. A., & Gardner, J. T. (1999). Building successful logistics partnerships. *Journal of Business Logistics*, *20*(1), 165–181.

Langlois, R. N. (2002). Modularity in technology and organization. *Journal of Economic Behavior & Organization*, *49*(1), 19–37. doi:10.1016/S0167-2681(02)00056-2

Lee, J.-N. (2001). The impact of knowledge sharing, organizational capability and partnership quality on IS outsourcing success. *Information & Management*, *38*(5), 323–335. doi:10.1016/S0378-7206(00)00074-4

Logan, M. S. (2000). Using agency theory to design successful outsourcing relationships. *International Journal of Logistics Management*, *11*(2), 21–32. doi:10.1108/09574090010806137

Mannaert, H., Verelst, J., & De Bruyn, P. (2016). *Normalized systems theory: From foundations for evolvable software toward a general theory for evolvable design*. Koppa.

Modarress, B., Ansari, A., & Thies, E. (2014). The impact of technology transfer through foreign direct investment in developing nations: A case study in the United Arab Emirates. *International Journal of Economics and Finance*, *6*(7), 108–126. doi:10.5539/ijef.v6n7p108

Nauman, A. B., Aziz, R., & Ishaq, A. F. M. (2009). Information system development failure and complexity: A case study. In M. G. Hinter (Ed.), *Selected reading on strategic information systems* (pp. 251–275). IGI Publications. doi:10.4018/978-1-60566-090-5.ch017

Peterson, B. L., & Carco, D. M. (1998). *The smart way to buy information technology: How to maximize value and avoid costly pitfalls*. Amacom.

Rottman, J. W. (2008). Successful knowledge transfer within offshore supplier networks: A case study exploring social capital in strategic alliances. *Journal of Information Technology*, *23*(1), 31–43. doi:10.1057/palgrave.jit.2000127

Sako, M. (2005). Modularity and outsourcing. In A. Principe, A. Davies, & M. Hobday (Eds.), *The business of system integration* (pp. 229–253). Oxford University Press. doi:10.1093/acprof:oso/9780199263233.003.0012

Sanchez, R., & Mahoney, J. T. (1996). Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management Journal*, *17*(S2), 63–76. doi:10.1002mj.4250171107

Sanchez, R., & Shibata, T. (2021). Modularity design rules for architecture development: Theory, implementation, and evidence from the development of the Renault–Nissan alliance "common module family" architecture. *Journal of Open Innovation*, *7*(4), 1–22. doi:10.3390/joitmc7040242

Schmidt, N., Zoller, B., & Rosenkranz, C. (2016). The clash of cultures in information technology outsourcing relationships: An institutional logics perspective. In J. Kotlarsky, I. Oshri, & L. P. Willcocks (Eds.), *Shared services and outsourcing: A contemporary outlook* (pp. 97–117). Springer. doi:10.1007/978-3-319-47009-2_6

Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society*, *106*(6), 467–482.

Tang, C. S., & Zimmerman, J. D. (2009). Managing new product development and supply chain risks: the Boeing 787 case. *Supply Chain Forum: An International Journal, 10*(2), 74-86.

Tee, R. (2009, January). *Product, organization, and industry modularity exploring interactions across levels* [Paper presentation]. DRUID-DIME Academy Winter Ph.D. *Conference on Economics and Management of Innovation, Technology, and Organizational Change*, Aalborg, Denmark.

Terlouw, L. I. (2011). *Modularization and specification of service-oriented systems* [Doctoral dissertation, Delft University of Technology]. Gildeprint.

Tiwana, A., & Bush, A. A. (2007). A comparison of transaction cost, agency, and knowledge-based predictors of IT outsourcing decisions: A U.S.–Japan cross-cultural field study. *Journal of Management Information Systems*, *24*(1), 259–300. doi:10.2753/MIS0742-1222240108

Van der Linden, D., Mannaert, H., & De Bruyn, P. (2012, February). Towards the explicitation of hidden dependencies in the module interface. *Proceedings of the Seventh International Conference on Systems (ICONS)*.

Van Hoek, R., & Harrison, A. (2003). The Smart Car and smart logistics. In R. Johnston & ... (Eds.), *Cases in operation management* (pp. 316–326). Prentice Hall.

Voss, C. A., & Hsuan, J. (2009). Service architecture and modularity. *Decision Sciences*, *40*(3), 541–569. doi:10.1111/j.1540-5915.2009.00241.x

Wojewoda, S., & Hastie, S. (2015). *Standish group 2015 chaos report - Q&A with Jennifer Lynch*. Infoq. Retrieved August 20, 2022, from https://www.infoq.com/articles/standish-chaos-2015

Yin, R. K. (2014). *Case study research: Design and methods*. Sage Publications.

Zheng, Y., & Abbott, P. (2013, June) Moving up the value chain or reconfiguring the value network? An organizational learning perspective on born global outsourcing vendors. *Proceedings of the 21st European Conference on Information Systems*.

ADDITIONAL READING

Mannaert, H., De Bruyn, P., & Verelst, J. (2022). On the interconnection of cross-cutting concerns within hierarchical modular architectures. *IEEE Transactions on Engineering Management*, *69*(6), 3276–3291. doi:10.1109/TEM.2020.3040227

Mannaert, H., & Verelst, J. (2009). *Normalized systems: re-creating information technology based on laws for software evolvability*. Koppa.

Simon, H. A. (1981). *The Sciences of the Artificial* (2nd ed.). The MIT Press.

Van der Linden, D., De Bruyn, P., Mannaert, H., & Verelst, J. (2016). An explorative study of module coupling and hidden dependencies based on the normalized systems framework. *International Journal on Advances in Systems and Measurements*, *6*(1), 40–56.

KEY TERMS AND DEFINITIONS

Complex System: A complex system, whether it is a product design, organization structure or business process, consists of parts that interact and are interdependent to some degree (Sanchez & Mahoney, 2003, 2013).

Modularity or the Concept of Modularity: Is defined as a property of a complex system, whereby the system is decomposed into several subsystems (i.e., modules). Simon (1962, p. 474) explained modularity as "nearly decomposable systems, in which the interactions among the subsystems (modules) are weak, but not negligible".

Outsourcing: This study is using 'outsourcing or IS/IT project outsourcing' as a key term which includes all form of outsourcing arrangements. Among many services, IS/IT project outsourcing may include application development, application support, systems integration, data management, data center management, distributed computing services, and telecommunications-network management (Lacity, Yan & Khan, 2017).