Valuating Modular Architectures for Cross-Company Electronic Interaction

Beat Schmid, University of St. Gallen, St. Gallen, Switzerland, beat.schmid@unisg.ch
Christoph Schroth, University of St. Gallen, St. Gallen, Switzerland, christoph.schroth@unisg.ch
Markus Miche, SAP Research CEC Zürich, Switzerland, markus.miche@sap.com
Till Janner, University of St. Gallen, St. Gallen, Switzerland, till.janner@unisg.ch

Abstract

The persistent improvements in the cost-performance ratio of information technology today facilitate the provision and consumption of electronic business services across corporate boundaries. However, existing technical solutions and organizational approaches are not adequate to support the efficient management of loosely coupled, highly agile business networks which incorporate heterogeneous requirements. To address this challenge, we propose a modular architecture framework for cross-company electronic interaction which covers both organizational and technical aspects. We show its applicability in the field of Swiss public administration and investigate its economic potential based on a novel architecture valuation model which is particularly adequate to capture the value of agility.

Keywords: Modular architecture framework, option value, electronic markets, information technology valuation

1. Introduction

Cross-organizational electronic interoperation has become a key enabler for today's global service economy [10, 11, 12]. Service-oriented Architectures (SOAs) are acknowledged as promising architectural style for the design of such electronic business relationships.

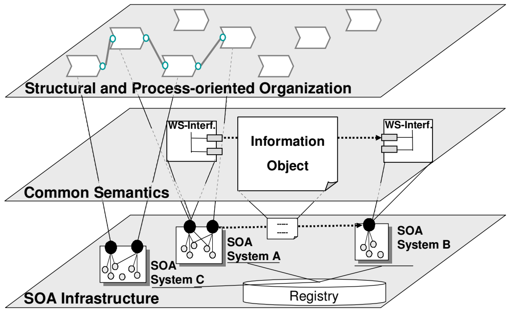


Fig 1. Basic Service-Oriented Architecture [14]

Figure 1 illustrates the fundamental concepts of SOA: On the lowest level, organizations expose dedicated services (black circles) and make them publicly retrievable via certain registry mechanisms. Such services can be composed of a number of other, only internally visible services (symbolized as small gray circles), complying with the principle of information hiding. On the basis of this infrastructural level, a common understanding of the semantics of exchanged messages needs to be ensured. The second level in Figure 1 illustrates the requirement of mutually comprehensible information objects. On the third level, services are orchestrated according to the previously specified structural as well as process-oriented organization.

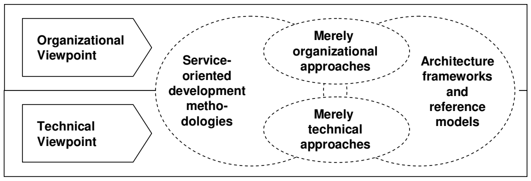


Fig 2. State-of-the-Art Analysis

While SOA represents a general architectural style, it does not provide concrete methodologies for the design and implementation of electronic business relationships that span across corporate boundaries. In fact, as depicted in Figure 2, four major clusters of methodologies have been identified that help enterprise architects to actually build a cross-organizational SOA:

First, **merely organizational approaches** allow for the modeling of interaction: While most of these strongly support the design of the process-oriented aspects of a given interaction scenario [7], they often do not provide integrated methods or modeling notations for the specification of the structural organization or the information objects exchanged in the course of interaction of companies. The lacking consideration of structural organization entails deficient organizational abstraction and thus flexibility. As also argued by other scholars, the principle of information hiding is required to allow for the encapsulation of company-internal design information and also for clearly specified interfaces between private and public (visible to other companies) views. The non-modular, comprehensive workflow models which are based on predefined process logic, offer little support for today's complex and dynamic business environments. Particularly in case of business networks which comprise knowledge-intensive tasks that are subject to strong variations, novel ways for reducing the complexity and increasing the agility are required. Also, all surveyed artifacts act on the assumption that the business processes governing the interaction of *one specific* business community need to be specified. They do not foresee possibilities to organize several communities in parallel of which each follows an individual organization but is still interoperable with the other communities.

Besides of these focused design methods, comprehensive **enterprise architecture frameworks and reference models** such as the Zachman Framework [20], the DoDAF, and the TEAF [16] are readily available. These represent essential means for managing intra-enterprise architectures as they structure architectures into domain-specific views to reduce inherent complexity. However, many of these can be considered system-centric since they mainly focus on aspects within the boundaries of an enterprise and thus do not necessarily optimize the design or governance of federated environments which need to accommodate heterogeneous requirements. Others already acknowledge the need for federated architectures but do not provide comprehensive methodological means for the decomposition of interaction scenarios and their subsequent assembly.

Similarly, **service-oriented development methodologies** (e.g., the IBM SOAD) already offer means for the design and implementation of a SOA, but follow a system-centric paradigm that does not support the design of highly heterogeneous, market-like IT service infrastructures in the inter-organizational realm.

Mainly **technical approaches** (see Figure 2) such as the Web services stack already represent a sound fundament for the implementation of inter-organizational relationships. Dozens of firms have emerged by today that rely on these technologies: Integration-as-a-Service (IaaS) providers [9, 14, 15], for example, offer reliable communication, partner management, technical integration services and application services. However, most technical approaches support the setup of hard-wired and stand-alone “island solutions” [9, 16] which are also often referred to as “B2B hubs”. Particularly the message formats used for the information exchange between agents represent a central issue: In fact, a plethora of different, mostly domain-specific standards for the modeling and physical representation of business information exist: RosettaNet Business Document (electronic components, and telecommunications industry), ACORD (insurance industry), CIDX (chemical industry), HL7 (healthcare industry), Papinet (paper and forest industry), PIDX (oil and gas industry), and SWIFT (financial industry) represent only a small selection of them.

Summing up, existing organizational as well as technical approaches, service-oriented development methodologies, and architecture frameworks are not sufficient to cope with the heterogeneity inherent in cross-organizational electronic interaction: Therefore, today, the frustration of organizations in establishing multiple, single-purpose partner communities grows as most existing solutions are not built to allow for extensibility and decentralized management [9].

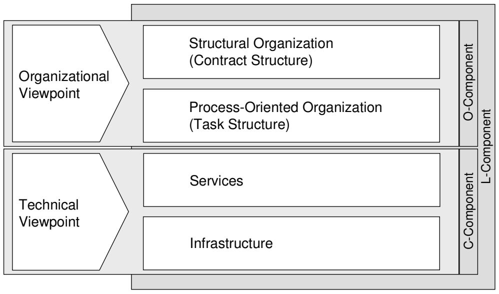
In this work, we propose a modular framework for distributed enterprise architectures [16] which support the organization and implementation of seamless electronic interoperation (Chapter 2). Leveraging the St. Gallen Media Reference Model (MRM) [12], the framework builds on the design principle of modularity [1, 3] to increase efficiency, flexibility, extensibility, to reduce design and management complexity, to account for uncertainty and finally to enable a decentralized and collaborative evolution of business media for electronic, cross-organizational interaction. We show the architecture framework’s applicability by presenting the electronic platform HERA which has been designed and built to support the cross-company scenario of corporate tax declaration processing in Switzerland. Chapter 3 elaborates on a novel architecture valuation method which goes beyond traditional approaches and is particularly adequate for capturing the value of agile IT service architectures in the cross-organizational realm. The method is used to investigate the economic value of the presented HERA platform. Chapter 4 closes the work with a brief summary and an outlook on future work.

2. Modular Cross-Organizational Interaction

2.1 Architecture Framework

For providing and consuming business services across corporate boundaries, media are required. According to Schmid [12], media are enablers of interaction, i.e. they allow for communicative exchange between agents which can be individuals, organizations or machines. In the course of interaction between agents, information objects can be created, modified, or exchanged via the medium. Media can be structured into three main components (see Figure 3): First, an organizational component (**O-Component**) defines a structural organization of agents, their roles, rules which impact the agents’ behaviour as well as the process-oriented organization of agents’ interactions. Second, a logical component (**L-Component**) comprises a common “language”, i.e. symbols used for the communication between agents and their semantics.

Fig 3. St. Gallen Media Reference Model [12]



Third, a physical component (**C-Component**) supports the technical interaction of physical agents. Based on these components, the MRM [12] structures the modelling of electronic media architectures into two major viewpoints: The organizational viewpoint allows for a technology-independent specification of the architectural artefacts constituting the O-Component as well as those parts of the L-Component which merely concern organizational aspects (e.g., the semantics of role descriptions). The technical viewpoint accounts for artefacts comprised by the C-Component and the parts of the L-Component which are related to technical aspects (such as the physical representation of information object semantics).

2.1.1 Organizational Component: Service-Oriented Modeling Based on Task Structure Matrices

As a first step in designing a modular architecture for electronic cross-organizational interoperation, the concerned interaction scenario is decomposed into its atomic building blocks which we refer to as tasks. Tasks are defined as (business) activities related to specific information objects. In the corporate tax declaration context which we focus on in Chapter 3, “send accounting documents to advisor for processing” represents an exemplary task. Its detailed specification encompasses one or a set of information objects, a general explanation of the operations associated with those objects as well as the agent role authorized to perform it. As a second step, the identified atomic tasks are assigned to both the x-, and the y-axis of a square task structure matrix in order to model the fundamental contract structure (structural organization) and task structure (process-oriented organization) of the scenario.

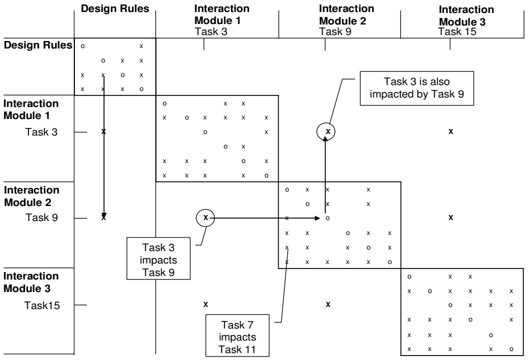


Fig 4. Identification and Full Decoupling of Interaction Modules (adapted from [3])

Task structure matrices have been introduced by Steward [19] in the product manufacturing context and have been further developed by Baldwin & Clark [3] for product design modularization purposes. We adapted the methodology for the purpose of modularizing “real-time” informational tasks which are conducted by agents in the course of cross-organizational electronic interaction [16]. Figure 4 visualizes an exemplary assignment of tasks to a task structure matrix:

All tasks which have been identified as constituent building blocks of a scenario are assigned to both the x-and the y-axis of the matrix (the exemplary tasks 3, 9, and 15 have been depicted in this figure). For each of the tasks, all “predecessor” tasks need to be determined in the next step since certain tasks require the input of another task before they can eventually be executed. In order to visualize such temporal interdependencies, the following rule is applied: In case task i precedes task j, a mark (x) is put in column i and row j of the matrix. In the scenario shown in Figure 4, for example, task 7 precedes task 11; therefore, a mark needs to be put in column 7 and row 11.

The setup of a comprehensive task structure matrix allows for the retrieval of interdependencies between single tasks and therefore can also be used for the conscious splitting apart of the overall scenario into autonomous, but interoperable **Interaction Modules** (**IAMs**). An IAM is defined as clearly delineated sphere comprising one medium as well as a set of agents connected to it. For the thorough identification and modeling of these IAMs, the following approach is suggested: A specific area within the matrix which features a high amount of marks indicates a group of highly interdependent tasks. These represent the constituents of an IAM. In the exemplary scenario depicted in Figure 4, three IAMs could be identified. For example, IAM2 encompasses tasks 7, 8, 9, 10, 11, and 12 which are subject to a number of mutual precedence relationships. Within the modules, high interdependencies exist between different tasks. Between the modules, on the other hand, as few as possible interdependencies shall exist in order to optimally decouple the interaction of agents. In case the structural or process-oriented organization of interaction within one of the modules needs to be changed, the impact is limited to this module and does not impact others. Each of the off-diagonal xs which are not included in one of the interaction modules basically represents an infringement of the principle of information hiding [3]. In the task structure matrix shown in Figure 4, for example, task 3 (part of IAM1) is required to be performed prior to task 9 (part of IAM2). This cross-module interdependency prevents from an entire decoupling and shall be removed through the definition and enforcement of one or several organizational **design rules** if possible: An organizational design rule can be considered as privileged design information which concerns aspects of structural organization (contract structure), process-oriented organization (task structure), and information objects.

Two different kinds of design rules exist: First, **prescriptive design rules** induce a change in the interplay of a set of IAMs, and aim to reduce mutual interdependencies. Second, **descriptive design rules** are used to make remaining interdependencies explicit to allow for the autonomous operation of the modules (which remain interoperable as long as the descriptive design rules are considered).

Figure 5 illustrates the comprehensive transformation from an interconnected design defining a single contract structure (the inner circle within the magnifier symbol) and task structure (the rectangle within the magnifier symbol) to a nested hierarchy of loosely coupled IAMs on the right side of the figure. First, the three IAMs identified in Figure 4 are now modeled as separate spheres of which each comprises one medium as well as a set of connected agents. In fact, these IAMs can now be considered as agent modules encapsulating several tasks and mutually hiding parts of their internal organization. IAM1, for example, is modeled as agent encapsulating a medium (M2) and three agents (A 1.1, A1.2, and A1.3).

IAM2 encompasses medium M3 and agent A1.1 (one agent may participate in more than one IAM). Finally, medium M4 and agents A3.1, A3.2, A3.3, and A3.4 constitute IAM3. By modeling those IAMs as higher-level agents which follow a dedicated contract and task structure (symbolized by the outer circle and rectangle on the right side of Figure 5), cross-IAM interdependencies can be removed. The organization of interaction (namely the contract and the task structures) between agents situated within one of these IAMs (e.g., agents A 1.1, A1.2, and A1.3 within IAM1) is symbolized by the circle and rectangle drawn around the respective medium (in this exemplary case, M2). It is hidden and thus decoupled from the internal contract and task structures established within IAM2 and IAM3. The black bars in Figure 5 represent the interfaces required between agents and media.

In case an agent follows design rules different from those established within the IAM it intends to participate in, adapters are required. In the scenario depicted in Figure 5, agent A1.3 in IAM 1 requires an adapter in order to connect to medium M2 (illustrated by the diamond-like symbol). It is important to emphasize the recursive character of the organizational modeling approach: Each agent performing one or several tasks may, if required, be further decomposed into an IAM, encapsulating another medium as well as connected agents. In the example shown in Figure 5, agent A1.3 is split into a medium M2.1, governing the interaction between agents A1.3.1, A1.3.2, and A1.3.3 according to a further, hidden contract and task structure (symbolized as circle and rectangle drawn around medium M2.1).

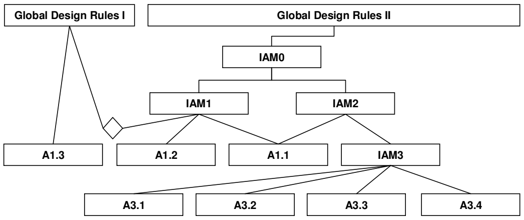
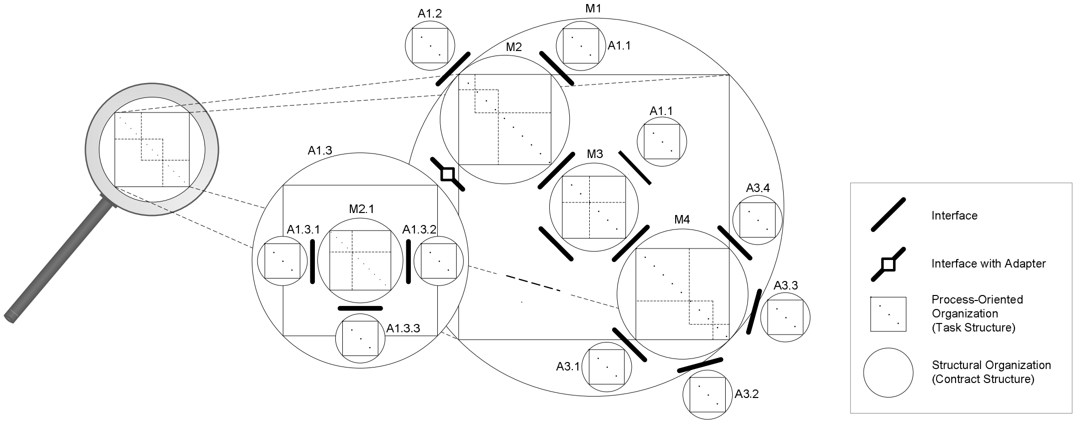


Fig 6. Design Hierarchy

Figure 6 shows the nested **design hierarchy** governing the relationships between the identified IAMs. On the topmost level, two sets of generic, global design rules (GDR) specify the minimal guidelines to be incorporated by their respective subordinate modules. IAM0 thereby represents a top-level module in this scenario; it determines design rules for decoupling the interaction of agents A1 and A2 (which encapsulate IAM1 and IAM2). In addition to the mere identification of IAMs (performed based on the TSM), design hierarchy diagrams allow to

Fig 5. Deriving the Organizational Viewpoint



unambiguously specify hierarchical relationships: In this scenario, for example, IAM3 shall be encapsulated as agent A3 that interacts within IAM 2 and thereby hides its internal design information.

As argued above, these agents encapsulate further multi-agent systems. In order to ensure interoperability between different IAMs, inheritance relationships are applied:

- Each IAM implements one specific set of design rules that govern and decouple the interaction of the agents interacting within the respective module.
- Each module (e.g., IAM1) can be considered as the ”heir” of its respective super-ordinate IAM (the module within which it is acting as agent; in this case, IAM0).
- Inheritance then implies the following: The design rules (specifying the contract and task structure as well as information objects) of IAM1 need to comply with the design rules of IAM0. In other words, IAM1 needs to implement all those tasks, consider their precedence relationships, and maintain all those information objects that are defined within IAM0 and are also related to agent A1 (encapsulating IAM1).
- IAM1 extends these inherited design rules by adding more tasks, information objects, and agent roles that are only relevant within IAM1 and are hidden from super-ordinate modules such as IAM0.
- Inheritance relationships are introduced to ensure a high degree of interoperability between different IAMs, while leaving each of them a considerable amount of design freedom.

2.1.2 Technical Component

In order to realize the “ideas“ modeled within the organizational component, the following approach is proposed (see Figure 7). For each of the identified IAMs, services are specified both for the comprised agents and the media in a recursive manner. Beginning with the “top-level” IAM0 (consisting of medium M1 and the two agents (A1 and A2) encapsulating IAM1 and IAM2 in the exemplary scenario), agent service interfaces are defined based on the information objects the respective agent is allowed to send or receive. As second major step, the services (and their respective interfaces) provided by the medium need to be specified. According to our architecture framework, three kinds of services need to be provided:

First, certain services need to exist which allow for the mere reliable and secure exchange of information objects between agents (such as services for routing, exception handling, and encryption). We refer to these services as **Operational Services** as they implement the C-Component of a medium. Second, services need to be provided which implement the above mentioned L-Component of the medium and ensure the common semantic understanding of all involved agents. We propose to implement one such **Semantic Service**, the Object Catalogue Service, which defines all information objects (by referencing XML Schema documents) which may be exchanged between agents within the sphere of one IAM. Third, **Coordination Services** implement the actual organization of interoperation (as opposed to the Operational Services which merely support reliable message transfer). Our architecture framework foresees two major services which together cover both the structural and the process-oriented organization. First, the Contract Structure Service incorporates the structural organization within this top-level IAM. Second, the Task Structure Service encompasses all the knowledge about the task structure inherent in the IAM. In specific, it knows which tasks (tasks are always defined with respect to one or several information objects defined as part of the Object Catalogue Service) can be performed within an IAM and also by which precedence relationships these are governed.

Figure 7 depicts the technical viewpoint that corresponds to the organizational model illustrated in Figure 5. The Operational, Semantic, and Coordination Services defined as part of the top-level medium M1 determine the structural and process-oriented organization governing the interaction between the three top-level agents A1 and A2 (which encapsulate IAM1 and IAM2) as well as the information objects exchanged between them. The internal organization of the two top-level agents may be fully hidden at this level of the design hierarchy. The modeling of the services constituting the lower-level, “sub-ordinate” IAMs (such as IAM1, IAM2, and IAM1.1) follows the same approach as shown above. Agent interfaces are modeled according to their respective roles and the tasks these roles are authorized to use within the sphere of this IAM. Medium-based services comprise Operational, Coordination, and Semantic Services, allowing for mere information object exchange, realizing the structural and process-oriented organization defined within their respective IAM and providing access to the definition of the information objects which may be exchanged via the respective medium. In case a lower-level IAM exactly follows the design rules implemented by its super-ordinate IAM, the “upwards-propagation” of information regarding task structure, object catalogues and agent directories is facilitated. We will elaborate on a proper example in the case study (Chapter 2.2). In case a lower-level IAM does not follow the design rules of its super-ordinate IAM, adapter modules are required.

2.1.3 Logical Component

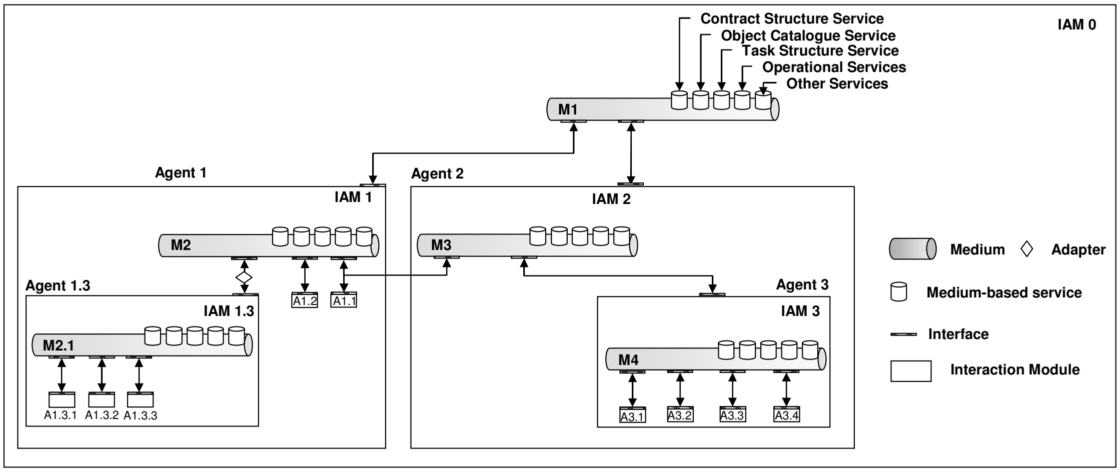


Fig 7. Technical Viewpoint of the Exemplary Scenario

The above discussed organizational as well as the physical components are of paramount importance for the design and implementation of cross-organizational electronic interaction. However, the mere reliance on an agile infrastructure and a modular concept for its organization alone are not sufficient for a comprehensive architecture framework: With the help of Web services, for example, particularities of heterogeneous applications can be encapsulated and made publicly available as uniform interfaces, thereby eliminating the need for huge application integration efforts. One challenge of cross-organizational interoperability that has still to be addressed, however, is that of semantic integration. Philip Merrick, former chairman and CEO of webMethods, has described this issue in the following way:"...there's a whole other layer to deal with, what I call the semantic integration problem. Web services are great but they standardize pure connectivity between applications. The applications still have highly varied data models, extremely different ideas of what business processes should look like". According to Roger Sippl, co-founder and chairman of Above All Software , the establishment of a common understanding of business semantics is one of the major "silo walls" that need to be taken down on the way to true enterprise application interoperability.

Today, a plethora of different, mostly domain-specific standards for the modelling and physical representation of business information exist: RosettaNet Business Document (electronic components, and telecommunications industry), ACORD (insurance industry), CIDX (chemical industry), HL7 (healthcare industry), Papinet (paper and forest industry), PIDX (oil and gas industry), and SWIFT (financial industry) represent only a small selection of these.

For this reason, our architecture framework foresees a modular, core-component-based modelling approach which relies on existing standards such as the OASIS Universal Business Language (UBL), the UN/CEFACT Core Component Technical Specification (CCTS), and, on a technical level, the W3C XML schema: Based on libraries comprising modular semantic building blocks, business documents can be flexibly assembled for the exchange between agents. The following paragraphs provide more detailed explanation of the most central concepts:

As illustrated in Figure 8, four abstract entities constitute the nucleus of the information object modelling approach: First, generic core components need to be defined which act as reusable, modular building blocks for the design of generic business documents. These generic (context-neutral) document descriptions encapsulate the organization of whole documents (e.g., order, invoice) shared by one or more instantiations which are referred to as specific business documents. Specific business documents, in turn, are constituted of specific core components, i.e. the context-specific instantiations of their generic counterparts, the generic core components.

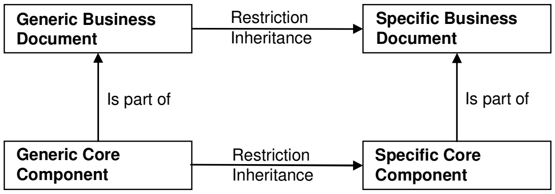


Fig 8. Information object modeling entities

The mechanism by which specific documents and core components are derived corresponds to the mechanism of "restriction inheritance". Only those information object constituents are selected that are

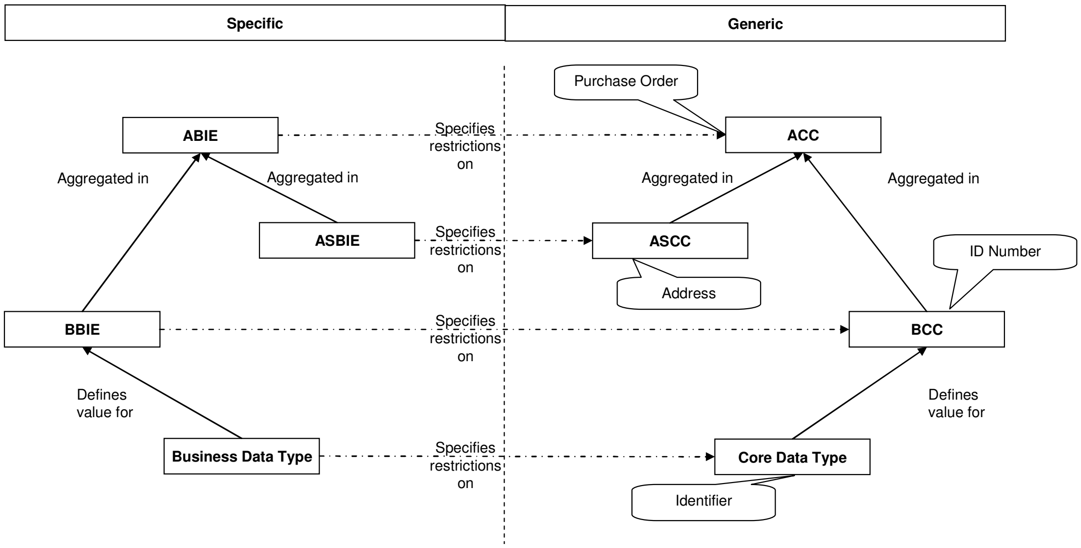


Fig 9. Generic and specific core components [14]

relevant in a given context. As mentioned above, the approach builds upon and extends the UN/CEFACT CCTS standard which foresees each information object to be composed of a set of building blocks, which are- in their most generic form- referred to as core components.

The following paragraphs elaborate on the detailed design of generic and specific business documents as well as their constituents (the core components): As illustrated on the right side of Figure 9, the most fundamental components are called Core Data Types (CDTs). The UN/CEFACT has established 21 of these CDTs (e.g., Amount, Binary Object, Code, Date, Date Time, Duration, Graphic, Identifier, etc.) which can be considered as atomic classifiers of pieces of information. The CDTs are used to define the allowed values for the Basic Core Components (BCCs), which represent basic information objects which cannot be split apart into different modules (e.g. ID Number or Street Name). Different BCCs can then be aggregated into an Aggregate Core Component (ACC), a super-ordinate module encapsulating its constituent core components. In order to allow for truly nested hierarchies of information modules, Association Core Components (ASCCs) exist which are composed of two ore more core components and can be included into a super-ordinate ACC themselves (Figure 9). A comprehensive Purchase Order Document can be modelled as ACC consisting of a number of ASCCs and BCCs. An address field may be organized as ASCC module as it comprises a set of subordinate modules such as Street Name, Postal Codes, etc. The Purchase Order ID Number, on the other hand, represents a BCC as it cannot be decomposed further.

All the generic core component modelling entities are context-sensitive and can be translated into specific Business Information Entities (BIEs) as illustrated in Figure 9: As part of every core component module, a set of context parameter categories is defined. The underlying methodology has been developed (and is still subject to amendments) as the Unified Context Methodology (UCM).

Parameter categories such as country or concerned industry as well as related values can be specified for each core component. Within the context-agnostic generic realm all core components (independent of their hierarchy level) comprise the full set of subordinate components. When “contextualizing” them, restrictions are imposed on each of them: The generic address core component, for example, features a wide set of subordinate components such as diverse kinds of street identifiers, postal codes, names and others. Depending on the country of a user, the component (which can be considered as generic template) can be restricted to those components which are specifically relevant in this country’s context.

Figure 10 illustrates the full “lifecycle” of our modelling approach. Based on a graphical modelling environment, users may document the constituents of the information objects they desire to exchange [18]. In a second step, these so far unstructured data models are transferred into formalized models which may draw on pre-existing core component libraries such as UBL. In case several context-specific instantiations of one document type exist, the specific, structured business document models are transferred into generic documents by building the superset and documenting the context annotations. These steps can be repeated by the stakeholders of a specific business community until a comprehensive library of information object models has been completed (see the middle of Figure 10). For the actual usage of the information objects, stakeholders may select generic documents from the library, apply the contextualization mechanism and translate the so far technology-agnostic models into a technical representation which can be used during run-time.

This component-based information object modelling approach [18] is new as it spans the bridge between unstructured modelling of data and core-component-based, formal representations and also because it integrates contextual information in order to allow for deriving tailor-made business information documents from generic information objects. The approach has been published in a number of conference proceedings and journal articles; it ranges from (tool-supported) graphical data models to the technical representation of the business documents such as XML schema documents designed in compliance with the UN/CEFACT XML schema Naming and Design Rules (NDR). The proposed approach incorporates the principle of modularity as it builds upon a number of semantic building blocks which can be assembled to whole business documents according to a clearly specified methodology. It facilitates cross-organizational interoperability (through homogeneously defined data components) while leaving significant design freedom to all involved stakeholders: Based on standardized document templates, context-specific derivatives can be created which fulfil heterogeneous requirements.

2.2 Case Study: Applying the Architecture Framework in the context of Swiss Public Administration (HERA)

In the course of the HERA project, the framework presented above has been applied to the context of corporate tax declaration in Switzerland. Mainly four different roles are involved in the scenario of collaborative tax declaration: Companies intend to finish and submit their tax declaration to the cantonal tax office. For this purpose, they may engage an external accountant, who often finishes accounting documents, creates the actual tax declaration on behalf of the firm and provides suggestions for optimal profit appropriation. By law, companies are enforced to interoperate with auditors who examine compliance of the declaration. Today, all organizations involved in this scenario interact via different media, often in paper format or based on proprietary electronic interfaces (some rely on stovepipe software solutions) and protocols. Besides technical challenges such as lacking interoperability, the collaboration between the stakeholders is managed in an informal fashion, differing from case to case. Depending on the cantons concerned (their respective legislation may vary), the structure of the business community (an external accountant may or may not be part of it), several context-dependent parameters, decisions made by agents internally or work results and individual preferences, the final flow of interaction strongly varies. Based on our architecture framework, a highly agile electronic platform (which we refer to as HERA platform) has been designed to support this cross-organizational tax declaration scenario [17].

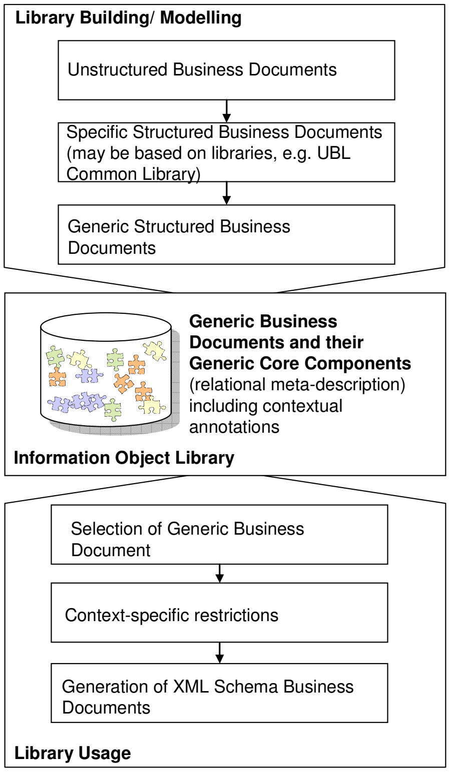


Fig 10. Library building and usage lifecycle

The application of the organizational modelling approach proposed by our architecture framework leads to the definition of atomic tasks, associated roles, information objects and a system of IAMs (see Figure 11). Thereby, IAM1 (Accounting/ Auditing) comprises all the tasks which are conducted by a taxable company, its accountant and an auditor prior to the submission of a tax declaration to the cantonal office. The second module (IAM2) comprises the tasks which constitute the exclusive interaction between the taxable company and the respective cantonal tax office. A third IAM has been defined concerning the mere interaction between the cantonal offices, which interact to determine the distribution of tax load on the different cantons the company has premises in. While high interdependencies exist between the tasks which constitute an IAM, as few interdependencies as possible shall exist between different modules. Through the establishment of organizational design rules, these have therefore been reduced significantly. Before the application of our architecture framework, for example, companies were required to directly interact with tax offices of all cantons they have at least one premise in. The diverse document formats as well as heterogeneous business processes induced huge administrative efforts for the companies and also required a significant number of point-to-point connections, preventing from agility. One of the organizational design rules now allows companies to only interact with their respective "main canton K0" (where the company's headquarters are located) and foresees it to deal with the other cantons in a way hidden from the company (see the top of Figure 11). After their identification and thorough modelling [16], the IAMs have been instantiated as follows: For each company, a separate module IAM1 "Accounting/Auditing" can be implemented. "Assessment/Enactment" IAMs (IAM2) are implemented once per canton, as the underlying structural organization, process-oriented organization, and the information objects to be exchanged between the agents depend on the respective cantonal legislation. Finally, the module accounting for government-internal interaction exists once.

The technical component of the HERA platform exactly corresponds to the design shown in Figure 7: Medium 1 (M1) supports the interaction between the high-level IAMs 1 (Accounting/Auditing), and 2 (Enactment/Assessment) as it governs the information objects exchanged between these as well as the roles they play and the process-oriented organization underlying the information exchange. The interaction between the agents constituting these IAMs is supported by separate media M2, and M3 of which each features Coordination Services, Semantic Services, and Operational Services as describe above. As the proposed architecture is fully recursive, individual agents (such as an accountant) can internally be organized as IAMs (encompassing a medium and diverse agents).

3. The Value of Modular Architectures for Cross-Organizational Interaction

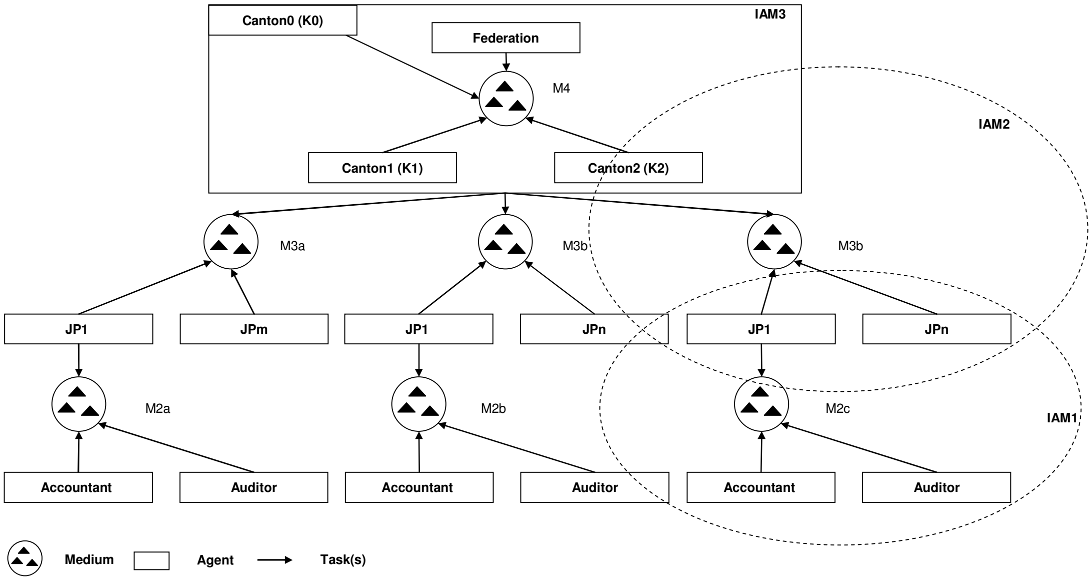
After the presentation of the fundamentals of our architecture framework as well discussing its first application in the context of the HERA project, a critical and systematic evaluation of benefits as well as potential risks is required.

3.1 Architecture Valuation Framework

To analyze and visualize utility and costs of software architectures several diverse procedures have been proposed in the past. In general, one can distinguish between checklist-, simulation- and scenario-based models as well as combinations of them. The approach illustrated in Figure 12 integrates the two scenario-based processes Architecture Tradeoff Analysis Method (ATAM) [5] and the Cost Benefit Analysis Method (CBAM) [8] which are both modified with respect to the specific requirements of architectures for electronic cross-company collaboration.

As a first step in this combined valuation method (see black-colored numbering in Figure 12), the architecture and related stakeholders have to be documented with respect to the investigated interaction scenario. Subsequently, a quality attribute tree shall be created in order to define the measures eventually used for quantifying the value of the architecture at hand. This tree starts with the general attribute "utility" as root node and divides into business drivers (high-level architectural goals)

Fig 11. HERA Architecture: Modular Organization of Interaction



which are identified during workshops conducted together with key stakeholders. For each of the business goals, concrete measures (the Quality Attributes) are specified in a hierarchical fashion. In order to concretize these attributes, scenarios must be described that specify a stimulus, a corresponding response and environmental conditions. The refinement of scenario descriptions represents an extension of the existing approaches: First, the frequency of occurrence is forecasted for each scenario. Moreover, the benefit is not assessed directly for each scenario, but broken down to the affected stakeholders and cumulated afterwards to get the total benefit of each scenario. This procedure enables a differentiated assessment of the benefit the architecture offers to each stakeholder. Expert estimates or comparisons between current and expected (after adoption of the architecture) costs can be used to quantify the benefit values. Finally, the costs accrued from the architecture's design and implementation must be valuated as well. Different effort estimations, like the Use-Case-Points method [6] or the Constructive Cost Model method (COCOMO), can be applied for this quantification. If possible, the architecture can also be decomposed into architecture strategies [5] which can be assigned to the scenarios concerned. This comparison of both the costs of the architecture on the one hand and the quantified benefits structured along key scenarios on the other hand form a basis for different qualitative and quantitative valuations and the calculation of performance indicators like the return on investment, the net present value or the break-even point for each architectural strategy as well as for the architecture on the whole.

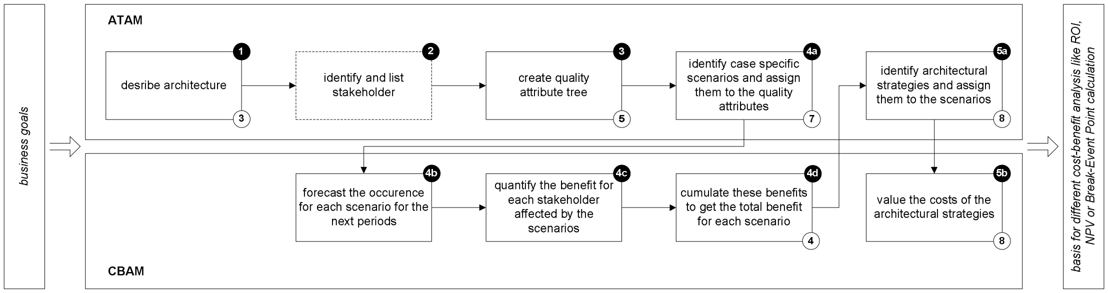
3.2 Valuating the HERA Architecture

During comprehensive workshops with representatives of all relevant stakeholders representing both providers and users of the HERA platform, three business drivers have been identified as particularly relevant: First, the modular architecture is supposed to improve the operational agility and thus reduce the costs of change. As quality attributes of this driver, we leverage the six "modular operators" proposed by Baldwin & Clark [3]: Augmenting (adding another module to an existing system), Excluding (the opposite of augmenting), Inverting (making so far hidden information visible to other modules), Substituting (exchanging one module against another), Splitting (breaking apart one module into several autonomous, but interoperable modules), and Porting (connecting one module following design rules A to a system which follows design rules B). We investigate the costs of performing each of these operators based on scenarios which have been defined in expert workshops and compare these with the respective situations before the implementation of the HERA platform. The second business driver interoperability shall be used to analyze the transaction costs reduction potential of the HERA platform. Transaction costs are measured based on the four quality attributes which have been identified in the course of the European ATHENA project [2] which will be explained below. The third and final business driver concerns data security: Similarly, the situation before and after the adoption of HERA will be investigated. The quality attributes mentioned in the paragraph above as well as the corresponding case-specific scenarios are presented in Table 1. Due to space constraints we reduce the proposed valuation method to a merely qualitative valuation by omitting the steps 4.b – 4.d.

Augment. The HERA platform allows for connecting a cantonal tax office to the "Governmental Interaction" module (IAM3) efficiently and without impacting other cantons: The existence of organizational and technical design rules ensures that the new agent can be added in order for it to be interoperable with all other cantons, thereby leveraging network effects for the whole system. Only one additional interface must be implemented between the agent and the medium rather than numerous, potentially proprietary interfaces to each of the other cantons as it is the case today. The existence of design rules for each of the IAMs allows for the efficient deployment of multiple instances of a specific IAM. The IAM1 module, for example, can be replicated numerous times based on the same set of general design rules. Software firms may offer the basic module for the interaction between a company, its accountant, and an auditor as a platform which can be adapted to individual requirements very quickly.

Exclude. The modular character of the HERA platform allows for offering a reduced set of modules

Fig 12. Process Model for Architecture Valuation



and extending these with the help of the augment-operator later. Rather than modeling all conceivable processes of interaction between companies, auditors, accountants and cantonal offices in the first place (involving a high risk of sunk costs as users might be reluctant to adopt the solution), the architecture allows for adding proper modules and medium-based services later.

Invert. In case a service developed and offered within an IAM constituting a cantonal tax office (for example, A3.1 in Figure 6) becomes of particular interest for other cantonal offices as well, the information necessary to access this service can be made publicly available to other cantons in their super-ordinate IAM (IAM3 in Figure 6) through propagating it upwards (in the design hierarchy) to the “Governmental Interaction” medium. This inversion implies a change in the design information that is visible to A3.1, A3.2, A3.3, and A3.4 who may then all use the service.Technically, this can be achieved through updating the Contract Structure Service, Task Structure Service, and Object Catalogue Service implemented as part of the medium in IAM3. In this way, cost savings through service reuse can be materialized.

Substitute. Due to the existence of clear design rules governing structural and process-oriented organization, exchanged information objects as well as infrastructural aspects within the “Accounting/Auditing” IAM, the accountant servicing a company can easily be substituted against another one.

Split. Due to the modularization of the process-oriented organization within all IAMs into atomic, loosely coupled tasks which are governed by precedence relationships, the holistic business process initially established within the “Accounting/Auditing” IAM can be broken apart into two modules: The first one only covers the interaction between company and accountant, the second one the interaction between company and auditor.

Port. The availability of open interfaces for adding new modules inherent in the HERA platform allows cantonal tax offices to seamlessly integrate legacy software applications such as their “Veranlagungssoftware”. The possibility to design adapter modules for porting those applications provides for huge cost savings potential.

Data Processing. During the interaction of companies, accountants and auditors, standardized information objects and automated processing of incoming and outgoing objects reduce costs which today incur from frequent transformations between different “island applications” and their respective standards. In particular, the incremental assembly of a company’s dossier not only across the boundaries of agents within IAM1, but also across the other IAMs until final enactment strongly diminish the efforts required for (manual) information processing.

Data Retrieval. Standardized message formats with well-defined, semantically meaningful meta information enable automatic filtering and extracting of tax declarations by cantonal offices.

Data Accuracy. The consistency-testing services implemented on the “Governmental Interaction” IAM allow cantonal offices to reduce opportunity costs by improving quality of information.

Maintenance. As opposed to a large, interconnected design, the modular HERA platform reduces complexity through information hiding and thus simplifies the error retrieval and removal process.

Data Security. The decomposition of the interaction scenario in organizationally and technically separated modules facilitates data security as information exchanged within one IAM can be hidden from other IAMs.

4. Conclusion

In this work, we analyzed managerial and technical weaknesses inherent in existing approaches to support the electronic interaction across corporate boundaries. To cope with these challenges, we

Table 1: Quality Attribute Tree for Investigating the Economic Potential of the HERA Platform

		Quality Attribute	Scenario
Utility	Agility	Augment	Integration of a new cantonal office to the “Governmental Interaction” IAM
		Exclude	Provision of a reduced set of modules while having the option to easily add others later
		Invert	Upwards propagation of services provided by cantonal offices for reuse purposes
		Substitute	A company changes its accountant
		Split	Splitting the ”Accounting/Auditing” IAM due to new legal specifications into two modules
		Port	Connection of legacy systems “Veranlagungssoftware” to the “Assessment/Enactment” IAM
	Interoperability	Data Processing	Creation/processing of a company’s dossier along a cross-medium information supply chain
		Data Retrieval	A cantonal office filters tax declarations for specific information artifacts
		Data Accuracy	A cantonal office accepts tax declarations without additional consistency verification
		Maintenance	Communication errors can be found and removed
	Data Security		Information exchanged within the ”Accounting/Auditing” IAM is not visible to other IAMs

presented a modular architecture framework and employed it to the scenario of collaborative tax declaration in Switzerland to illustrate its real-word applicability. In this way, we proved that both the physical medium's design as well as the organization of agent interaction could be truly modularized.

The economic potential of the HERA platform has then been investigated based on a novel architecture valuation method which optimally captures the value of modular designs. With the help of expert workshops conducted in the course of the HERA project, we identified interoperability, agility, and data security as the major business drivers underlying the analysis of architectural benefits. Based on fine-granular quality attributes and associated scenarios, the economic value of organizing and implementing electronic interaction based on our architecture framework could be shown.

One of the key insights gained is: The systematic splitting apart of cross-organizational interaction scenarios as well as their underlying information technology into modules with clearly defined interfaces allows for an unprecedented degree of agility. Modularity accommodates uncertainty and multiplies design options, thus creating a "portfolio of options" rather than an "option on a portfolio". [3] As opposed to the simple, Net Present Value (NPV)-based valuation techniques often used today, a model considering the availability of design options is needed to capture the benefit of modular architectures.

In modular designs, options can be leveraged through applying one or more of the modular operators discussed above. In the case of systems supporting cross-organizational electronic interaction between agents, for example, agent or media modules can be added, excluded, split, substituted, inverted and ported. These operators are applied entailing costs and benefits different from those in case of monolithic systems. Future publications will deal with leveraging the fundamental insights gained in the financial sector regarding option pricing for the development of quantitative theoretical valuation framework for modular IT service infrastructures.

5. Acknowledgements

This paper builds on the work conducted in different projects such as the EC-funded GENESIS (FP6) project, as well as the Swiss HERA project. HERA is funded by the CTI Swiss Confederation innovation promotion agency under grant number 8617.1.

The project comprises the following partners: University of Applied Sciences St. Gallen, Institute IPM; University St. Gallen, Institute MCM; University St. Gallen, Institute IDT; Abraxas Informatik AG; Abacus Research AG; BOC Information Systems GmbH; OBT AG; Finanzdirektion Kanton Schwyz; Finanzdirektion Kanton St. Gallen; Kantonale Steuerverwaltung Thurgau.

The first part of this paper draws on results stemming from the dissertation pursued by Christoph Schroth, while the valuation method has been conceptualized by Markus Miche in his diploma thesis.

6. References

[1] Armstrong, D. J. The Quarks of Object-Oriented Development, in: *Communications of the ACM*, 49 (2), ACM, New York 2006, pp. 123–128.

[2] ATHENA. Interoperability Impact Analysis Model, Deliverable B3.4, ATHENA project, visited 5.7.2007 under: http://www.athena-ip.org.

[3] Baldwin, C. Y., Clark, K. B. *Design Rules: The Power of Modularity Volume 1*, MIT Press, Cambridge, Massachusetts 1999.

[4] Bass, L., Clements, P., Kazman, R. *Software Architecture in Practice*, Addison-Wesley, Boston, Massachusetts 1998.

[5] Clements, P., Kazman, R., Klein, M. *Evaluating Software Architectures: Methods and Case Studies*, Addison-Wesley, Boston 2002.

[6] Frohnhoff, S., Jung, V.; Engels, G. Use Case Points in der industriellen Praxis, in: Applied Software Measurement - *Proceedings of the International Workshop on Software Metrics and DASMA Software Metrik Kongress*, Shaker Verlag 2006, pp. 511-526.

[7] Hammer, M. Reegineering Work: Don't Automate, Obliterate, in: *Harvard Business Review*, 68 (4), 1990, pp. 104-112.

[8] Kazman, R., Asundi, J., Klein, M. *Making Architecture Design Decisions: An Economic Approach*, Carnegie Mellon University 2002.

[9] Lheureux, B. J., Biscotti, F., Malinverno, P., White, A., Kenney, L. F. *Taxonomy and Definitions for the Multienterprise/B2B Infrastructure Market*, Gartner Research Paper, 2007.

[10] Malone, T. The Future of E-Business, in: *MIT Sloan Management Review*, 43 (1), 2001, p. 104.

[11] Mcafee, A. Will Web Services Really Transform Collaboration, in: *MIT Sloan Management Review*, 46 (2), 2005, pp. 78–84.

[12] Picot, A., Reichwald, R., Wigand, R. *Die Grenzenlose Unternehmung - Information,*

Organisation und Management: Lehrbuch zur Unternehmensführung im Informationszeitalter, Gabler, Wiesbaden 2003.

[13] Schmid, B. F., Lindemann, M. Elements of a Reference Model for Electronic Markets, in: *Thirty-First Annual Hawaii International Conference on System Sciences-Volume 4*, IEEE Computer Society, 1998.

[14] Schroth, C., Janner, T., Stuhec, G. UN/CEFACT Service-Oriented Architecture: Enabling Both Semantic And Application Interoperability, in: *Proceedings of the symposium "Communication in Distributed Systems" (KiVS), Workshop: Service-Oriented Architectures und Service-Oriented Computing*, VDE Verlag, 2007.

[15] Schroth, C. *Loosening the Hierarchy of Cross-Company Electronic Collaboration.* Springer Lecture Notes in Business Information Processing (LNBIP), 5, Springer, 2008, pp. 567-578.

[16] Schroth, C., Schmid, B. F. *A Modular Reference Architecture Framework for Electronic Cross-Organizational Interoperation*, in: Springer Lecture Notes in Computer Science (LNCS), Springer, 2008.- Seventh international EGOV conference 2008 (EGOV 2008), Turin, p. 12.

[17] Schroth, C. A Service-oriented Reference Architecture for Organizing Cross-Company Collaboration, in: *Enterprise Interoperability III: New Challenges and Industrial Approaches* (K. Mertins, R. Ruggaber, K. Popplewell, X. Xu, Editors), Springer, Berlin 2008.

[18] Schroth, C., Pemptroad, G., Janner, T. CCTS-based Business Information Modelling for Increasing Cross-Organizational Interoperability. *Interoperability II. New Challenges and Approaches* (R. J. Gonçalves, J. Müller, K. Mertins, M. Zelm, Editors), Springer, Berlin 2007.

[19] Steward, D.V. The Design Structure System: A Method for Managing the Design of Complex Systems, in: *IEEE Transactions in Engineering Management*, 28 (3), 1981, pp. 71-84.

[20] Zachman, J. A. A Framework for Information Systems Architectures, in: *IBM Systems Journal*, 26 (3), 1987.