Bus Inf Syst Eng 57(1):37–49 (2015)
DOI 10.1007/s12599-014-0364-y

RESEARCH PAPER

Designing the Shift from Information Systems to Information Services Systems

Jolita Ralyté · Abdelaziz Khadraoui · Michel Léonard

Received: 8 March 2014 / Accepted: 22 October 2014 / Published online: 3 February 2015

Abstract Service-oriented paradigm is emerging as a promising solution to handle enterprise information systems (IS) complexity, interoperability, and evolution issues. However, the shift from a conventional IS architecture to a service-oriented one is not an easy task. Various approaches have been proposed in the literature and most of them deal with either business strategic service design or a technical level IS transformation and migration. There is a need for approaches forming the link between these two levels and designing the information level transformations. In the paper, the concepts of information service and information services system (ISS) are promoted and three different ways to design information services and ISS are proposed taking into account enterprise legacy IS evolution. The three approaches are illustrated with examples taken from industrial projects and case studies.

Keywords Information service · Information services system · Service-oriented paradigm · Legacy information systems · Information overlap

Accepted after three revisions by the editors of the special focus.

J. Ralyté (✉) · A. Khadraoui · Prof. M. Léonard
Institute of Services Science, CUI, University of Geneva,
Battelle Bâtiment A, 7 Route de Drize, 1227 Carouge,
Switzerland
e-mail: Jolita.Ralyte@unige.ch

A. Khadraoui
e-mail: Abdelaziz.Khadraoui@unige.ch

Prof. M. Léonard
e-mail: Michel.Leonard@unige.ch

1 Introduction

Introducing the service-oriented paradigm into enterprise information systems (IS) engineering promises a gain in IS sustainability, a support for legacy IS evolution, and an advance in agility of new IS developments. Knowing that today many enterprises depend on their legacy IS, which support their critical business activities, the evolution of these IS becomes vital for ensuring their competitiveness in the market or even for their survival. Considering the fact that enterprises constantly undergo business, structural and technological changes, replacing enterprise legacy IS by new and modern ones is not always a possibility. Legacy IS has to evolve together with enterprise changes, and this evolution can take different forms, as for example: the integration of new IS components acquired on the market or custom developed, the reuse of data and applications of legacy systems in order to provide new services, or the establishment of interoperability between two IS.

Service-oriented paradigm is founded on principles, and in particular those of modularity, reusability, and evolution, that aim to support IS adaptability in their ever changing environment and to ensure their compliance with the enterprise business and governance activities. Well defined modularity is necessary to avoid chaotic IS fragmentation and to ensure incremental and evolutionary IS development. To achieve it, an IS has to be designed as a collection of interrelated and autonomous components, and the notion of service seems to best serve this purpose. Reusability means the exploitation of legacy data and applications to provide new services while the evolution principle consists in the ability to easily replace an existing service by a new one.

In our previous works, we have introduced the notions of information service (Arni-Bloch and Ralyté 2008; Arni-

Bloch et al. 2009) and information services system (ISS) (Ralyté 2012; Ralyté et al. 2013) that we consider a natural evolution of a conventional IS concept into a service-oriented one. Similarly to IS, ISS put emphasis on the value of information, its creation, management and sharing. The difference between IS and ISS consists in the way the information is structured – ISS improves its modularity, agility, and interoperability, notably because it is based on the concept of semantic unit (Turki and Léonard 2002b). Introducing the notion of ISS helps us to handle the evolution of legacy IS as well as to design new evolutionary systems. In particular, we have developed three different but complementary ways to design ISS which we present in this paper. The three approaches, introduced in Ralyté et al. (2013), are the outcome of several internal projects and a long-term collaboration with the Information Technology Center of the State of Geneva (Switzerland). In this paper we aim to provide a detailed overview of our work in the domain of ISS engineering taking into account legacy IS evolution, to situate each approach in different ISS engineering contexts and to discuss their complementarity.

The rest of the paper is organized as follows: in the next section we present the related work and how our contribution complements it. In Sect. 3, we define the notions of information service and ISS that represent the foundation of our work. Then, in Sect. 4 we present and illustrate three approaches supporting ISSs design, and we discuss the situationality and complementarity of these approaches. Finally, we conclude our presentation and discuss future perspectives of this work.

2 Related Work

2.1 The Concept of Service

The concept of service has multiple facets and various definitions. Quartel et al. (2007) classify them into six categories and provide an overview of various service usages. This overview demonstrates that the notion of service is equally used in business and in software and information technology domains and its granularity varies from a simple action on an object to an application or a software component (web service). Here we only pay attention to the concept of web service (Bichler and Lin 2006) that enables the creation of new business models – e-business – with the help of service-oriented architectures (SOA) (Erl 2007; Krafzig et al. 2004; MacKenzie et al. 2006; Papazoglou 2008; Booth et al. 2004). E-business and value modeling approaches are proposed to design new e-service-based collaborations and service systems (Baida et al. 2004; Gordijn et al. 2009; Regev et al. 2011).

The concept of information service, as we define it in this paper (Sect. 3), is particularly adapted for designing service-oriented IS which we call ISSs. Service granularity level, focus, and scope are the main characteristics that make our definition of information service different from the aforementioned service definitions. The granularity level of information services is quite large – a service represents a full business unit, not just a simple action or application. Besides, an information service can not only support inter-organizational collaboration and value exchange between organizations and their clients but also, and in particular, intra-organizational activities.

2.2 Adoption of Service-Oriented Paradigm in IS Engineering

Today, service orientation is considered as a new design paradigm for increasingly complex IS engineering which promises to improve their flexibility and changeability. However, adoption of SOA approaches turns out to be a real challenge for IS engineers and managers. A few publications discuss how to assess legacy IS for evolution towards SOA (Reddy et al. 2009; Ransom et al. 1998) and analyze the impact of SOA on enterprise systems (Bieberstein et al. 2005). Other research works define critical success factors of service orientation in IS engineering (Aier et al. 2011), discuss strategies for service-oriented IS design (Aier 2012) and how service-oriented design should be applied in an organization in order to adopt SOA (Chua 2009), and analyze SOA application in practice (Legner and Heutschi 2007).

Most of the current SOA approaches (Erl 2007; Krafzig et al. 2004; MacKenzie et al. 2006) do not pay much attention to the legacy IS reuse and evolution, which is the main preoccupation of our work. Moreover, the compliance of services with the laws and enterprise regulation policies is not considered in these approaches, while it is an inherent part of our contribution. In the context of IS engineering, conventional SOA approaches can only recommend to rebuild enterprise IS in terms of autonomous services that could be composed in different ways. Such services have to be elaborated from scratch in order to avoid any overlap between them. This type of development represents a rather extreme solution and is not adapted to the legacy IS evolution. The lifecycle of an IS is a continuous incremental and evolutionary process and it is not possible to rethink the entire IS at each iteration in order to guarantee the autonomy and correctness of the existing and new services.

2.3 Service-Oriented Approaches for IS Engineering

The number of proposals to define SOA for IS is growing. However, most of these approaches consider only technical integration or migration of legacy IS to the service-oriented technology (Channabasavaiah et al. 2003; Umar and Zordan 2009; Cuadrado et al. 2008; Khadka et al. 2011; Almonaies et al. 2010), and propose to reuse legacy code to provide new services, generally web services (Sneed 2006; Liu et al. 2008).

Other approaches define SOA for IS at the conceptual level. We can mention the conceptual framework for designing service-oriented inter-organizational IS (Le Dinh and Nguyen-Ngoc 2010) and the conceptual framework for service modeling in a network of service systems (Le Dinh and Pham-Thi 2010). A model-driven approach for service-oriented IS development is introduced in El Fazziki et al. (2012), but mainly focuses on mapping rules from BPMN models to SOAML diagrams. Thomas and vom Brocke (2010) present a value-driven approach to design service-oriented IS; the approach is based on business process modeling and cost/benefit analysis to determine whether the introduction of SOA justifies the effort. Lo and Yu (2007) propose a reference catalogue approach to design an SOA system. This approach uses the i* modeling technique adapted to service-oriented business modeling for the selection of reference business models from the catalogue and their adaptation to the particular case. Estrada et al. (2010) introduce a service-oriented organizational model in order to reduce the mismatch between business models and service-oriented designs.

Most of the aforementioned approaches propose SOA for IS either at business strategic or at a low technical level, without considering the link between them. Besides, most of them only consider the customer–provider perspective – how the organization could offer services to its customers. The intra-organizational perspective of a service as a support for the internal enterprise activities and information exchange are not really considered.

Unlike the aforementioned contributions, our three approaches for ISSs engineering (presented in Sect. 4) aim to establish a concordance between the business and technical implementation levels throughout the conceptual design of the information management level. Besides, they support service-oriented inter- and intra-organizational ISS development taking into consideration legacy IS and their evolution.

3 Information Service and Services System

3.1 Information Service

In the domain of enterprise IS, the notion of service was introduced as a potential means to improve legacy IS agility and evolution and to facilitate IS interoperability. It is built upon the concept of IS component (Turki and Léonard 2002b), which is defined over classes assembled into hyperclasses, methods, integrity rules (IR), processes, roles, and events that constitute a semantic unit where several actors aim to achieve a common goal. In order to fit the IS context, a service is expected to support inter-organizational and/or intra-organizational business activities through a collaborative creation and transformation of shared information. This type of service is named an information service (Arni-Bloch and Ralyté 2008; Arni-Bloch et al. 2009). An information service is defined as "a component of an information system representing a well defined business unit that offers capabilities to realize business activities and owns resources (data, rules, roles) to realize these capabilities". Therefore, an IS can be seen as built of a collection of interoperable information services. In this section, we demonstrate how we derive the metamodel of information service from the IS metamodel.

3.2 Information System Metamodel

As a prerequisite for defining the notion of information service, the IS metamodel must take into account the necessary transparency of information services, notably their information resources and their capabilities in the organizational context. In particular, the IS metamodel must distinguish four interrelated information facets which we call information spaces – static, dynamic, rule, and role. Figure 1 shows such an IS metamodel in its simplified version – including only the main concepts that are necessary for the subject of this paper.

The IS static space defines its data structure in terms of classes and relationships between classes. Classes are linked only via existential dependencies and specialization relationships. A class c_1 is existentially dependent on the class c_2 if any object of c_1 is permanently associated with one and only one object of c_2. Such an existential dependency is materialized by a directed edge from the node of c_1 to the node of c_2. If c_1 is a specialization of c_2, then c_1 must firstly be existentially dependent on c_2.

The dynamic space defines methods specified over the classes, and actions that can be executed in the IS by means of methods, which produce effects on IS classes: access, update, creation, and deletion of objects. The context of an action is the set of such classes. An action is described by a process to be executed, which can be a simple function or a more complex interaction involving several actions. It is triggered by one or several events.

The rule space deals with regulation policies which are formalized as IR defined over classes that constitute the IR context. A risk of an integrity rule refers to a class atomic operation – create, delete, or update – which could transgress it if no validation is executed. Consequently, every

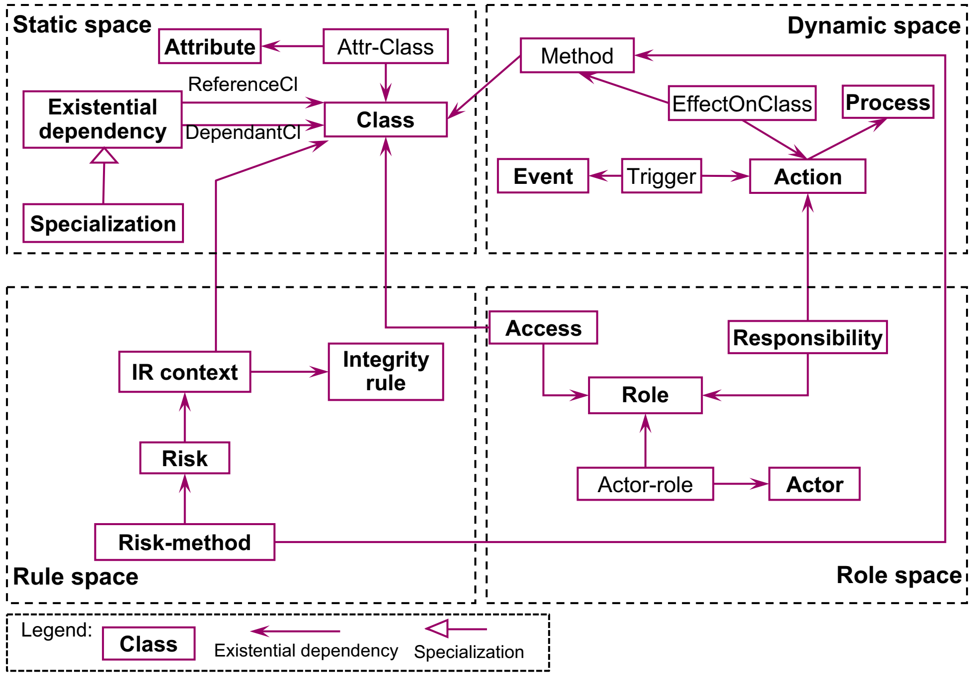

Fig. 1 Simplified information system metamodel

method using such atomic operations must cover all their risks (class Risk-method in Fig. 1).

Finally, in the role space, actors can take/be assigned to roles to obtain access to the objects of certain classes and/or to take responsibility for certain actions. An actor can take several roles and can share the same role with other actors. Of course, for the purpose of transparency, any role taking responsibility of an action should have access to all classes of the context of this action. Furthermore, two roles can have an overlap composed of classes and actions they share.

3.3 Information Service Metamodel

We claim that in the IS context it is not sufficient to consider services as black boxes with only an interface part available for their selection and composition purposes. It is essential to make explicit the information concerning service structure, processes, rules, and roles, and to be able to identify those shared with other services. For us, information services have the same conceptual basis as IS. Indeed, an information service, by itself, is an information system. However, the complexity comes from the fact that an information service is generally related with one or several IS, and, furthermore, several information services can have overlaps between them. In particular, they can share the same classes and actions. Thus, the metamodel of information service inside an IS can be obtained from the IS metamodel shown in Fig. 1. It is depicted in Fig. 2 in a simplified way. The grey elements in this metamodel correspond to the concepts proper to services.

The static space of a service is defined over one hyperclass (Turki and Léonard 2002a). A hyperclass is defined upon a set of classes and is used to represent a complex concept by joining together all the classes of the set. Then this complex concept provides a semantic unity for the service. The classes of the hyperclass can be proper to the service; but, in most cases, they are also classes of the IS.

The dynamic space of a service, as for an IS, determines the set of actions that can be executed by the service. Of course, the contexts of these actions must be included in the hyperclass of the service for consistency reasons. These actions can be specific to the service or be defined at the IS level. In the latter case, the service uses actions which are also used by other processes in the IS.

The rule space of a service, as for an IS, deals with regulation policies that are formalized as IR defined over classes. Every integrity rule is associated with a context and a set of risks. Besides, there are additional complexities due to the interrelations between the information service, a kind of local IS, and the IS at the global level. We illustrate such complexities by means of generic situations. Let's look at an example where an action of a service validates all the rules defined over this service – the local IR. However, without any other information, this local validation does not guarantee the global IR validation. For instance, this action can create an object of the class c_1 without validating a possible global integrity rule defined over the classes c_1 and c_2, because c_2 does not belong to the hyperclass of the service, and thus cannot be validated locally.

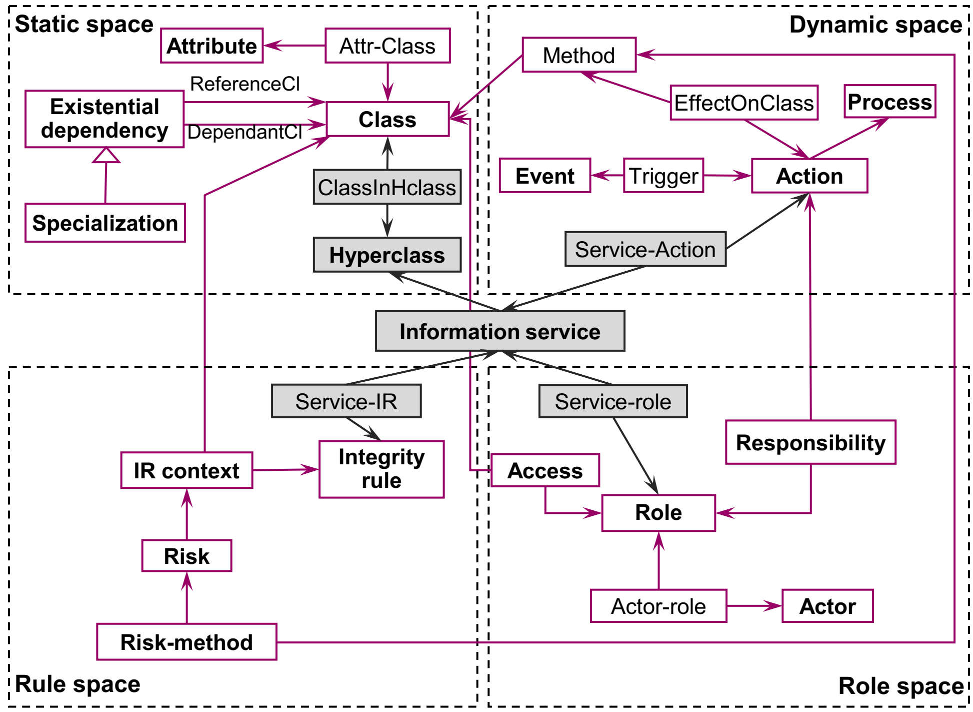

Fig. 2 Simplified metamodel of an information service inside an IS

The reverse situation is also embarrassing: an action executed at the global IS level on a class c_1 (e.g., creates an object of the class c_1) would validate all global rules defined over this class. However, because c_1 belongs to some particular service where potentially some local rule defined over c_1 exists, obviously this local rule would not be validated at the global level, where it is unknown.

Finally, in the role space of a service an actor who takes a role associated with a service may then access all the classes of the service and take responsibility for all the actions of the service. Generally, the design of the role space of a service is more sophisticated, but the complexities are similar as in the case of IS. However, an additional level of complexity is induced by the interwoven situations of roles in IS and roles in services. We will illustrate this with the following two examples.

A role can be associated with a service, and therefore the actors taking this role can access the class c_1 of this service. However, at the IS level, it is possible that the same role does not have access to this class. How to face such a generic situation?

Another generic situation concerns the overlap between roles of services s_1 and s_2. Let's say these two services have an overlap composed of the class c_{12}. Therefore, it is possible that two actors a_1 and a_2 can work with the roles r_1 and r_2 by means of the services s_1 and s_2 respectively on the class c_{12} while ignoring each other. Is this situation correct? Maybe yes, maybe not. However, it is clear that in this situation a protocol of the interactions between the actors a_1 and a_2 upon the class c_{12} needs to be designed. In a special situation, the protocol could say "do nothing"!

3.4 Information Services System

According to (Spohrer et al. 2007, 2008) a service system is a configuration of people, technology, shared information (such as language, processes, metrics, prices, policies, and laws), and other resources that interact with other service systems to create mutual value. Spohrer et al. (2007, p. 72) also say, "service systems comprise service providers and service clients working together to coproduce value in complex value chains or networks". In their vision, there is a clear separation between the service provider and its client. However, this distinction is not so obvious from the IS point of view. An actor of an IS can be allowed to provide and to consume service information and capabilities. In this case, the term "service prosumer" matches the situation better. But the notion of value coproduction is also a key one in the domain of IS.

In Spohrer et al. (2007) the authors claim that the main difference between service systems and computational systems is people – unlike computational system components, the behavior of people doing work in service systems cannot be easily modeled and simulated which can create risk but also generate innovations. We agree with these authors and we take into account this issue. We argue that in ISSs people are not considered as independent system components but rather as actors enabling capabilities of information services. The scope of their behavior inside an ISS is explicitly described by means of their organizational roles and their responsibilities involved in the information services enactment (provision and/or consumption of information resources) and is implemented by a set of rules embedded in the services. We also think that the unpredicted behavior of actors could be explored as a potential source of ideas for future business innovations. It could be captured in the form of initiatives. As service systems, ISSs aim to stay dynamic and open in order to enable innovation and facilitate their evolution. They have to take the risk to give people some liberty to informally and formally change rules and policies.

Therefore, in our approach an ISS is seen as a collection of interoperable information services. The notion of ISS aims to transform an integrated and rather rigid IS architecture into a more flexible, modular and sustainable one where services can be modified or replaced and new services can be integrated.

The metamodel presented in Fig. 2 defines the foundation for engineering information services and services systems. In particular, it supports information services definition, composition, identification of the overlap between information services, and also new services integration into an ISS.

4 Designing Information Services Systems

Undertaking the shift from conventional IS to the service-oriented ones is not an obvious task, especially when various legacy IS are at stake. Such transformation and transition needs to be carefully designed and governed. In this section we present three approaches to guide the design of ISSs while taking into consideration the evolution of enterprise legacy IS.

4.1 Information Services upon Legacy IS

Our first approach, introduced in Khadraoui et al. (2011), guides the definition of new information services upon various legacy IS. It aims to reuse resources of existing IS (their data, processes, rules, responsibilities) for providing new services. The main challenge of this approach is to ensure that the consistency of legacy IS will not be jeopardized while executing the new services.

The approach consists in identifying for each new service the existing resources that are potentially scattered in different IS and guaranteeing that the execution of the service will keep these legacy IS in a consistent state, i.e., will ensure data consistency and will not violate their rules and responsibilities. The key step of this approach implicates the construction of a common base on top of a set of existing IS. The role of the common base for each service consists in:

- Specifying the overlapping information available in different IS,
- Enabling access to the precise and consistent information distributed in various IS, and
- Guaranteeing the service compliance with the existing IS and with the enterprise's legal frame, which is a composition of laws and regulation policies that govern enterprise activities.

Therefore, this approach is largely based on the analysis of:

- The organizational context that defines the scope of service capabilities,
- The organizational roles responsible for the execution of service actions, and
- The legacy IS that contain information necessary for service execution.

The metamodel shown in Fig. 3 is defined to support this analysis. It specifies the key concepts necessary to link new services to the existing IS that contain resources necessary for their execution, and to lead the construction of a common base on top of several legacy IS. The definitions of the concepts of service and information system are compliant with the ones given in Sect. 3.

A service can be defined for one or several organizational contexts (OC) that generally describe business rules, legal constraints, and the capability of the organization to enforce laws and policies. The notion of hyperconcept (HC) is used to specify how the OC are formalized in different IS. A HC is a complex concept (composed of several sub-concepts) representing a semantic unity in the domain of analysis. The fact that an organizational context can contribute to the definition of several hyperconcepts is formalized in the class HC-OC. A HC contains concepts from one or several legacy IS, which is formalized in the class IS-HC. The class Service-OC-HC allows to establish the link between a service, its OC and the hyperconcepts formalizing them, while the class IS-HC-OC connects the OC, hyperconcepts and related IS. In addition, the role concept represents the responsibilities and authorizations to execute IS activities in a particular organizational context. The class IS-HC-Role-OC expresses the direct link between the IS, hyperconcepts, roles, and OC.

4.1.1 Process Model for Defining Information Services upon Legacy IS

The process model of this approach is formalized in four steps:

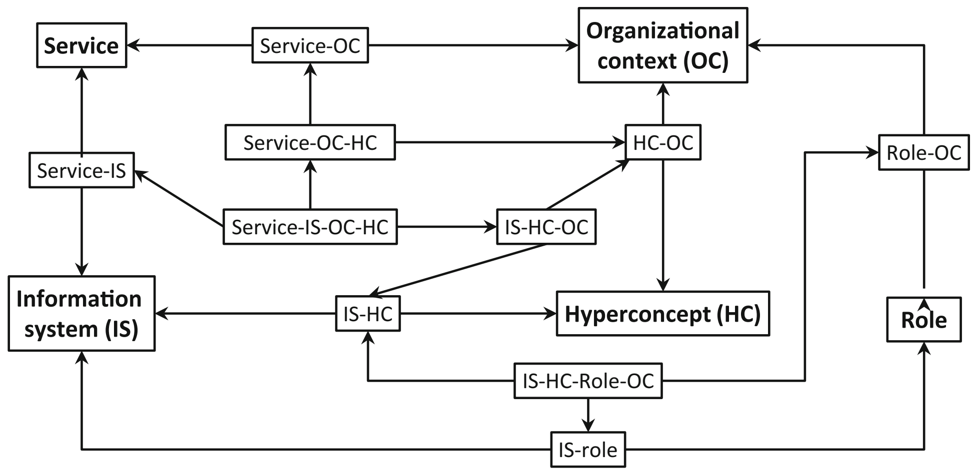

Fig. 3 A metamodel for defining information services upon legacy information systems (adapted from Khadraoui et al. 2011)

Step 1: Construct a Frame of Reference The first step consists in constructing a frame of reference – the conceptual model that includes the definition of service OC as well as service roles and identifies the legacy IS that contribute to the service realization, by instantiating the metamodel shown in Fig. 3.

Step 2: Select Candidate Information Elements During the second step, the frame of reference serves as a basis for the selection of informational elements (hyperconcepts and roles) that are candidates for the common base. For example, the objects of the class Service-OC specify which OC have to be taken into consideration as a legal basis for the common base. A typical candidate for the common base would be a HC implemented in more than one IS (class IS-HC-OC) and representing a part of the service organizational context. The same applies to the roles related to the service OC and defined in different IS (class IS-HC-Role-OC).

Step 3: *Construct a Common Base* Construction of the common base from the selected informational elements can be achieved in a collaborative decision-making way where project stakeholders (business practitioners, IS architects, database architects, etc.) are invited to assess different formalized alternatives and to choose the elements (e.g., service related roles) to be implemented in the common base.

Step 4: Adapt the Existing IS Finally, the last step is dedicated to the adjustment of the related legacy IS in order to guarantee that their integrity is still guaranteed and they are interoperable with the new common base. If necessary, the legacy IS can be adapted by adding new elements (e.g., adding new role) or transforming the existing ones (e.g., changing a business rule). The number and complexity of the identified transformations indicate the weight of the impact of the new service creation on the legacy IS, and this impact should be minor.

4.1.2 Illustrating Example

The approach Information Services upon Legacy IS was applied in collaboration with the Information Technology Center of the State of Geneva (Switzerland) to design new services for e-administration based on seven existing independent IS (see Fig. 5). Four of these IS operate at the cantonal level: (1) the Commercial Register (RC) that allows to build and identify all legal entities in the State of Geneva and to register their associated legal events, (2) the Tax IS (R-Fisc) that stores the taxation data about businesses at the cantonal level, (3) the Geneva Business Directory (REG) that contains administrative information on companies located in the canton of Geneva and makes this information available for administrative purposes and for the applicable dissemination in public and private sectors, and (4) the Cantonal Office of Statistics (OCSTAT). These cantonal IS have to interact with three similar IS in use at the federal level: (5) the Federal Commercial Register (RCF), (6) the Federal Business Directory (REE) and (7) the Federal Office of Statistics (OFS).

Several information services have been identified for this project; Fig. 4 illustrates three of them. The first service concerns the "Transmission of business statistical data". It aims to help companies transmit their data to the cantonal and federal offices in order to build business statistics. Another service deals with the "Processing of business statistical data" and helps both cantonal and federal offices of statistics to process businesses data and produce reports. Finally, the third service concerns the "Use of business statistical data" by different organizations in adequacy with the legal framework. The organizational context of these services is based on two laws on public statistics: (LStat)[1] at the cantonal level and (LSF)[2] at the

[1] http://www.geneve.ch/legislation/rsg/f/s/rsg_B4_40.html.

[2] http://www.admin.ch/ch/f/rs/431_01/index.html.

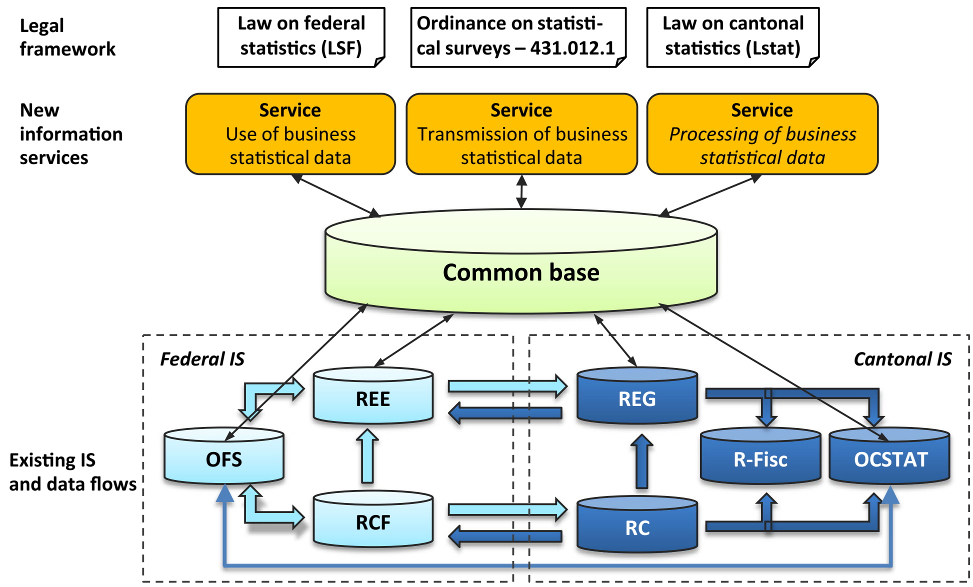

Fig. 4 Example of service organizational context analysis

federal level, and on the Ordinance on statistical surveys.[3] The data related to this context is available in four IS: REG, REE, OCSTAT and OFS. The construction of the common base consisted in the selection of elements (data, roles, rules) that overlap in the related IS and are necessary to realize the aforementioned services. The main complications encountered in this project were related to the common base management. In particular, it was difficult to decide who will be responsible for this common base and how to manage the existing flows between the existing IS (REG REE, OCSTAT, OFS).

4.2 Fully Service-Oriented ISS

Our second approach, named Fully Service-Oriented ISS, deals with the extension of an existing IS, already designed as a collection of services (as defined in Sect. 3 and formalized in Fig. 2), with new information services.

In Arni-Bloch et al. (2009), Ralyté et al. (2010) we have proposed a service-oriented and situation-driven approach for legacy IS evolution by integrating new services. Here, we argue that this approach can also be used for ISS engineering in an incremental way by progressively integrating new information services. The main difficulty of this approach lies in the fact that services composing an ISS are not totally independent components – they inevitably overlap, which is mainly because of the data sharing. In fact, the overlap between information services can appear in the four information spaces (static, dynamic, rule and role), and each new service integration can create new overlap situations. Therefore, this approach is based on the analysis and resolution of overlap inconsistencies between legacy and new services by preserving the legacy ones as much as possible.

4.2.1 *Illustrating Example*

In order to validate this approach, we have conducted a case study that concerns the extension of a University Students Management System with new information services. In particular, we have considered the diploma management service (DMS) as an existing information service and the online registration service (ORS) as a new service to be integrated in the existing ISS. The DMS provides several diploma degree management capabilities such as: to create the curriculum of each diploma by defining its courses and linking them to their lecturers, to manage students' registration to different diploma degrees and to the corresponding courses, to manage examination results, etc. The new service, the ORS, enables students' registration by providing a web interface for this purpose. A candidate can create a university registration request and include different required documents by using this interface. Then, the students' administrator validates the on-line created registration requests. He/she is responsible to record the corresponding person as a student and to register him/her for the selected diploma. A detailed description of this case study can be found in Arni-Bloch et al. (2009). We use this case for illustration purposes.

4.2.2 *Process Model for New Service Integration into ISS*

The process model for new service integration into ISS is formalized in five steps (Ralyté et al. 2010), summarized as follows.

[3] http://www.admin.ch/opc/fr/classified-compilation/19930224/index.html.

Step 1: Identify and Characterize Overlap The first step of the approach consists in identifying and characterizing the overlap between the four information spaces (static, dynamic, rule, and role) of the new service and the services already existing in the ISS.

- Specification of the static space overlap consists in identifying overlap relationships between classes of the new service and the ISS. A class is in overlap if it is used in the definition of the static space of more than one service. For each class in overlap, the overlap report has to specify if there are any structural and/or semantic disparities that should be resolved to enable service integration. For example, both services, DMS and ORS, contain classes "Person", "Student", "Diploma", with the same or with a different structure, which represent the static space overlap between these services.
- Specification of the dynamic space overlap consists in identifying actions in the new service and in the ISS which have some functional overlap, e.g., produce the same effect on a class in overlap such as updating an attribute or creating a new object of this class. An effect is in overlap if it can be generated by more than one service. In our case, the effects "update student information", "update registration to diploma date" are examples of dynamic space overlap.
- Specification of the rule space overlap consists in analyzing IR defined in the service and the ISS rule spaces, and identifying similarities and potential inconsistencies, e.g., considering if the integrity rule defined in the service rule space could be violated by the actions of the ISS and vice versa. The set of classes and attributes that participate in the validation of the integrity rule *ir* define its validation context (*context(ir): {class}*). An integrity rule is in overlap if any classes of its context are in overlap. Therefore, a rule space overlap appears when the same class is governed by different rules in different services. For example, the IR that validate the creation and modification of objects in classes "Person", "Student", and "Diploma" are in overlap because these classes are in overlap. But, if an integrity rule (e.g., "UniversityRegistrationRequest.dateOfRequest < Diploma.registrationLimit") is defined in one service (e.g., ORS) but not in the other service (e.g., DMS) and has in its context a class which belongs to the static space of both services (e.g., "Diploma"), this rule is in inconsistent overlap.
- Finally, specification of the role space overlap deals with the analysis and characterization of roles defined in the service and the ISS and their responsibilities for the corresponding service actions. A role is in overlap if it belongs to several services. In our case, roles such as "Student" and "Diploma Administrator" belong to both services, DMS and ORS.

Step 2: Validate Overlap Conformity Once the overlap report is specified, the conformance of each couple of overlapping elements (classes, actions, rules and roles respectively) has to be evaluated. Two elements are considered as conform to each other if they can be substituted one by the other. That means, two classes conform if they have the same name, their sets of attributes and sets of methods are identical and they have the same super-classes. For example, the structure of the class "Person" could be slightly different in the services DMS and ORS (e.g., the attribute Birthdate could be missing in one of the classes), and would then represent a non-conforming static space overlap. Similarly, two conforming actions could have the same name, and identical sets of parameters and effects. Two rules are in conformity if they have the same name and their contents are identical. Finally, two roles conform if they have the same name and there is no inconsistency in their responsibilities. If two overlapping elements are not in conformity, one of them (generally the element of the new service) has to be adapted.

Step 3: Settle Overlap Conformity The role of this step is to settle the overlap conformity for each couple of non-conforming overlapping elements according to the type of identified disparity. The disparities can be of semantic nature (the same name but different meaning or vice versa) or of structural nature (different set of attributes in similar classes) and lead to the appropriate unification.

Step 4: Organize Overlap At this step the overlap has to be organized in a way that clarifies the relationships between the new service and the legacy ones and eliminates all inconsistencies. Depending on the situation, it can be necessary to adapt service responsibilities on common elements – to determine the effects (create, update, delete, etc.) that the service is still authorized to realize on the shared data while being compliant with the legacy services. In order to clarify the visibility of effects on common elements it is important to define a cooperation protocol for each of them. Finally, to make sure that the new service respects the regulation policy of the ISS, we need to guarantee that IR defined on legacy services will not be violated by the new service after its integration.

Step 5: Consolidate Integration The integration of a new service can generate new situations that did not exist before, neither in the new service nor in the legacy ones. Therefore, it can be necessary to add new actions, rules, roles, and/or authorizations to face these situations and to consolidate the integration.

4.3 Information Kernel-Based ISS

In our third approach for ISS engineering we propose an intermediate architecture founded on the information kernel, which can be extended by several information services.

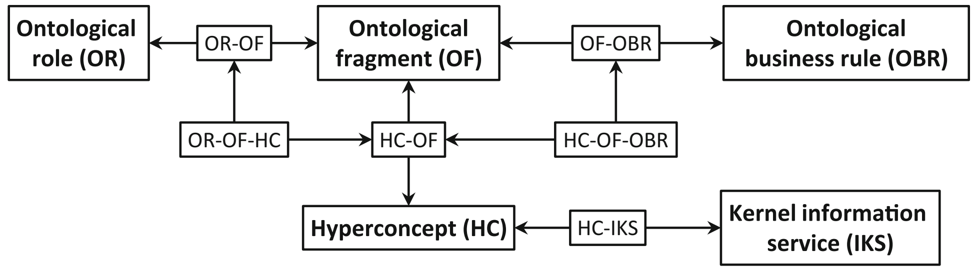

Fig. 5 Ontological level information kernel metamodel (simplified)

The information kernel captures the invariant data, processes and rules, while information services offer capabilities for business activities that are subject to change. The main challenges of this approach are the definition of the information kernel, which is formalized as a collection of kernel services, and the preservation of this kernel when adding new services to the ISS.

In this approach we consider that the core and invariant information (including data, processes, rules and roles) can be found in laws and other regulation policies governing enterprise activities, and should be captured in the kernel of the ISS independently of the information services that could extend this ISS later. This approach is especially adapted for ISS development in public and governmental sectors. We argue that legal documents include precise definitions of concepts, rules and constraints governing institutional activities and represent a rich source of knowledge for the ontological information extraction and the information kernel conceptualization. Moreover, the use of laws permits to enhance the adequacy and compatibility of an institutional IS with the corresponding institution activities and to construct a stable information kernel as a basis for the sustainable ISS development. Therefore, this approach consists in constructing the ontological level model based on the analysis of different legal sources, and in a next step mapping it into the conceptual model representing the kernel ISS. For the extension of this kernel ISS with new services we can then follow the Fully Service-Oriented ISS engineering approach presented above.

4.3.1 Process Model for the Construction of Information Kernel-Based ISS

The process model of this approach is formalized in three steps with the first two dealing with the construction of the ontological level model while the third one formalizes the conceptual level of the kernel information services.

Step 1: Identify the Hyperconcepts The first step of this approach consists in identifying the hyperconcepts that correspond to different ontological information spaces related to the organization activities. As depicted in Fig. 5 representing the metamodel for the ontological level information kernel construction, we use the notion of the HC to capture the information related to a fragment of any legal and/or ontological source named here an Ontological Fragment (OF). A HC is composed of a set of concepts extracted from ontological fragments and represents an ontological unity with precise semantics. Ontological Business Rules (OBR) can be extracted from one or several ontological fragments and are related to one or more hyperconcepts (class HC-OF-OBR). The notion of Ontological Role (OR) represents a set of responsibilities and permissions to perform business activities in the organization. Each ontological role is defined in at least one ontological fragment and is valid in the context of one or more hyperconcepts (class OR-OF-HC). The identification of hyperconcepts can be achieved in two complementary ways: (1) by selecting and analyzing different ontological fragments and (2) by analyzing the organization and identifying its ontological roles. An ontological fragment is selected if it is considered to be stable and invariant for the lifespan of both the organization and the ISS under construction. It is obvious that this type of decision requires some experience and risk management abilities.

Step 2: Build the Hyperconcepts This step completes the ontological model construction by extracting concepts and business rules from the previously selected ontological fragments and by defining the related hyperconcepts. Their structure is refined by means of a set of validation criteria based on the ontological level metamodel (Fig. 5).

Step 3: Construct Kernel Information Services This step consists in constructing the information services which compose the kernel of the ISS. As shown in Fig. 5, we use the notion of kernel information service (IKS) as an autonomous part of the kernel ISS. It is based on one or more hyperconcepts from the ontological level (HC-IKS). The structure of the kernel information service is formalized by using the information service metamodel shown in Fig. 2. We use a collection of mapping rules to extract from the ontological model the four information spaces (static, dynamic, rules, and roles) of each kernel information service which is a part of the conceptual model of the kernel ISS.

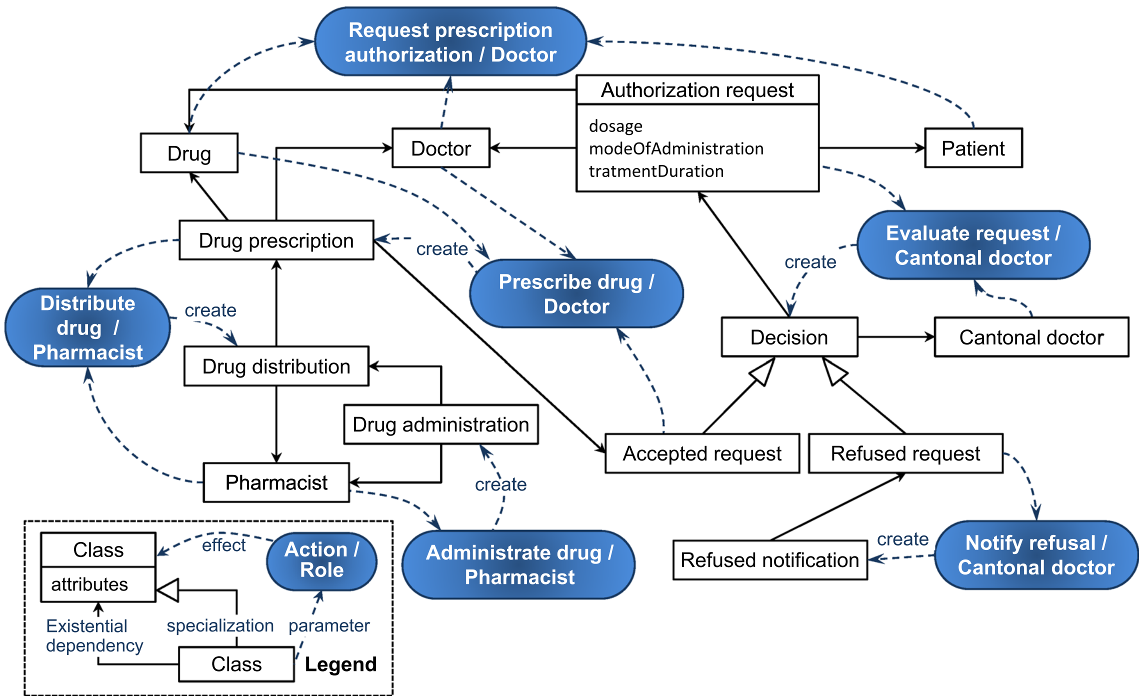

Fig. 6 Example of ISS kernel information model (adapted from Khadraoui et al. 2009)

4.3.2 Illustrating Example

The approach was applied in several e-government projects (e.g., Khadraoui et al. 2008, 2009) in collaboration with the Information Technology Center of the State of Geneva (Switzerland). One of the projects concerned the development of the kernel ISS for managing the prescription and delivery of narcotics for the treatment of addicted. We have used the Geneva law K 4 20.06[4] as a legal source of knowledge to build and to implement the kernel ISS supporting the activities of prescription, distribution and administration of narcotics. This law describes the procedure that medical doctors have to follow in order to record a request of authorization allowing the prescription of a narcotic for a dependent person's (drug addict's) treatment. For example, the law says that a doctor has to obtain authorization from the cantonal doctor before the prescription of any narcotic to a patient. Then, the pharmacist, on the basis of the authorization delivered by the cantonal doctor, provides the doctor, or directly the patient, with the drug prescribed. From the law text we can identify a set of invariant concepts (Doctor, Patient, Authorization, Cantonal Doctor, Prescription, Drug, etc.). The law also describes how the drug has to be distributed and administered by listing different cases in which a patient can benefit of the administration of narcotics. Thus, the law additionally defines business rules and constraints (ontological business rules) to be respected. Finally, the law enables us to identify organizational roles (ontological roles) such as Doctor, Pharmacist, Cantonal Doctor, etc. The kernel ISS was built directly from this law by extracting first its ontological model and then mapping this into the conceptual model of the kernel information services: treatment with narcotics authorization request, request approval, prescription of narcotics, etc. An excerpt of this model is illustrated in Fig. 6.

[4] Today, Geneva Law K 4 20.06 is abrogated and replaced by the new law K 4 20.02 http://www.geneve.ch/legislation/rsg/f/rsg_k4_20p02.html.

4.4 Summary and Usage Contexts

The three approaches presented in this paper are aimed at supporting different services-oriented IS engineering situations. Because they are based on the same fundamental concepts, namely information service and ISS, they can be easily combined.

The first approach, named information services upon legacy IS, applies to the situation where multiple legacy IS have to continue to operate and cannot be transformed drastically. This context is common to large and/or highly structured organizations, such as public administration, containing tens or even hundreds of IS usually comprising overlapping data and functionalities. This approach helps to preserve existing IS by creating information services upon them via a common base capturing their overlap. The main difficulties of this approach are related to:

- The number and size of existing IS that have to be analyzed and used by the information services,
- The organizational changes (e.g., new roles, new responsibilities) entailed by the introduction of new services,
- The possibility of data opening and sharing, and
- The assignment and/or sharing of the responsibilities for the common base.

The second approach, named fully service-oriented ISS, is dedicated to support incremental ISS construction as well as legacy IS evolution by extending it by new information services. It is applied in the context of evolution of one particular IS and requires a prior "decomposition" of this IS into information services. This approach is based on the analysis and resolution of the overlap inconsistencies between the legacy information services and the new ones. The main challenge is to preserve as much as possible of the legacy services by rather modifying the new services if necessary. The approach defines several different ways (formalized in terms of method chunks (Ralyté et al. 2010) to handle different overlap inconsistencies, each of them having a different impact on the integration outcome.

Finally, the third approach, named information kernel-based ISS, aims to construct a stable and invariant basis for an ISS – the kernel information services – that could then be complemented with more volatile services following the second approach. According to this approach, information stability can be identified from the legal framework (laws and other regulation policies) governing enterprise activities. Therefore, this approach is especially suitable for ISS engineering in public administration, institutions and governmental sectors, such as e-government services, where laws and regulation policies do not change as fast as business rules in private sectors. In this approach, the kernel information services constitute a reference and a foundation for the ISS designers. They help understand which information is invariant and stable during the ISS lifecycle and design new services to be added to the ISS. Thus, various situations of services interoperability are identified, discussed and settled at early development stages.

5 Conclusion

Service-oriented paradigm seems to be well adapted to deal with the complexity, interoperability and evolution of enterprise legacy IS, and appears as the most promising one for the development of the next generation ISSs.

In this paper we aim to demonstrate how IS-specific SOA can be elaborated to deal with legacy IS evolution and new ISS design. The three approaches presented in this paper offer an overview of our work in the domain of service-oriented IS development, where we introduce the notions of information service and ISS. We demonstrate that the ISS development approach and the ISS architecture can be different depending on the enterprise legacy IS situation and its evolution strategy. For example, if there is a need to provide access to the resources of multiple existing IS while the requirement to preserve these IS untouched is very strong, the approach for defining services upon legacy IS helps to deal with such a situation. On the other hand, if a new services system has to be developed from scratch in order to support some new business or to computerize services that until now were provided manually, the information kernel-based approach seems to be the most appropriate. Finally, the fully service-oriented approach is particularly helpful to deal with the evolution of an IS which is already designed as a composition of services. Indeed, it provides guidance for extending existing ISS with new services.

The notion of information overlap is recurrent in the three approaches and it is also original in comparison with conventional SOA approaches, which consider software services as completely autonomous and independent modules. In the context of IS this type of autonomy is out of reach because different information services have to share data, roles and rules governing enterprise business.

The three approaches have been applied in various case studies and collaborative projects in the sector of public administration. Our current preoccupation is to apply them in different industrial settings and to expand them by using additional techniques.

References

Aier S (2012) Strategies for establishing service oriented design in organizations. In: Proceedings of 33rd ICIS 2012, Orlando

Aier S, Bucher T, Winter R (2011) Critical success factors of service orientation in information systems engineering: derivation and empirical evaluation of a causal model. Bus Inf Syst Eng 3(2):77–88

Almonaies A, Cordy J, Dean T (2010) Legacy system evolution towards service-oriented architecture. In: Proceedings of international workshop SOAME 2010, IEEE, pp 53–62

Arni-Bloch N, Ralyté J (2008) MISS: a metamodel of information system service. In: Papadopoulos GA, Wojtkowski W, Wojtkowski G, Wrycza S, Zupancic J (eds) Information systems development. Springer, Heidelberg, pp 177–186

Arni-Bloch N, Ralyté J, Léonard M (2009) Service-driven information systems evolution: handling integrity constraints consistency. In: Proceedings of PoEM 2009, LNBIP 39. Springer, Heidelberg, pp 191–206

Baida Z, Gordijn J, Omelayenko B (2004) A shared service terminology for online service provisioning. In: Janssen M, Sol HG, Wagenaar RW (eds) Proceedings of ICEC 2004, ACM, pp 1–10

Bichler M, Lin KJ (2006) Service-oriented computing. IEEE Computer, pp 88–90

Bieberstein N, Bose S, Walker L, Lynch A (2005) Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. IBM Syst J 44(4):691–708

Booth D, Haas H, McCabe F, Newcomer E, Champion M., Ferris C, Orchard D (eds) (2004) Web services architecture. W3C Working Group Note. http://www.w3.org/TR/ws-arch/. Accessed 2014-08-14

Channabasavaiah K, Yuggle E, Holley K (2003) Migrating to a service-oriented architecture. http://www.ibm.com/developerworks/webservices/library/ws-migratesoa/. Accessed 2014-08-14

Chua FF (2009) Adoption of service-oriented architecture by information systems. Int J Auton Adapt Commun Syst 2(4):317–330

Cuadrado F, Garcia BJ, Dueas B, Parada H (2008) A case study on software evolution towards service-oriented architecture. In: Proceedings of 22nd international conference on advanced information networking and applications-workshops (AINAW'08), IEEE, pp 1399–1404

El Fazziki A, Lakhrissi H, Yétongnon K, Sadgal M (2012) A service oriented information system: a model driven approach. In: Proceedings of 8th international conference SITIS 2012, IEEE, pp 466–473

Erl T (2007) SOA principles of service design. Prentice Hall PTR, Upper Saddle River

Estrada H, Martinez A, Pastor O, Mylopoulos J, Giorgini P (2010) Extending organizational modeling with business services concepts: an overview of the proposed architecture. In: Proceedings of ER 2010, LNCS 6412. Springer, Heidelberg, pp 483–488

Gordijn J, van Eck P, Wieringa R (2009) Requirements engineering technique for e-services. In: Georgakopoulos D, Papazoglou MP (eds) Service-oriented computing, cooperative information systems series. MIT Press, Cambridge, pp 331–352

Khadka R, Reijnders G, Saeidi A, Jansen S, Hage J (2011) A method engineering based legacy to SA migration method. In: Proceedings of 27th IEEE ICSM 2011, pp 163–172

Khadraoui A, Opprecht W, Aïdonidis C, Léonard M (2008) Laws-based ontology for e-government services construction case study: the specification of services in relation-ship with the venture creation in Switzerland. In: Proceedings of PoEM 2008, LNBIP 15. Springer, Heidelberg, pp 197–209

Khadraoui A, Léonard M, Pham Thi TT, Helfert M (2009) A framework for compliance of legacy information systems with legal aspect. AIS Trans Enterp Syst 1:15–26

Khadraoui A, Opprecht W, Léonard M, Aïdonidis C (2011) Service specification upon multiple existing information systems. In: Proceedings of IEEE RCIS 2011, IEEE

Krafzig D, Banke K, Slama D (2004) Enterprise SOA: service-oriented architecture best practices. Prentice Hall PTR, Upper Saddle River

Le Dinh T, Nguyen-Ngoc AV (2010) A conceptual framework for designing service-oriented inter-organizational information systems. In: Proceedings of SoICT 2010, pp 147–154

Le Dinh T, Pham-Thi TT (2010) A conceptual framework for service modelling in a network of service system. In: Proceedings of IESS 2010, LNBIP 53. Springer, Heidelberg, pp 192–206

Legner C, Heutschi R (2007) SOA adoption in practice—findings from early SOA implementations. In: Proceedings of 15th European conference on information systems (ECIS)

Liu Y, Wang Q, Zhuang M, Zhu Y (2008) Reengineering legacy systems with RESTful web service. In: Proceedings of COMPSAC 2008, IEEE, pp 785–790

Lo A, Yu E (2007) From business models to service-oriented design: a reference catalog approach. In: Proceedings of ER 2007, LNCS 4801. Springer, Heidelberg, pp 87–101

MacKenzie CM, Laskey K, McCabe F, Brown PF, Metz R (eds) (2006) Reference model for service oriented architecture 1.0, Oasis standard. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html. Accessed 2014-08-14

Papazoglou MP (2008) The challenges of service evolution. In: Proceedings of CAiSE 2008, LNCS 5074. Springer, Heidelberg, pp 1–15

Quartel DAC, Steen MWA, Pokraev S, van Sinderen M (2007) Cosmo: a conceptual framework for service modelling and refinement. Inf Syst Fron 9(2–3):225–244

Ralyté J (2012) Applying transdisciplinarity principles in the information services co-creation process. In: Proceedings of IEEE RCIS 2012, IEEE. doi:10.1109/RCIS.2012.6240429

Ralyté J, Arni-Bloch N, Léonard M (2010) Information systems evolution: a process model for integrating new services. In: Proceedings of AMCIS 2010, paper 431

Ralyté J, Khadraoui A, Léonard M (2013) From information systems to information services systems: designing the transformation. In: Proceedings of PoEM 2013, LNBIP 165. Springer, Heidelberg, pp 69–84

Ransom J, Sommerville I, Warren I (1998) A method for assessing legacy systems for evolution. In: Proceedings of CSMR'98, IEEE, pp 128–134

Reddy VK, Dubey A, Lakshmanan S, Sukumaran S, Sisodia R (2009) Evaluating legacy assets in the context of migration to SOA. Softw Qual J 17(1):51–63

Regev G, Hayard O, Wegmann A (2011) Service systems and value modeling from an appreciative system perspective. In: Proceedings of IESS 2011, LNBIP 82. Springer, Heidelberg, pp 146–157

Sneed HM (2006) Integrating legacy software into a service oriented architecture. In: Proceedings of CSMR 2006, IEEE, pp 3–14

Spohrer J, Maglio PP, Bailey J, Gruhl D (2007) Steps toward a science of service systems. Computer 40(1):71–77

Spohrer J, Vargo SL, Caswell N, Maglio PP (2008) The service system is the basic abstraction of service science. In: Proceedings of HICSS 2008, pp 104–114

Thomas O, vom Brocke J (2010) A value-driven approach to the design of service-oriented information systems—making use of conceptual models. Inf Syst E-Business Manag 8(1):67–97

Turki S, Léonard M (2002a) Hyperclasses: towards a new kind of independence of the methods from the schema. In: Proceedings of ICEIS'2002, vol. 2. pp 788–794

Turki S, Léonard M (2002b) IS components with hyperclasses. In: Proceedings of OOIS 2002, LNCS 2426. Springer, pp 132–141

Umar A, Zordan A (2009) Reengineering for service oriented architectures: a strategic decision model for integration versus migration. J Syst Softw 82(3):448–462