Flexible Service Systems

Artem Polyvyanyy and Mathias Weske

Abstract Service science combines scientific, management, and engineering disciplines to improve the understanding of how service systems cooperate to create business value. Service systems are complex configurations of people, technologies, and resources that coexist in a common environment of service provisioning. While the general concepts of service science are understood and agreed upon, the representation of service systems using models is still in its infancy. In this chapter, we look at business processes and their role in properly representing service systems. We propose flexible process graphs, a high-level process modeling language, and extend it in order to specify service systems and their compositions within shared environments in a flexible way. The discussion in this chapter is the first step towards a formal description of service science environment, including service systems, networks, and whole ecology.

Keywords: Service science, service systems, flexible process graph, flexible service systems, modeling

1 Introduction

Service computing and service oriented architectures (SOA) have gained increasing attention recently as a new way of designing complex software systems consisting of service components (Burbeck, 2000; Gottschalk, 2000; Newcomer and Lomow, 2004). To provide business agility, one of the main promises of SOA is to bridge the gap between business aspects and information technology. Despite considerable efforts by industry and standardization consortia, as of today, this gap remains.

Business Process Technology Group
Hasso Plattner Institute at the University of Potsdam
Prof.-Dr.-Helmert-Str. 2–3, D-14482 Potsdam, Germany
Artem.Polyvyanyy@hpi.uni-potsdam.de
Mathias.Weske@hpi.uni-potsdam.de

The term service oriented architecture was coined by the computer science community, focusing on how software functionality can be specified, wrapped and discovered to be easily re-usable. It turns out that the technical point of view taken by available approaches is too narrow to provide a solid understanding of service environments. Therefore, service science has been founded, an integrating discipline that investigates not only technical aspects of services, but also their economical and organizational foundation (Spohrer et al, 2008).

While the main concepts and the overall design of service systems are beginning to shape, the representation of service systems is still in its infancy. In computer science, complex systems are represented by models, such as data models in database design (Silberschatz et al, 1997) or process models in the design of process oriented information systems (Weske, 2007). Once the modeling technique for capturing service systems and their environments is in place, they can be studied, analyzed, compared, classified, etc.

In this chapter, we introduce an approach to model service systems using flexible process graphs (Polyvyanyy and Weske, 2008a,b), which is a method for modeling business processes with a limited degree of structuring. Limited process model structuring leaves an opportunity for flexible behavior within captured process scenarios and supports straight-forward merging of models to reflect partner relations between service systems. The flexibility of the process is essential, since service episodes are in general not following a strict and well-defined business process, but result from a rather loose couplings of independent service systems, particularly if we talk about professional, scientific, and technical service systems.

The rest of the chapter is organized as follows: In the next sections we give preliminaries on service systems and flexible process graphs. Afterwards, in section 4, an approach of employing flexible process graphs to represent service environments is presented. Section 5 illustrates the concepts by an example. Concluding remarks complete this chapter.

2 Service Systems

This section introduces the main concepts of service systems, identifies their key properties and motivates the use of modeling techniques to represent service systems.

2.1 Foundation

In the foundation of service science, the transition from a Goods-Dominant to a Service-Dominant (S-D) logic of value creation has played a crucial role (Vargo and Lusch, 2004). In particular, S-D logic places the service—a process of doing something for another party—in its own right, without a reference to goods as the primary focus of economic exchange (Vargo and Lusch, 2006). The authors introduce fundamental principles of Service-Dominant logic. Ten foundational premises are proposed:

- The service, the application of operant resources (skills and knowledge) for the benefit of another party, is the fundamental basis of exchange
- Indirect exchange masks the fundamental nature of exchange
- Goods are a distribution mechanism for service provision
- Operant resources are the fundamental source of competitive advantage
- All economies are service economies
- The customer is always a co-creator of value
- The enterprise can not deliver value, but only offer value propositions
- A service centered view is inherently customer oriented and relational
- All economic and social actors are resource integrators
- Value is always uniquely and phenomenologically determined by the beneficiary

The declared principles are adopted to become the foundation of the theory of service—the *service science* (Spohrer et al, 2007, 2008; Lusch et al, 2008). In S-D logic, a *service* is the application of competence for the benefit of another party. From this basic definition, it is clear that each service involves at least two roles: at least one role possesses a competence and is able to apply it—the *provider* of a service, and at least one is willing to integrate external competence with available resources—the *customer* of a service.

These concepts are illustrated by some every-day examples. Firstly, regard a visit to a restaurant as a service interaction. A client orders a dish and, in collaboration with a waiter, clarifies how she wants a meal to be prepared. The restaurant applies its competence and the customers are willing to accept it.

Another example is a person getting optical glasses. A patient might use several service providers, e.g., one from a medical authority in order to obtain a medical prescription, another from an optical store to produce the glasses. There are several roles involved, each of which applies its competence for mutual benefit.

Finally, the research work performed during a collaborative project can be regarded as a service. A company has a concrete research problem, and a research group is asked to use its competences to solve the problem. In this case, again, competence is integrated into the resources of the company, solving its problem.

In most real-world scenarios, the competences of the parties participating in the service are far from trivial, assume complex interaction scenarios and, hence, cannot be easily decomposed into precise instructions. Furthermore, services that seem straightforward can be immensely complicated beneath the surface.

The key concepts used in the examples are now described in more detail.

- A *competence* identifies the ability of a service provider to apply knowledge and skills at a level of expertise sufficient for the accomplishment of a requested work specification by a customer in given settings. Customers in need of a competence enter markets to evaluate and pursue competence propositions presented by service providers.
- Upon an agreement between a customer and a provider a *service episode*, i.e., an occurrence of a service, takes place. During a service episode the provider provisions the service, often with the help of the customer and/or with access to the customer's resources.

- As an outcome of a service episode, the involved parties identify *value* resulting from the service. Value in business markets is the monetary worth of the technical, economic, service, and social benefits customers receive in exchange for the price they pay for a market offerings (Anderson et al, 2007). Therefore, a provider receives the value as a price for applying a competence, whereas a customer sees value in the application of an external competence and in results integration. In many cases, the value generated during a service episode can be measured quantitatively.
- In contrast to the physical assets common in goods dominated logic, such as equipment that eventually wears out and materials that are eventually depleted, the intangible assets common in a service provisioning environment may gain value with each additional use (Ricketts, 2008). As an outcome of a service episode, the acquired value for a service provider also includes the *experience* gained after a completion of a service.

Service episodes can be subjects of a quantitative performance measurement. In (Spohrer et al, 2008), the authors propose the ISPAR model for the qualification of service episodes. The model proposes the classification of ten possible outcomes for any particular service episode. Although the ISPAR model is a concrete model, it can be adopted to meet additional requirements. Once the ISPAR, or a similar classification of service episode outcomes, is accepted, it can be used to measure service performance over time. For instance, statistical methods can be employed to derive the qualitative signature of a service as a distribution of observed service episode outcomes.

In S-D logic, as well as in service science, goods play an important role. Services are either provided directly, or conveyed through a good. However, competences and skills are still the aspects creating value during service episodes. Goods result from services that are involved in manufacturing procedures and are used to provision services.

2.2 Properties of Service Systems

Service science initiative refers to participants of service episodes as service system entities. In (Spohrer et al, 2008), the authors give a precise definition: A *service system* is a dynamic value co-creation configuration of resources, including people, organizations, shared information (language, laws, measures, methods), and technology, all connected internally and externally to other service systems by value propositions. A service system, as an open system, is capable of improving the state of another system through sharing or applying its resources. A system sees an interaction with other systems as having value, and is capable of improving its own state by acquiring external resources, i.e., the system itself sees value in its interaction with other systems. The service systems that participate in a service episode willingly engage in cooperation upon mutual agreement. As the result of a service occurrence both systems are improved.

For our purpose and to our understanding, we summarize a *service system*, or a *system*, as a dynamic configuration of competence propositions. A *competence* *proposition* can either be of a *provider* role or of a *customer* role. A competence proposition of a provider role hints at the temporal ability of a service system to apply the competence, i.e., the service system has knowledge and skills sufficient to fulfill the competence it proposes to a market. A competence proposition of a customer role hints at the temporal desire of a service system to integrate external competence, i.e., the service system is looking for the competence in a market to derive and integrate potential value with its resources. A *service episode* takes place as a result of a match of provider and customer roles for the same competence proposition.

Diversity

Service systems are extremely diverse. The diversity partly arises from the fact that service systems are dynamic configurations of competence propositions. One can envision many combinatorial possibilities to compose competence propositions into one whole service system. Alternatively, the diversity is caused by the procedure of boundaries identification in service systems, which are often blurry. Can something be regarded as more or less a whole or as a whole for some purposes but not for others? Can a whole be also part of another whole or even of several other wholes (Vickers, 1983)? At what granularity is a whole acceptable for the identification as a service system? All of the proposed questions relate to the identification of an atomic service system.

Complexity

Atomic service systems can be combined to form *composite* service systems. For instance, one can envision hierarchical composite structuring or market based economic organizations of systems (Williamson, 1985). In part, the complexity also arises from the fact that a service system can simultaneously fill multiple roles in many service episodes with other service systems. Multitasking in the multiple roles carried out by service systems adds a further dimension to the combinatorial possibilities in overall service system diversity. Moreover, the mutual penetrations of the systems while engaging in service episodes aggravate the problem of identifying single service systems and results in intersections of service system configurations. Finally, one might address a sub-system by identifying the boundaries within the whole system (containment relation), e.g., a research group within an institute. However, every identified service system can be addressed by its unique identity as an instance of a type, or a family, of similar service systems.

Dynamism

Service systems compose or decompose over time with the main building blocks of service system compositions being competence propositions. Service systems, when together, constitute an extremely dynamic environment—a *market* of service systems. In this environment, every service system is struggling to gain value, which is a comparative concept. Customers assess the value proposition of a given market offering relative to what they regard as the next-best alternative to it. Every market proposes alternatives. The alternatives originate from: an offering from a competitor, the decision by a customer to source an item (to apply a competence) from another partner or to produce the item (to conduct a service episode) by themselves (to perform a self-service), the decision of not doing anything, the option of the most recent offering from the same partner (Anderson et al, 2007), etc.

Service systems change over time; they acquire new competences, give up on supporting economically unprofitable competences, enter new markets, leave declining markets, introduce innovations, etc. Service systems experience the need for external competences and propose competences to markets on the temporal basis following market trends. Service systems exist in time and, thus, have a beginning, a history, and an end. The history of a service system is a log of separate service episodes, conducted with sub-systems within its own configuration or with external partners, as well as a log of configuration snapshots of competence propositions over time.

Value Creation

The primary goal of each service system is to increase its value. A primary source for value increment are partner relations with market participants. A service system can increase its accumulated value in a given period of time by engaging in interactions with partners. The increase is expected even if the system outperforms every potential partner in every competence required by the system. The rationale behind this paradoxical statement can be explained by Ricardo's law of comparative advantage (Hardwick et al, 1999; O'Sullivan and Sheffrin, 2005). The principle behind the law proposes to service systems to concentrate more on their core competences and to outsource the competences that they do least well to partners. For this reason, the need for cooperation with other service systems has a strong motivation: The natural desire of a service system to increase the generated value.

Ricardo's law provides an instinctive reason for behavior of a service system. The law does not necessarily imply that a service system always looks for an external partner to outsource its secondary activities. A system might as well decide to perform a *self-service*. A self-service might occur in cases where a service system possesses both propositions of provider and customer roles for the same competence. Usually, a service system sees a solid economical profit when it decides to perform a self-service.

Service science initiative describes an innovative perspective on the environment of service provisioning. The core observation of service science is that in many cases, services such as professional, scientific, and technological services do not fit into the widely-accepted picture of repetitive and best-practice service specifications. Complex services may also need special tools and materials, but they often require sufficient levels of expertise (Ricketts, 2008). A service episode occurs upon mutual agreement between several partners and results in a transfer of a competence from service providers to customers. Finally, all partners that participate in a service episode see cooperation as having value. The additional value gained from the shared environment is the driving force bringing service systems in partner relations.

3 Flexible Process Graphs

In this section, we present flexible process graphs (FPG), a technique to represent business processes on a high level of abstraction. The formalism considers business processes as collections of activities with execution order constraints. Rather than detailing the control flow by edges between two activities, execution order constraints are defined in a much more flexible way, using subsets of activities. In particular, an activity can be performed once all its prerequisites are accomplished. This notion provides much more flexibility than existing control flow based approaches.

FPG were first introduced in (Polyvyanyy and Weske, 2008a) as a formal way for representing control flow in ad-hoc business processes. In (Polyvyanyy and Weske, 2008b), it is explained how FPG process instances can be parallelized for collaborative execution. At the core of FPG lies the generalization of a directed process graph edge which defines the sequential execution of two adjacent activities. The generalization of a graph is a hypergraph, as introduced in (Berge, 1985, 1989). Hypergraph edges (or hyperedges) consist of arbitrary sets of nodes. Thus, a hyperedge is an edge that can connect multiple activities. Different than in the graph-based sequence control flow pattern, a process participant is allowed to choose which activity to execute next within a hyperedge. This way a flexible execution of a process is achieved. A process model becomes hypergraph-structured, rather than graph-structured.

Definition 1. A *flexible process graph* (FPG) is a triple (A,E,T) where:

- A is a finite set of activity nodes
- E is a finite set of edges $e = \langle I(e), O(e) \rangle \in E, A \cap E = \emptyset$
  - $I : E \rightarrow \mathscr{P}(A)$ is a function defining edge input activities
  - $O : E \rightarrow \mathscr{P}(A) \backslash \emptyset$ is a function defining edge output activities
  - $\forall e \in E : I(e) \cap O(e) = \emptyset$
- T is an edge type function, $T : E \rightarrow \{and, xor, or\}$.

Each edge $e \in E$ is split in two subsets of input $I(e)$ activities and output $O(e)$ activities. Thus, the structure of an FPG is given by a directed hypergraph. Unlike regular graph-structured process models that contain special gateways to define control flow, FPG introduces edge types which implement routing decisions. The behavior of FPG processes is defined by the FPG execution semantics, which specifies state transition principles. At every point in time, an FPG process is in a certain state:

Definition 2. A *state* of a flexible process graph (A,E,T) is defined by a state function $S : A \rightarrow \mathbb{N}_0 \times \mathbb{N}_0$ mapping the set of activity nodes onto the pairs of natural numbers including zero.

In a state S, each activity node $a \in A$ is assigned a pair of numbers $S(a) = (i, j) \in \mathbb{N}_0 \times \mathbb{N}_0$. $S_\omega(a) = i$ (*white* tokens) specifies the number of instances of activity a that need to be accomplished from now on in the process instance. Respectively,

$S_\beta(a) = j$ (*black* tokens) specifies the number of activity instances accomplished in the process instance.

Process instantiation. Process instantiation is performed in two steps: $S(a)$ is set to $(0,0)$ for all $a \in A$. For each activity $a \in A$ the initial enabling is performed. An activity a is enabled at process start if $\varepsilon^*(a)$ holds:

$$\varepsilon^*(a) = \exists e \in E : a \in O(e) \wedge I(e) = \emptyset \wedge cond(e,a).$$

The *cond* predicate implements edge type $T(e)$ routing decision for edge e, e.g., $\forall a \in O(e) : cond(e,a) = true$, if $T(e) = and$. If $\varepsilon^*(a)$ holds, the process state S is modified to result in S', such that $S'(a) = S(a) + (1,0)$.

Activity firing. An activity $a \in A$ can fire in an FPG process instance if it is enabled ($S_\omega(a) > 0$). Activity firing results in the process state S change to S', such that $S'(a) = S(a) + (-1,1)$, i.e., one white token gets painted black.

Activity firing is instantaneous, consumes no time, and indicates a completion of the corresponding activity. After activity a has fired, the activity enabling has to be performed on a set of activities: $\bigcup_{\{e \in E | a \in I(e)\}} O(e)$. The enabling is performed for all the activities that are in the output sets of edges that contained the accomplished activity in the input set.

Activity enabling. An activity $a \in A$ can be enabled after execution of an activity a_β if $\varepsilon(a_\beta, a)$ holds:

$$\varepsilon(a_\beta, a) = \exists e \in E \forall a_i \in I(e) : a_\beta \in I(e) \wedge a \in O(e) \wedge S_\beta(a_i) \geq S_\beta(a_\beta) \wedge cond(e,a).$$

Enabling of activity a depends on execution of the preceding activity, e.g., activity a_β. An activity a can be enabled if there exists an edge $e \in E$, such that a is the output activity of e and a_β is the input activity of e. Further, for each input activity a_i of the edge e it holds that the number of accomplished instances of a_i is at least the number of accomplished instances of a_β. Finally, the edge type $t \in T$ condition for edge e must hold. If $\varepsilon(a_\beta, a)$ holds, the process state S is modified to result in state S', such that $S'(a) = S(a) + (1,0)$. Intuitively, new activities are available for execution in a process once all the prerequisites are accomplished.

Process termination. A process instance terminates when there is no activity to execute, i.e., no activity is enabled: $\forall a \in A : S_\omega(a) = 0$.

Process participants execute process activities following the proposed execution semantics to achieve a process goal. In business processes, activities can be fully automated by software systems, partially automated, or carried out by humans. For the sake of simplicity in FPGs we abstract from the diversity of process participant types and address them as *roles*. We identify each role as a sequential system, i.e., a role can only execute a single activity at a time. Therefore, true parallelism can only be achieved when several roles execute different activities at the same time. In order to coordinate efforts, each process participating role is assigned activities for execution.

Definition 3. A flexible process graph $FPG = (A,E,T)$ *role assignment* is a pair (R,W) where:

- R is a finite set of roles,
- $W : A \rightarrow \mathscr{P}(R)\backslash\emptyset$ is a roles assignment function.

Every activity in an FPG process has to be associated with at least one role, otherwise there might be no role responsible for accomplishment of an activity. Once enabled, an activity $a \in A$ can only be executed by a role from the assignment $r \in W(a)$. During FPG process instance execution, each participating role observes a subset of activities currently available for execution—the *role task list*. A participating role contributes to the achievement of the process goal by selecting and accomplishing an activity from the proposed list. The list is referred to as a role task list.

Definition 4. A *role task list* for the role $r \in R$ from the role assignment (R,W) for the flexible process graph $FPG = (A,E,T)$ is a function $L(r) = \{a \in A | r \in W(a) \wedge S_\omega(a) > 0\}$, where $r \in R$, defined on a subset of FPG activities ($L : R \rightarrow \mathscr{P}(A)$).

FPG is a simple formalism to specify allowed state transitions which describe process execution principles. A process is a collection of activity execution constraints. Each constraint allows the accomplishment of process activities only after all the designed prerequisites are fulfilled.

4 Formalization of Service Science Environments

In this section, we formalize the environment proposed by service science initiative. The structural as well as behavioral aspects of the service science concepts discussed in section 2 are proposed as a mapping onto the FPG formalism from section 3.

Informally, we understand a service science environment as a competitive environment, or a market, of service systems. Market participants engage in service episodes in order to apply their skills and knowledge, or to discover and to integrate the competence they need to fulfill their needs. In both of the cases, a service system expects to generate value. This highly dynamic environment, in which businesses join or leave markets, new markets appear, and strategic plans change, leads to the ad-hoc nature of service episodes. We propose to capture the state space of possible scenarios for occurrences of service episodes in the market as an ad-hoc process.

A service episode involves at least two service systems: one applying and one integrating the competence. Each service system can be seen as a collection of competence propositions. Furthermore, there is a clear distinction between the provider and customer roles of competence propositions. We formalize competence propositions by introducing a dedicated concept and a modeling construct. Figures 1(a) and 1(b) indicate the visual differentiation of service propositions of a provider role and, respectively, of a customer role, for a competence proposition a, e.g., a competence of conducting research. Service systems advertise their competence propositions to the market. In our example, a competence proposition a of a provider role

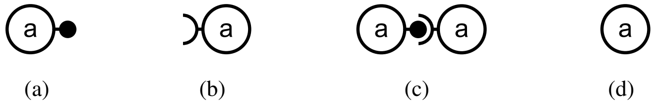

Fig. 1 (a) A competence proposition of the provider role, (b) a competence proposition of the customer role, (c) a competence match, (d) a service episode

means that a service system which advertises the competence in the market possesses sufficient skills in order to perform the research, e.g. a research group within an institute.

Conversely, a competence proposition a of a customer role signals to the market that a service system is advertising the need for an external competence, e.g., an enterprise that looks for innovations and is willing to finance a research.

A service system co-creates value with other service systems by engaging in service episodes. A service episode can occur as a result of a *competence match*, and is supplied within a *service episode*. There is a possibility for a competence match on the market if the market possesses competence propositions of both roles, provider and customer, for the same competence. Competence propositions might even belong to a single service system, but must be advertised in the same market. Figure 1(c) exemplifies competence match based on the competence of conducting research. By performing a competence match, a research group and an enterprise willingly engage in service related interactions. Figure 1(d) proposes a visualization approach for service episodes. The concept of service episodes aggregates information about a competence match and abstracts form the internal logic of the service provision.

Service systems are subject to constraints. If they had no constraints they could grow as large and as fast as they wanted without any restrictions (Ricketts, 2008). Constraints in a service system can be deduced from supported service episode scenarios, and therefore can be propagated to the competence propositions. In the following, we give a formal definition of a service system as an FPG composed of competence propositions, which also incorporates the constraints. We introduce an extension to the FPG formalism to allow differentiation between competence proposition roles.

Definition 5. A *service system*, or a *system*, is a configuration of competence propositions given by a quadruple (C,E,T,R_C), where:

- (C,E,T) is a configuration of the service system, given as an FPG composed of a finite set of competence propositions C
- $R_C : C \rightarrow \{provider, customer\}$ is a competence proposition role function.

Service systems are hardly useful in isolation. While a competence match might occur within a single system, it is far more likely to occur across service systems. Matching across service systems results in the service systems merging. A service science environment is obtained by merging service systems based on their competence propositions. Under a service science environment we understand a temporal co-existence of service systems which are in competitor or partner relations.

Once competence propositions are matched, service episodes might happen. A service episode is enabled for execution if it was obtained as a result of a competence match and is enabled in all participating service systems, assuming the underlying FPG enabling semantics (see section 3). An enabled service episode can occur. An occurrence of a service episode results in a competence transfer from providers to customers. A successful service episode completion signals for an environment state change. The corresponding competence propositions fire following the FPG execution semantics in all participating service systems.

In general, the structure of an FPG is fixed. However, in the case of service systems, we foresee the necessity for structural changes during system lifecycles. A service system is expected to change its structure as a response to market trends. In this case, the restriction of the fixed FPG structure must be waived. Service systems might decide to introduce new constraints or to give up on old ones in order to pursue market trends.

Service systems can rely on FPG role assignment capabilities to distribute operand resources (available products that support service episodes) and operant resources (people or machines producing an effect of competence application) among service propositions. Execution of a service episode can be parallelized following FPG principles (Polyvyanyy and Weske, 2008b) and monitored with the help of FPG task lists. The history of a service system can be tracked by logging FPG firings (production of black tokens), as well as by monitoring structural changes of the system. In order to allow quantitative service system evaluation, each token can be enhanced to carry service episode outcome information, like the one proposed within the ISPAR classification. The quantitative signature of a service proposition can further be used as a notion of the experience of a service system in delivering or consuming the corresponding competence.

5 Example of a Service Science Environment

In this section, we present an example of a service science environment obtained by merging several service systems. The example illustrates concepts and principles of the service science initiative.

The example scenario is a joint research project that involves three institutions. The scenario specifies an episode from a simplified and anonymous version of a real-world research project. The settings assume a transfer of a research competence from a research group to an enterprise through the application of developed mechanisms to the process model repositories of the enterprise.

In this setting, the enterprise is willing to use the competencies of the research group for process analysis and transformation. The enterprise is willing to integrate the research results in the company (by adding the new process models to the process model repository of the company). The research group provides the competence which is necessary to perform the requested analysis and transformations. If both institutions decide to partner for mutual benefit they require external assistance to settle legal issues in a project contract.

Each of the three proposed institutions, the research group, the enterprise, and the legal authority, is an example of a service system type. In order to partner, the instances of the mentioned service system types need to be present at the same market simultaneously.

In the remainder of this section, we formalize one instance of each service system type mentioned. Afterwards, we exemplify the merging procedure of service systems and discuss the behavior of such temporal phenomena. For the sake of simplicity, we specify service systems to a level sufficient for scenario coverage. Throughout the examples we visualize the formalism by following the proposal for graphical representation of flexible process graphs suggested in (Polyvyanyy and Weske, 2008a): Edges in flexible process graphs are represented by regions. Input edge nodes are located on the borderline of the corresponding region, whereas output edge nodes are placed inside the edge region. We employ the visual notation from Figure 1 to differentiate between the roles of competence propositions and to represent service episodes.

Next, we discuss one concrete example of a service system for each of the following types: a research group, an enterprise, and a legal authority.

Research Group

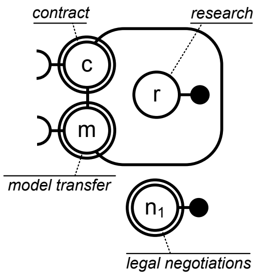

Fig. 2 A research group system

A research group is an institution with expertise in a certain domain of science and which pursues challenges and innovations in order to contribute to the overall body of knowledge. In our simplistic example, we treat a research group as the decomposition of four competence propositions: *c*—preparation of the project contract, n_1—contract legal issues negotiation with authorities, *m*—process model repository transfer, and finally *r*—the research undertaken.

In our example, the research group has expertise in business process management. It is interested in obtaining real-world data—process model repositories. To do that, a project contract which negotiates work packages needs to be developed. To derive a contract, a research group supplies its legal regulations which have to be obeyed.

A service system of a research group is shown in Figure 2. It can be formalized as a configuration of competence propositions: (C,E,T,R_C). $C = \{c,m,r,n_1\}$, $E = \{e_1,e_2,e_3,e_4\}$, where $e_1 = \langle\emptyset,\{c\}\rangle$, $e_2 = \langle\emptyset,\{m\}\rangle$, $e_3 = \langle\emptyset,\{n_1\}\rangle$, and $e_4 = \langle\{c,m\},\{r\}\rangle$. All of the edges are of *and* type ($\forall e \in E : T(e) = and$), $R_C(c) = R_C(m) = customer$, $R_C(r) = R_C(n_1) = provider$.

Internal constraints of the research group are enforced by edges which describe the structure of the service system. It requests external competences to engage into service episodes for negotiating the project contract and obtaining process model repositories. These requests are represented by the competence propositions c and m, both of the customer role.

The outer line around competence propositions, e.g., c and m, stands for an FPG edge which contains only the competence proposition corresponding to the output node—there are no prerequisites for the proposition. If the service system succeeds in obtaining the value from service episodes that involve competence propositions c and m, it can proceed with supplying the competence of research (r). At any time, the research group can engage into a service episode of legal negotiation for a project contract (n_1), as no prerequisites are modeled.

Enterprise

For the purpose of our example, we see an enterprise as a decomposition of four competence propositions: c, m, and r are similar as in the case with the service system of the research group, and n_2—contract legal issues negotiation with authorities specific to an enterprise. An enterprise is looking to obtain research results. For this purpose, an enterprise is ready to provide its process model repositories. Upon request, an enterprise is ready at any moment to negotiate legal issues and to set up a contract.

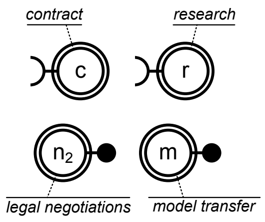

Fig. 3 An enterprise system

A service system of an enterprise is given in Figure 3 and can be described by a configuration of competence propositions (C,E,T,R_C). $C = \{c,m,r,n_2\}$, $E = \{e_1,e_2,e_3,e_4\}$, where $e_1 = \langle \emptyset,\{c\}\rangle$, $e_2 = \langle \emptyset,\{m\}\rangle$, $e_3 = \langle \emptyset,\{r\}\rangle$, $e_4 = \langle \emptyset,\{n_2\}\rangle$. All of the edges are of *and* type ($\forall e \in E : T(e) = and$), $R_C(c) = R_C(r) = customer$, $R_C(m) = R_C(n_2) = provider$.

To simplify the example, we abstract from complex internal enterprise logic and assume all the competence propositions to be enabled. Both n_1 from Figure 2 and n_2 from Figure 3 are competence propositions of the same type—legal issues negotiation. Please note that the fact that all competence propositions are enabled does not imply that a service system has no constraints. An enterprise is constrained to be able to only participate in service episodes that involve competence propositions c, r, m, and n_2.

Legal Authority

An institute of a legal authority is included in our example to have a system which is capable of delivering a competence of setting up a contract. A legal authority is capable of conducting legal negotiations with each of the contractors and, afterwards, to issue a legal document which regulates partner relations—a contract.

For our purposes, it is sufficient to see a legal authority as a system consisting of three competence propositions: n_1 and n_2 are both competence propositions of the type legal issues negotiation, and c—similar as proposed above, preparation of a project contract.

A service system of a legal authority is visualized in Figure 4 and is a configuration of competence propositions (C,E,T,R_C). $C = \{n_1,n_2,c\}$, $E = \{e_1,e_2,e_3\}$, $e_1 = \langle\emptyset,\{n_1\}\rangle$, $e_2 = \langle\emptyset,\{n_2\}\rangle$, $e_3 = \langle\{n_1,n_2\},\{c\}\rangle$. All of the edges are of *and* type ($\forall e \in E : T(e) = and$), $R_C(n_1) = R_C(n_2) = customer$, $R_C(c) = provider$.

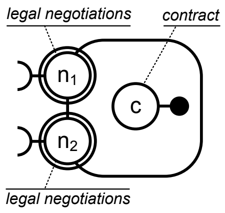

Fig. 4 A legal authority system

A service system of legal authority has a constraint which states that it is capable of delivering the competence of setting up a contract (c) only after it has finalized service episodes of legal issue negotiations (n_1 and n_2) with each of the partners.

Shared Environment

Operational principles of service systems are governed by FPG execution semantics. State transitions result from the successful completion of service episodes, implying prior competence match. Service systems from Figures 2, 3, and 4 are scarcely useful in isolation. Standalone systems only describe their own constraints, i.e., the way they do their business. The real value comes from service system compositions. In the following, we discuss two examples of service system environments formed by merging service systems.

Figure 5(a) shows a potential shared environment of a research group and an enterprise. The merging results in two competence matches based on the competences of process model repository transfer (m) and research (r). The research group imposes the constraint to the overall environment—a contract must be settled (c) and models transferred (m) before research (r) can take place.

In the environment, a service episode m might happen; it is obtained as a result of a competence match and all competence propositions that participate in a match are enabled within the corresponding service systems (see Figures 2 and 3). A completion of service episode m results in a firing of corresponding competence propositions in the participating service systems. However, a service episode of research (r) cannot occur after this. Although it is enabled within the enterprise, the research group requires a contract (c) before it enables r. The environment proposed in Figure 5(a) does not have a competence match for c. Therefore, the partners may start with model transfer, but still require external competence to assist with contract preparation in order to proceed with research.

Figure 5(b) completes the composition of the environment by additionally merging the service system of a legal authority. Now, all the competence propositions are matched and can occur. By participating in the service episodes, the participants of the environment start to collectively approach realization of their goals: First, ser-

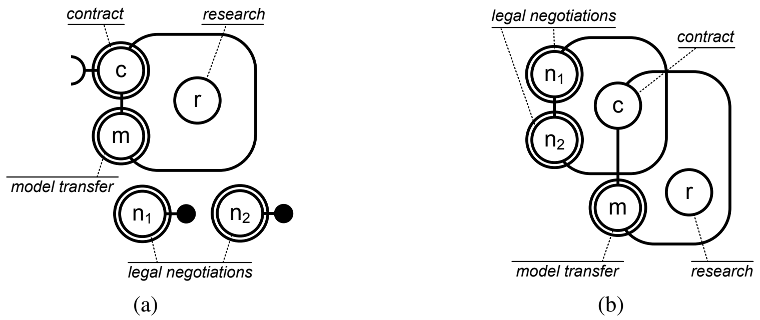

Fig. 5 (a) An environment of a research group and an enterprise, (b) An environment of a research group, an enterprise, and a legal authority

vice episodes of legal negotiations (n_1, n_2) and model transfer (m) are enabled. Once negotiations are finalized, the project contract can be settled (c). Once the contract is ready and the models are transferred, the partners can proceed with research (r).

Each service episode within the project can be addressed as a complex interaction that in the end delivers value to the participants. For example, model transfer (m) can involve complex interaction on shaping, correcting, finalizing, or enhancing models. A service episode of research (r) is aimed at delivering desired results to the enterprise, but may also result in new findings and methods for the research group. Service episodes n_1 and n_2 are the examples of knowledge transfer services. The service episode of setting up a contract (c) is obtained as a result of merging of all three participants with one provider role and two customer role competence propositions; it can represent a joint meeting.

In our example we have performed all competence proposition matches possible. However, in a general case, a service system should decide on the desired configuration of competence proposition matches once it enters an environment.

6 Conclusion

In this chapter, we made the first steps towards formalization of service science environments. Many open questions are still to be answered, and many issues are still to be concretized with our approach, as well as with S-D logic and service science initiative. As for the state of the art, we proposed a modeling technique to capture service systems and their ecology. Service systems are addressed as FPG configurations of service propositions. Models of service systems can then be used for execution, analysis, optimization, or redesign of service systems.

A service science environment is obtained once several service systems decide to merge. The merging is guided by matching competence propositions and results in resource integration. Such behavior of service systems is explained by the desire to achieve a synergy effect, i.e., the key to success is not to destroy but to enhance your partners. After the competence match is reached and is enabled within the environment following FPG execution semantics, the service episode might happen. Service systems split, once there is no need in further partnership.

The future work in service science formalization initiative will have to deal with a better understanding of a service system merging/splitting behavior. In this context, it is a challenging task to understand how competences are defined and brokered. Also, it is interesting to answer how a single service episode can be modeled if it is assumed to be a complex interaction scenario between service partners. All of the above mentioned initiatives can lead to a better understanding of the Moore's law for service system continuous improvement.

References

Anderson J.C., Kumar N. and Narus J.A. (2007) Value Merchants: Demonstrating and Documenting Superior Value in Business Markets. Harvard Business School Press, Boston, MA.

Berge C. (1985) Graphs and Hypergraphs. Elsevier Science Ltd.

Berge C. (1989) Hypergraphs: Combinatorics of Finite Sets. Elsevier Science Ltd.

Burbeck S. (2000) The Tao of e-business services: The evolution of Web applications into service-oriented components with Web services. http://www.ibm.com/developerworks/library/ws-tao/. Accessed November 2009.

Gottschalk K. (2000) Web services architecture overview: The next stage of evolution for e-business. http://www.ibm.com/developerworks/library/w-ovr/. Accessed November 2009.

Hardwick P., Khan B. and Langmead J. (1999) An Introduction to Modern Economics, 5th edn. Financial Times/Prentice Hall.

Lusch R.F., Vargo S.L. and Wessels G. (2008) Toward a conceptual foundation for service science: Contributions from service-dominant logic. IBM Systems Journal 47(1):5–13.

Newcomer E. and Lomow G. (2004) Understanding SOA with Web Services. Addison-Wesley.

O'Sullivan A. and Sheffrin S. (2005) Economics: Principles and Tools, 4th edn. Prentice Hall.

Polyvyanyy A. and Weske M. (2008a) Hypergraph-based modeling of ad-hoc business processes. In: Business Process Management Workshops, Springer Verlag, LNBIP, vol. 17, pp. 278–289.

Polyvyanyy A. and Weske M. (2008b) Flexible process graph: A prologue. In: Proceedings of the 16th International Conference on Cooperative Information Systems (CoopIS), Springer Verlag, LNCS, vol. 5331, pp. 427–435.

Ricketts J.A. (2008) Reaching the Goal: How Managers Improve a Services Business Using Goldratt's Theory of Constraints. IBM Press.

Silberschatz A., Korth H.F. and Sudarshan S. (1997) Database System Concepts, 3rd edn. McGraw-Hill Book Company.

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

Spohrer J., Vargo S.L., Caswell N. and Maglio P.P. (2008) The service system is the basic abstraction of service science. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS), IEEE Computer Society.

Vargo S.L. and Lusch R.F. (2004) Evolving to a new dominant logic for marketing. Journal of Marketing 68:1–17.

Vargo S.L. and Lusch R.F. (2006) The Service-Dominant Logic of Marketing: Dialog, Debate, and Directions, ”Service-Dominant Logic: What It Is, What It Is Not, What It Might Be”. M.E. Sharpe.

Vickers G. (1983) Human Systems are Different. Harper & Row.

Weske M. (2007) Business Process Management: Concepts, Languages, Architectures. Springer Verlag.

Williamson O.E. (1985) The Economic Institutions of Capitalism. The Free Press.