

# From Information Systems to Information Services Systems: Designing the Transformation

Jolita Ralyté, Abdelaziz Khadraoui, and Michel Léonard

University of Geneva, Institute of Services Science  
CUI, Battelle bat. A, 7 Route de Drize, CH-1227 Carouge, Switzerland  
{Jolita.Ralyte,Abdelazis.Khadraoui,Michel.Leonard}@unige.ch

**Abstract.** Service-orientation is currently considered as a promising paradigm to deal with the complexity, interoperability and evolution of enterprise Information Systems (IS), which are the foremost preoccupation in today's enterprises. However, the shift from a conventional IS architecture to a service-oriented one is not an easy task despite of the various service design approaches proposed in the literature. In this paper we promote the concepts of information service and Information Services System (ISS) and we present three different ways to design an ISS taking into account enterprise legacy IS and/or from scratch. We illustrate the three approaches with examples taken from industrial projects and case studies.

**Keywords:** Information service, information services system, service-oriented paradigm, IS evolution.

## 1 Introduction

Sustainability and evolution of legacy Information Systems (IS), including their maintenance, extension with new components, interoperability with other systems and applications, is one the foremost preoccupation in today's enterprises, not only at technical but also at information and business strategic levels. In addition, the need for inter-organizational and networked information systems, cloud platforms and services systems is growing because of the enterprise business models transformations into networked and service-oriented ones. At the same time, it is unthinkable to replace existing IS by the new ones for each enterprise business and/or organizational change. Legacy IS has to evolve together with enterprise changes, and this evolution can take different forms: integration of new components from the market or custom-made, development of services on top of the existing IS, establishment of interoperability between two or more IS, etc.

In this context, service-oriented approaches emerge as prospective ones to deal with IS fragmentation, interoperability and evolution problems [1, 5, 19] as well as to support inter-organizational IS development [14, 15, 18]. Modularity, reusability and evolution are considered as the main values that service-oriented paradigm brings to the IS domain. To reach modularity, an IS has to be composed of a collection of

interrelated and autonomous components. The notion of service, and in particular information service (see section 2.1), is introduced to cope with IS modularity. Modularity is the necessary basis to ensure incremental and evolutionary IS development and therefore a prerequisite for legacy IS evolution with new components/services. It also allows to avoid chaotic IS fragmentation. Reusability, as opposed to the more expensive “from scratch” development, means the reuse of legacy data and applications to provide new business functionalities – new services. Finally, the evolution principle consists in the ability to easily replace an existing component/service by a new one.

The literature review demonstrates the advent of proposals to redesign conventional IS architectures into the service-oriented ones [5, 9, 14, 15, 19, 22]. The notion of information services system (ISS) introduced in [3] (see section 2.2) appears as a natural evolution of the IS concept. Similarly to IS, ISS emphasizes the value of information, its creation, management and sharing, while improving its modularity, agility and interoperability.

The research question considered in this paper is how enterprise information systems could evolve from conventional to the service-oriented ones. In particular, we aim to identify and explore different ways to lead the shift from traditional information systems towards information services systems. Based on the related literature survey, our previous works [3, 4, 10, 11, 12, 19] and various projects realized in collaboration with the Information Technology Center of the State of Geneva (some of them are reported in [10, 11, 12]), we present in this paper three different but complementary approaches to design information services systems based on enterprise legacy IS and/or from scratch. The three approaches have been tested in various projects and case studies that we mention in this paper.

The rest of the paper is organized as follows: in the next section we define the notions of information service and information services system. Then, in section 3 we present and illustrate three approaches supporting ISS design. We discuss the related work and how our contribution complements it in section 4. Finally, section 5 concludes the paper and highlights future perspectives.

## 2 Information Services and Services Systems

### 2.1 Information Service

In the domain of enterprise information systems, 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 [25], which is defined over classes, methods, integrity rules, 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, transformation and transmission of information. This type of service is named an *information service* [3, 4] and 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 information

system can be seen as built of a collection of interoperable information services. The metamodel of the information service is an extension of other computational services metamodels such as: WSPER [6] or the model proposed in [2]. The particularity of the information service definition is that it requires to make the service transparent, i.e. it explicitly distinguishes four interrelated information spaces – static, dynamic, rule and role – that are necessary to define service resources and capabilities in the organizational context. We claim that in the IS context it is not sufficient to consider services as black boxes with only 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. Fig.1 shows the simplified metamodel of the information service where only the main concepts are represented (see [3] for the detailed version).

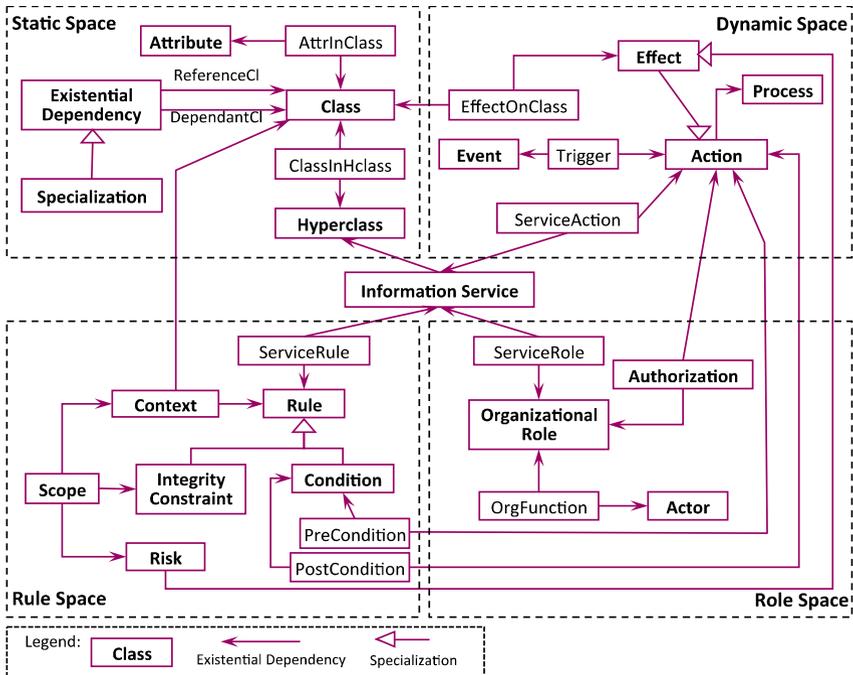


Fig. 1. Simplified metamodel of the information service

The *static space* of the service defines its data structure in terms of classes and relationships between classes. The notion of Hyperclass (introduced in [24] to specify information system components) is used to represent complex domain concepts by putting together the corresponding set of classes. Classes are linked only via existential dependencies and specialization relationships. An existential dependency is materialized via an attribute with mandatory and permanent constraints. The *dynamic space* defines service capabilities in terms of actions that can be executed by the service and their effects on service classes. An action is triggered by an event that occurs in the service information space and is described by a process to be executed, which

can be a simple function or a more complex interaction involving several actors. An action produces one or more effects on the static space (e.g. create an object of a class, modify an attribute). The notion of effect is used to characterize the result of the action and allows to evaluate the impact of the action on the rule space. The *rule space* deals with service regulation policies, which are formalized as pre-, post-conditions on service actions and integrity constraints on service data. An integrity constraint has a scope, which includes all the effects that could transgress the rule. Such an effect is called a risk of the rule. Finally, the *role space* defines service actors, the roles they play in the organization and the rights and responsibilities they have on service actions.

The metamodel presented above represents a 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 information services system.

## 2.2 Information Services System

According to [21, 22], 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 a mutual value. Spohrer et al., [21] 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 information systems point of view. An actor of an information service can be allowed to provide and to consume service information and capabilities. In this case the term “service prosumer” fits better the situation. Though, the notion of value coproduction is also key in the domain of information systems.

In [21], 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 Information Services Systems (ISS) people are not considered as independent system components but rather as actors enabling capabilities of information services. Their behavior is specified through the organizational roles and responsibilities involved in the information services enactment (provision and/or consumption of information resources) and governed by a set of rules implemented 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, information services systems aim to stay dynamic and open in order to enable innovation and facilitate their evolution. They have to take a risk to give people some liberty to informally and formally change rules and policies.

In our approach, an ISS is seen as a collection of interoperable information services. It 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.

### 3 Designing Information Services Systems

There are certainly many different ways to design Information Services Systems (ISS). In this work we focus our attention on the evolution of conventional enterprise IS into the service-oriented ISS and we discuss different transformation situations and approaches. In particular, we identify and illustrate three approaches taking into consideration different organizational contexts and ISS design situations, and legacy IS reuse:

- *Services upon legacy IS*. This approach aims to bring some flexibility and modularity to the rather monolithic and fragmented legacy enterprise information systems without inflicting to them any major transformation. Indeed, in this approach services are created upon legacy IS. They utilize resources existing in various IS (their data, processes, rules, responsibilities) and provide some added value to the service users. The approach consists in identifying for each new service the existing resources that are potentially scattered in different IS and to guarantee 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.
- *Fully service-oriented ISS*. This approach, in the contrary, considers an information system as a composition of information services and for that needs a more deep transformation of the existing IS into a services-oriented one. Indeed, the approach consists in “decomposing” an IS into a collection of information services and defining the overlap (common data, activities, roles, rules) between them. Information overlap management is the most important difficulty when including new services into an existing ISS.
- *Information kernel-based ISS*. This approach proposes an intermediate architecture based on a core IS and information services as its extensions. The core IS captures the kernel information – the invariant data, processes and rules, while information services offer capabilities for business activities that are subject to change. In this type of architecture the main challenges are (1) the definition of the information kernel, which is formalized as a collection of kernel services, and (2) the preservation of this kernel when adding new services to the ISS.

#### 3.1 Information Services upon Legacy IS

Even though enterprises are constantly seeking to renew the range of their IS to be compliant with their business evolution, replacing or transforming legacy IS can be very expensive and error-prone or even impossible. Introducing a service-oriented layer upon multiple legacy systems is considered here as a potential solution to limit the transformation of legacy IS and at the same time to bring some flexibility and agility to them. In this approach we consider that several existing IS and enterprise applications are information providers to the newly defined services. The identifica-



*Step 1: Construct a frame of reference* means for each service to define: (1) its organizational contexts and the existing IS containing the information necessary for the service execution, (2) the information necessary for the service implementation in the form of a conceptual model, (3) the roles responsible for its execution. The frame of reference is constructed by instantiating the *Service Analysis Metamodel* shown in Fig. 2b. This metamodel defines the key concepts necessary to link an information service to the existing IS that provide data necessary to the service execution. A service can be defined on one or several *Organizational Contexts* that describe business rules, legal constraints and the capability of the organization to enforce laws and policies. The notion of *Hyperconcept* is used to specify how the organizational contexts are formalized in different information systems. In fact, a hyperconcept is a complex concept (composed of several sub-concepts) representing a semantic unity in the domain of analysis. Finally, the *Role* concept represents the responsibilities and authorizations to execute IS activities in a particular organizational context.

*Step 2: Select the candidate informational elements* for the common base by analyzing the information collected in the frame of reference. This analysis can be done in three different and complementary ways: by organizational context analysis, by organizational role and/or by hyperconcepts analysis respectively. Typically, if a hyperconcept representing a part of the service organizational context is implemented in more than one IS, the data representing it is candidate to the common base. The same applies to the roles related to the service organizational context.

*Step 3: Construct a common base* by collaborative decision making where all project stakeholders (business practitioners, IS architects, database architects, etc.) are invited to discuss on several 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* by adding new elements (e.g. adding new role) or transforming the existing ones (e.g. changing a business rule) in order to guarantee that each legacy IS is interoperable with the new common base. The number and complexity of identified transformations indicate the weight of the impact of the new service creation on the legacy IS, and this impact has to be minor.

This approach was applied in collaboration with the Information Technology Center of the State of Geneva (Switzerland) to design new services for e-administration based on five existing independent information systems (see Fig.3). Three 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, and (3) the Geneva Business Repertory (REG) that contains administrative information on businesses and 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. These cantonal information systems have to interact with two similar IS in use at the federal level: the Federal Commercial Register (RCF) and the Federal Business Repertory (REE). Several information services were identified for this project. For example, one of them concerned the "Transmission of business statistical data" and should help companies to transmit their data to the cantonal and federal offices in order to build statistics.

The organizational context of this service is based on two laws on public statistics, one at the cantonal level (LStat)<sup>1</sup> and the other at the federal level (LSF)<sup>2</sup>, and the data related to this context is available in two information systems - the REG and REE. The construction of the common base consisted in the selection of elements (data, roles, rules) common to both concerned IS and necessary for the service execution, and also the creation on new elements such as new roles for the execution of service activities. The main difficulties encountered in this project were related to the common base management, in particular to decide who will be responsible for this common base and how to manage the existing flows between the two IS (REG and REE).

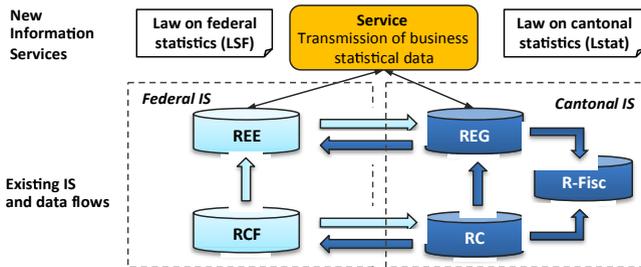


Fig. 3. Example of Service Organizational Context Analysis

### 3.2 Fully Service-Oriented ISS

The second approach for ISS engineering considers the enterprise IS as a composition of information services where each service provides a support for a particular business or administration activity. It differs from other service-oriented approaches in many perspectives. For example, the granularity level of information services is quite large – a service represents a full business unit not only a simple action or application. A service can represent an inter-organizational collaboration and value exchange between organizations as well as, and in particular, inter-organizational activities.

Indeed, most of the existing service-oriented approaches deal with the design of different computational services (e.g. web, mobile, cloud) and their supporting platforms in a pure “service customer vs. provider” frame. Up to now, only little attention has been paid to the intra-organizational IS context and legacy IS transformation into the service-oriented one. In [4, 19] a service-oriented and situation-driven approach for legacy IS evolution has been proposed. In this paper, we argue that this approach can be used for ISS engineering in an incremental way by progressively integrating new information services into an ISS. As said above, in this approach an IS is seen as a collection of information services, at least at the conceptual level. The main difficulty of this approach is in the fact that services composing the 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 exist in the four information spaces (static, dynamic, rule and role) (see Fig. 1). The construction of an ISS

<sup>1</sup> [http://www.geneve.ch/legislation/rsg/f/s/rsg\\_B4\\_40.html](http://www.geneve.ch/legislation/rsg/f/s/rsg_B4_40.html)

<sup>2</sup> [http://www.admin.ch/ch/f/rs/431\\_01/index.html](http://www.admin.ch/ch/f/rs/431_01/index.html)

consists in integrating progressively new services into the existing ISS as shown in Fig. 4a. Of course, each new integration creates new overlap situations – some data, activities, roles and rules can be shared between the existing and new services. Therefore, this approach is based on the analysis and resolution of the overlap between legacy and new services by preserving the legacy ones as much as possible. The service integration approach is articulated in five main intentions with several strategies available to reach each of them as shown in Fig. 4b.

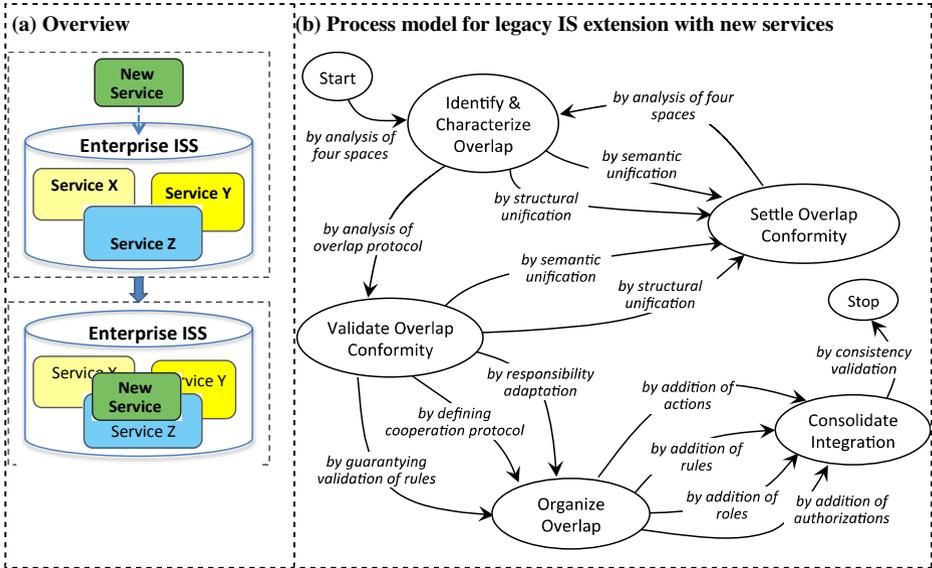


Fig. 4. The approach “Fully service-oriented ISS”

*Step 1: Identify and characterize the overlap* between the four information spaces of the new service and the services already existing in the ISS. An overlap report has to be elaborated for each space with the objective to identify similar elements and to evaluate if they need some structural or semantic unification or not. For example, two services can contain the same class (e.g. Student) with the same or different structure (e.g. the attribute Birthdate is missing in one of the classes), which represents a static space overlap. An example of the functional overlap would be when two services are producing the same effect on a class in overlap, as for instance updating an attribute or creating new objects of this class (e.g. create a new student). The rule overlap appears when the same class is governed by different rules in different services (e.g. the deadline for registering new students exist in one service but not in the other). Finally, the role space overlap means the existence of the same or similar roles in two services (e.g. Students’ administrator).

*Step 2: Validate the overlap conformity* for each couple of overlapping elements (classes, actions, rules and roles respectively) identified and characterized in the overlap report. That means to check if they conform to each other, i.e. can be substituted one by another. In the static space two classes are considered as conforming if they

have the same name, their sets of attributes and sets of methods are identical and they have the same super-classes. If two overlapping elements are not conforming, one of them (generally the element of the new service) has to be adapted in the next step.

*Step 3: Settle the overlap conformity* for each couple of non-conforming overlapping elements according to the type of the 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). Therefore, it can be necessary to modify the concerned classes, actions, rules and roles by semantic unification (i.e. renaming one of the elements) or by structural unification (i.e. transformation of the element structure) accordingly.

*Step 4: Organize the overlap* means to clarify the relationships between the new service and the legacy ones. 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 to be 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 legacy ISS, we need to guarantee that validity of rules, and especially of integrity constraints, defined on legacy services will not be violated by the new service after its integration.

*Step 5: Consolidate the integration* by revising the four service spaces and adding if necessary missing elements in each of them in order to ensure that the obtained integration is valid. In fact, 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, and addition of new actions, rules, roles and/or authorizations can be necessary to face these situations.

In order to validate this approach, we have realized a case study that concerned the extension of our 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 the new service to be integrated in the existing ISS. The DMS provides several diploma 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 diplomas 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 on-line by filling the registration form and by uploading different required documents. Then, the students' administrator validates the on-line created registration request and asks for additional documents if necessary. He/she is responsible for recording the candidate as a student and for registering him/her to the selected diploma.

Because, the two services have been developed independently, it is obvious that some information overlap between them is inevitable (e.g. class *Person*, class *Student*, activity *UpdatePerson*, rule *RgistrationToDiploma.RegistrationDate < Diploma.RegistrationLimitDate*, role *Student*, role *Diploma Manager*, etc.). These shared elements (identical or similar) constitute the information overlap between the existing and the new service and the integration process was to make this overlap consistent in order to enable reliable new service exploitation within the ISS and the non-violation of the already existing ISS consistency.

### 3.3 Information Kernel-Based ISS

The third approach for ISS engineering proposes to use a mixed architecture where a kernel IS is extended with a variety of information services. In fact, in this approach we consider that the core and invariant information (including data, processes, rules and roles) can be found in the laws and other regulation policies governing enterprise activities, and should be captured in the kernel of the IS independently of the information services that could extend this IS later. This approach is especially adapted for the ISS development in public and governmental sectors. We argue that legal documents include precise definitions of concepts, rules and constraints governing the 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 has two parts: (1) the ontological model construction based on the analysis of legal sources and (2) its mapping into the conceptual model representing the kernel IS as shown in Fig. 5a. This kernel IS then can be extended with new services following Fully Service-Oriented ISS engineering approach presented above.

Formally, the process model of the approach is defined in terms of three main intentions and several strategies to achieve each of them as depicted in Fig. 5c. The first two intentions, *Identify Hyperconcepts* and *Build Hyperconcepts*, deal with the ontological model construction in terms of a collection of interrelated hyperconcepts while the third one, *Construct Kernel Information Services*, defines the kernel IS as a collection of kernel information services.

*Step1: Identify the hyperconcepts* that correspond to the different ontological information spaces related to the organization activities. As depicted in Fig. 5b representing the metamodel for the kernel IS construction, we use the notion of the *hyperconcept* to capture the information related to a fragment of any legal and/or ontological source named here an *Ontological Fragment*. A hyperconcept is composed of a set of concepts extracted from ontological fragments and represents an ontological unity with precise semantics. *Ontological Business Rules* can be extracted from one or several ontological fragments and are related to one or more hyperconcepts. Besides, we use the notion of *Ontological Role*, which 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. The identification of hyperconcepts can be done 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 as stable and invariant for the lifespan of the organization and of the ISS under construction. It is obvious, that this type of decision requires some experience and risk management abilities.

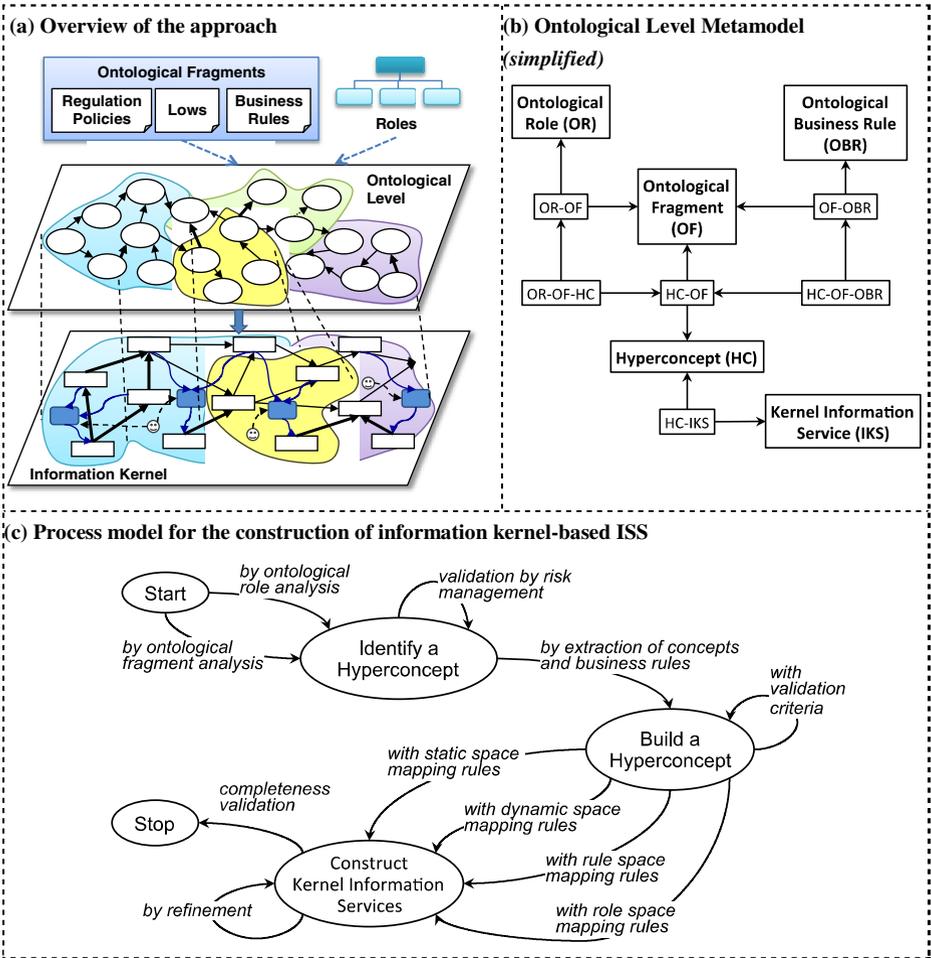


Fig. 5. The approach “Information kernel-based ISS”

*Step2: Build the hyperconcepts* by extracting concepts and business rules from the selected ontological fragments and refine their structure with a set of validation criteria based on the ontological level metamodel (Fig. 5b).

*Step 3: Construct information services* composing the kernel of the ISS. As shown in Fig. 5b, a *Kernel Information Service* is based on one or more hyperconcepts from the ontological level. The structure of the kernel information service is formalized by using information service metamodel shown in Fig. 1. 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.

This approach was applied in several e-government projects (e.g. [11, 10]) in collaboration with the Information Technology Center of the State of Geneva (Switzerland). One of the projects concerned the development of the kernel information services system for managing the prescription and delivery of narcotics for the

treatment of the addicted people. We have used the Geneva law K 4 20.06<sup>3</sup> as a legal source of knowledge to build and to implement the kernel ISS supporting the activities of narcotics prescription, distribution and administration. Indeed, this law describes the procedure that medical doctors have to follow in order to record a request of authorization allowing to prescribe a narcotic for the treatment of a dependent person (drug addict). For example, the law says that a doctor has to obtain an 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, the drug prescribed. From the law text a set of invariant concepts *{Doctor, Patient, Authorization, Cantonal Doctor, Prescription, Drug, etc.}* can be identified. The law also describes how the drug has to be distributed and administered by enumerating different cases where a patient can benefit of the administration of narcotics. Therefore, the law also defines business rules and constraints (ontological business rules) to be respected. Finally, the law allows to identify organizational roles (ontological roles) such as *Doctor, Pharmacist, Cantonal Doctor*, etc. A kernel ISS was built directly from this law.

## 4 Related Work and Discussion

Most of the current service-oriented approaches, like SOA [7, 13, 17] instruct how to develop service-oriented software systems from scratch and do not pay much attention to the legacy software, and in particular information systems, reuse and evolution. Moreover, the compliance of services to the laws and enterprise regulation policies is not considered. In the context of the IS engineering, these 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.

Up to now, we found relatively few attempts to define service-oriented architectures for information systems. We can mention the work of Chua [5] who discusses how service-oriented design should be applied in an organization in order to adopt SOA. Haesen [9] presents a research plan for developing an approach to design SOA for IS. Le Dinh and Nguyen-Hgoc [14] propose a conceptual framework for designing service-oriented inter-organizational IS. Thomas and Brocke [23] 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 [16] propose a reference catalogue approach to design an SOA system. This approach uses the *i\** modeling technique adapted to the service-oriented business modeling for the selection of reference business models

---

<sup>3</sup> 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](http://www.geneve.ch/legislation/rsg/f/rsg_k4_20p02.html)

from the catalogue and their adaptation to the particular case. Estrada et al. [8] introduce a service-oriented organizational model in order to reduce the mismatch between business models and service-oriented designs. Most of these approaches consider SOA for IS at business level rather than at information management and implementation level, and only from the customer-provider perspective – how the organization could offer services to its customers. The intra-organizational perspective of a service as a support for internal enterprise activities and information exchange is not really considered. Besides, the evolution of legacy IS into the service-oriented ones is not their major preoccupation.

Unlike the contributions mentioned above, the three approaches for ISS engineering summarized in this paper aim to support service-oriented inter- and intra-organizational ISS development taking into consideration legacy IS and their evolution. Each of the three approaches is dedicated to a particular situation but together they are considered as complementary ones.

The first approach, named information services upon legacy IS, deals with the situation where multiple legacy IS have to continue to operate and cannot be transformed drastically. It 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 new services, and the data opening and sharing.

The second approach, named fully service-oriented ISS, is dedicated to support incremental ISS construction as well as legacy IS evolution by extending them with new information services. This approach is based on the analysis and resolution of the overlap between the legacy information services and the new services and helps to preserve the legacy ones as much as possible thanks to the integration impact analysis.

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 extended with more volatile services. According to this approach, the information stability can be identified from the legal sources (laws and other regulation policies) governing enterprise activities and therefore this approach is especially suitable for the ISS engineering in public administration and governmental sectors such as e-government services. The kernel information services constitute a reference and a foundation for the ISS designers. They help to understand which information is invariant and stable during the ISS lifecycle and to 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 the most promising one to consider the development of the next generation information services systems.

However, the literature dealing with this topic is still modest. In this paper we aim to demonstrate how IS-specific service-oriented architectures can be elaborated, in particular to cope with the legacy IS evolution. The three approaches presented in this paper overview our work in the domain of service-oriented IS development where we introduce the notions of *information service* and *information services system (ISS)*. We aim to demonstrate that depending on the enterprise legacy IS situation and its evolution strategy the ISS development approach and the ISS architecture will be different. For example, if there is a need to open the access to the resources of multiple existing IS but the requirement to preserve these IS untouched is very strong, the approach for defining services upon legacy IS helps to deal with such situation. In the contrary, 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 or with the help of simple communication techniques like e-mail and/or fax, the information kernel-based approach seems to be the most appropriate one. 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 the conventional SOA, which considers software services as completely autonomous and independent modules. In the context of IS this type of autonomy is not reachable 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 future preoccupation is to apply them in different industrial settings and to extend them with additional strategies and techniques.

## References

1. Abbass, A.: Accessing Enterprise Information Systems in a Service-Oriented Architecture (2006), <http://www.ibm.com/developerworks/webservices/library/ws-soa-eisjca/>
2. Andrikopoulos, V., Benbernou, S., Papazoglou, M.P.: Managing the Evolution of Service Specifications. In: Bellahsene, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 359–374. Springer, Heidelberg (2008)
3. Arni-Bloch, N., Ralyté, J.: MISS: A Metamodel of Information System Service. In: Proceedings of ISD 2008, pp. 177–186. Springer Science+Business Media (2008)
4. Arni-Bloch, N., Ralyté, J., Léonard, M.: Service-Driven Information Systems Evolution: Handling Integrity Constraints Consistency. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 191–206. Springer, Heidelberg (2009)
5. Chua, F.F.: Adoption of Service-Oriented Architecture by Information Systems. *International J. of Autonomous and Adaptive Communications Systems* 2(4), 317–330 (2009)
6. Dubray, J.-J.: Wsper an abstract SOA framework. Technical report (2007), [http://www.ebpm1.org/wsper/wsper/wsper\\_primer.pdf](http://www.ebpm1.org/wsper/wsper/wsper_primer.pdf)
7. Erl, T.: SOA Principles of Service Design. Prentice Hall PTR (2007)

8. Estrada, H., Martínez, A., Pastor, O., Mylopoulos, J., Giorgini, P.: Extending Organizational Modeling with Business Services Concepts: An Overview of the Proposed Architecture. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 483–488. Springer, Heidelberg (2010)
9. Haesen, R.: Designing service-oriented information systems architectures. In: Proceedings of Workshops and Doctoral Consortium at CAiSE 2007, pp. 871–880 (2007)
10. Khadraoui, A., Léonard, M., Pham Thi, T.T., Helfert, M.: A Framework for Compliance of Legacy Information Systems with Legal Aspect. AIS Transactions on Enterprise Systems 1, 15–26 (2009)
11. Khadraoui, A., Opprecht, W., Aidonidis, C., Léonard, M.: Laws-Based Ontology for e-Government Services Construction. Case Study: The Specification of Services in Relationship with the Venture Creation in Switzerland. In: Stirna, J., Persson, A. (eds.) PoEM 2008. LNBIP, vol. 15, pp. 197–209. Springer, Heidelberg (2008)
12. Khadraoui, A., Opprecht, W., Léonard, M., Aidonidis, C.: Service Specification upon Multiple Existing Information Systems. In: Proceedings of RCIS 2011. IEEE (2011)
13. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall PTR (2004)
14. Le Dinh, T., Nguyen-Ngoc, A.V.: A Conceptual Framework for Designing Service-Oriented Inter-organizational Information Systems. In: Proceedings of SoICT 2010, pp. 147–154 (2010)
15. Le Dinh, T., Pham Thi, T.T.: A Conceptual Framework for Service Modelling in a Network of Service System. In: Morin, J.-H., Ralyté, J., Snene, M. (eds.) IESS 2010. LNBIP, vol. 53, pp. 192–206. Springer, Heidelberg (2010)
16. Lo, A., Yu, E.: From Business Models to Service-Oriented Design: A Reference Catalog Approach. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 87–101. Springer, Heidelberg (2007)
17. MacKenzie, C.M., et al. (eds.): Reference Model for Service Oriented Architecture 1.0, Oasis Standard (2006), <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html>
18. Ralyté, J.: Applying Transdisciplinarity Principles in the Information Services Co-creation Process. In: Proceedings of RCIS 2012. IEEE (2012)
19. Ralyté, J., Arni-Bloch, N., Léonard, M.: Information Systems Evolution: A Process Model for Integrating New Services. In: Proceedings of AMCIS 2010. Paper 431 (2010)
20. Rolland, C., Prakash, N., Benjamin, A.: A Multi-model View of Process Modelling. Requirements Engineering 4(4), 169–187 (1999)
21. Spohrer, J., Maglio, P.P., Bailey, J., Gruhl, D.: Steps Toward a Science of Service Systems. Computer 40(1), 71–77 (2007)
22. Spohrer, J., Vargo, S.L., Caswell, N., Maglio, P.P.: The Service System is the Basic Abstraction of Service Science. In: Proceedings of HICSS 2008, pp. 104–114 (2008)
23. Thomas, O., vom Brocke, J.: A value-driven approach to the design of service-oriented information systems—making use of conceptual models. Information Systems and E-Business Management 8(1), 67–97 (2010)
24. Turki, S., Léonard, M.: Hyperclasses: towards a new kind of independence of the methods from the schema. In: Proceedings of ICEIS 2002, vol. 2, pp. 788–794 (2002)
25. Turki, S., Léonard, M.: IS Components with Hyperclasses. In: Bruel, J.-M., Bellahsene, Z. (eds.) OOIS 2002. LNCS, vol. 2426, pp. 132–141. Springer, Heidelberg (2002)