14th CAiSE. Springer Lecture Notes in Computer Science, Berlin, pp. 500-515, 2002

Serviceflow Beyond Workflow? Concepts and Architectures for Supporting Inter-Organizational Service Processes

Ingrid Wetzel, Ralf Klischewski

Hamburg University, Department for Informatics
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
{wetzel,klischewski}@informatik.uni-hamburg.de

Abstract. With Serviceflow Management we put the service nature of inter-organizational processes into the center of modeling, design and architectures. The underlying conceptual distinction between the serviceflow, the portion of the process where the customer's concern is evaluated and cared for, and background processes, guides in (1) providers as well as designers to focus on service design and delivery, (2) to provide support for serviceflows with enhanced flexibility and service configuration and (3) to design service points where service workers and customers "meet". By this, the original workflow metaphor, which directs the design of process support from a mass production point of view, is questioned and replaced by a more suitable concept, which considers social and quality aspects in service delivery. Instance-based XML process representations and generic components and architectures for their exchange and for the provision of service tasks are presented, discussed, and exemplified by an e-health process.

1 Introduction

As different authors [11] [19] suggest, we are entering an advanced phase of Internet usage. The internet having become a key medium for global marketing in 1995 (known as brochureware), e-commerce initiatives started from 1997 upwards, e-business projects boomed among business partners since 1999 and we are now in the phase of e-enterprises [12], or virtual organizing [4], or business networking [21]. The overall idea is to apply *process orientation*, implemented in the 90s within organizations, to inter-organizational processes [21]. Furthermore, the aim is to create added-value by *converging e-commerce*, i.e. interacting with the customer in single transactions, and *customer relationship management*, i.e. comprehensive customer care over time, and *supply chain management*, i.e. in-time ordering of supplies, in order to provide an overall *comprehensive service*. Intended applications are, e.g. process portals as successors of web-portals, through which services and goods, often supplied by different providers, are offered by a single provider (the service integrator) in a personalized and configurable way. Moreover, and this is the special focus of this paper, the customer should be guided through the corresponding inter-organizational service process. This kind of system support has to overcome the still existing product-oriented manner in service delivery as it is visible e.g. in buying real estate [21], planning a move [5], undergoing an operation [17]. Efforts should be made to relieve the customer from being forced to become an expert in the process (over time) and being involved in coordination work.

Furthermore, support has to offer a configurable and flexible service, where the customer concern may *change* processes over time and the customer's *satisfaction* is determined by a multitude of factors [7]. Quality in service delivery, as we will see, will certainly go *beyond* mere *efficiency* and requires new approaches, which *go beyond familiar ways of systems design*. We argue accordingly.

Section 2 is centered around characteristics of inter-organizational service processes, illustrated by means of a healthcare example, and it discusses the limitations of available technology in the light of the given demands. Section 3 presents the overall concept and modeling approach of Serviceflow Management. Section 4 describes the use of different stages of realization among provider organizations, from XML-process representations to the provision of different generic components, to the configuration of different network architectures. After presenting Serviceflow Management, section 5 discusses whether we meet our claim to go beyond workflow or not. In doing this, we compare our approach with others found in the literature. Finally, section 6 will summarize the main results.

2 Characteristics of Inter-Organizational Service Processes and Limits in Available Technology

Taking a typical example from the healthcare domain for illustration – the description of similar cases are found in [17] [3] – we consider the process of the preparation for, performance and aftercare of an inpatient surgical operation [13].

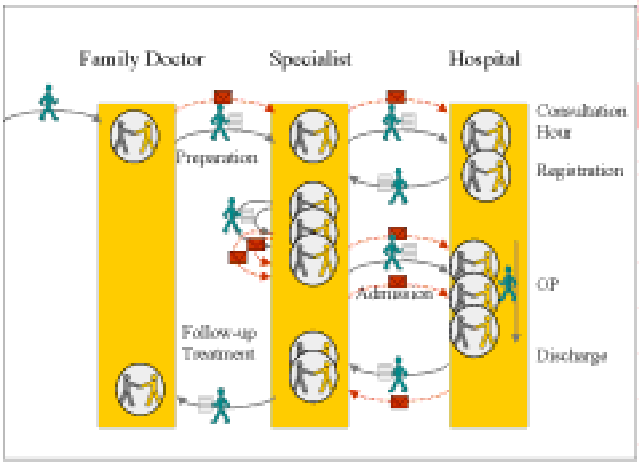

Figure 1: Service example: preparation, performance and aftercare of an operation

Based on our analysis in cooperation with a German clinic specialized in bone surgery and endoprothetic, the presented scenario describes a standard procedure for hip replacements. In this process, the patient usually moves back and forth between different physicians/specialists and a clinic to receive an in-depth diagnosis as well as appropriate medical care and treatment. A patient typically starts with consulting a family doctor, is directed to a specialist, chooses a hospital, goes through consultation and registration at the hospital with a schedule for further preparation, passes through all stages of preparation, stays in the hospital where the operation is performed, all of which is followed by aftercare treatment at specialists (see figure 1). In the course of this process, various documents have to be exchanged, some of which are delivered by the patient while others are sent by mail or fax.

The performance of this sample composite service exhibits a number of characteristics.

- *Concerning the process:*
  - *Process Responsibility and Flexibility.* There is a lack of overall responsibility for either the process or its planning. Rather, the process seems to rely on the ability of individual providers to flexibly shape the service according to their special insights about the patient's case (which is why they are called in). Furthermore, the patient's status and hence his concern, continually influence the provider's actions/services over time.
  - *Rules about Conducting the Service.* A further problem relates to the exchange of documents. At present, there are no clear rules. It is neither obvious in which way documents are to be delivered nor which kind of documents should be exchanged at all.
  - *Monitoring Process Status.* Providers involved do not receive a complete picture of the process, its current status or development. They often lack information about deviations from tacitly assumed ways to proceed, or are unable to obtain this knowledge.
- *Concerning the provider network*
  - The network exhibits a completely heterogeneous technical infrastructure, with missing standards in software and hardware equipment, at some sites paired with poor motivation to use computers at all as is still perceptible among health providers.
  - There is a substantial lack of motivation to invest in improved cooperation as articulation work is not rewarded financially at present (at least in the German health system) and is expected to be compensated in future in a probably unfortunately biased form.
- *Concerning the customer and his/her concern*
  - As patients hopefully undergo such operations only a very few times throughout their lives, they aren't knowledgeable about the process itself, even though some of the coordination tasks are left to them. However, over time, they learn about the process themselves in order to intercept some of the lack of coordination.
  - The patient usually expects the networking providers to adjust their actions according to actual findings, needs and changes as they evolve over time. Consequently, a high quality service can only be delivered if the different providers act in a coherent and constantly adapting way.

- The success of a delivered service can sometimes be evaluated instantly but more often only with a certain time-lag to the actual treatment.

These characteristics result in manifold requirements regarding the software support for inter-organizational services. They certainly depend on the kind of service process in question, as is clearly discussed in [1], and on the focus of support (e.g. [5][6][8][18]). To simplify matters, we structure the requirements in three dimensions (see figure 2):

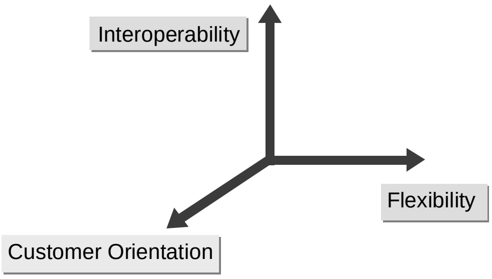

Figure 2: Requirements for the support of inter-organizational processes

- *Flexibility Support*. The process underlying the comprehensive service delivery is a main object to be supported. Support has to provide flexibility and dynamic selection and configuration of services as the process moves on. Furthermore, the steps of the process need to be transparent to the parties involved including the customer in order to support adjustment and service coherence.
- *Interoperability*. The underlying software systems of providers need to be able to communicate. This includes the exchange of information, may include the remote execution of software components or the sharing of (meta-) knowledge (e.g. about service templates) or services (including cooperation support tools).
- *Customer Orientation*. Service delivery has to consider the possibly changing customer concern and the subtleties in customer satisfaction. This goes beyond efficiency criteria of underlying processes and has to do with understanding the customer concern and with service customization, with the nurturing of personal relationships [9], the mutual adjustment of subservices, possibly the ubiquitous availability of services, and many more [7].

Currently, widespread approaches and tools only match these requirements in a limited way.

Regarding *workflow management,* the overall idea of executable process models comes to its limits. Being based on a central workflow engine that controls a case's execution according to a predefined process, it does not comply with the required flexibility. Although being criticized for the strictness of their execution over the years, it is only today that their former proponents admit this clearly by speaking about the ill-suitedness of conventional "bulk-based" workflow management with regard to ad-hoc flexibility [2]. Likewise, the centralized workflow engine is not suitable with respect to *interoperability* [1].

Internet technology, on the other hand, still lacks supporting processes. However, it makes major contributions towards interoperability. Whether it is the ubiquitous cost-effective network with its standard http-protocol or recently established standards such as XML and, based on it, SOAP or WSDL, these techniques allow for an exchange of standardized/typed messages, also in remote method calls.

As a consequence, recent efforts have been based on combining the two technologies. The respective approaches are called inter-organizational workflow, or composite e-services, e.g. [1][5], see also section 5.

3 The Serviceflow Concept

Accordingly, Serviceflow Management exploits both, workflow management and Internet technology, while adding further customer-related aspects. The overall idea is to

- use case-based XML process representations to allow for flexibility and adaptability as a basis for implementation concepts,
- supply middleware components for delivery and handling of these representations to achieve interoperability,
- conceptually distinguish customer-near parts of the service process from background processes in order to guarantee improved customer orientation and care.

In particular, a serviceflow is defined in terms of service points. A service always creates some social situation, it needs “places” [11] which frame the situation where service tasks are carried out in an individual way. These places we call service points, and the successive interrelation of a number of service points is a serviceflow.

However, from the service provider’s point of view, the challenge is to look for recurrent serviceflow patterns. In order to define these patterns, both the sequence of service points and the service at each service point have to be modeled. The sequence of service points for our example is shown in figure 3. Each service point captures specific service tasks to be carried out (forming a very flexible sort of subprocess) and their respective pre- and postconditions from the provider’s point of view. The pre- and postconditions represent the contract for interrelating the service points. Service tasks are modeled as UML use cases with each use case being further linked to a rich description, a scenario, and a use case picture. Cooperation pictures can augment the serviceflow representation to further illustrate cooperation among the actors involved, for more detail see [15]. Background processes may be modeled or not. As long as they are captured in some sort of interface description, as pointed out in [1], these interface specifications may suffice as a basis of agreement. In our approach, the formulation of pre- and postconditions serves for this purpose.

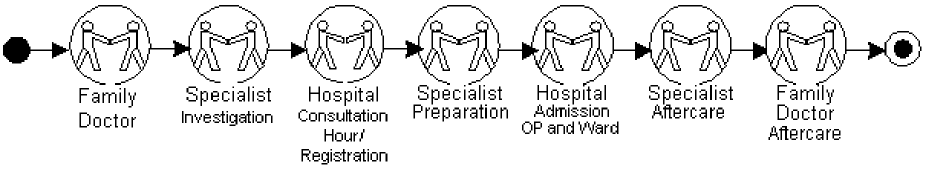

Figure 3: Serviceflow model for the healthcare service example (s. section 2)

Thus, we combine social “places” and individualized serviceflows for customer concern and care with serviceflow patterns in order to support the efficient organization of mass-service delivery between providers. The overall concept for serviceflow management is now centered around the technical representations of the modeled process patterns that lead to the notion of service float and service point script. Service floats are sent from service point to service point and capture personalized, always up-to-date process knowledge, whereas service point scripts support and document the standard and adaptable activities at each service point (see figure 4).

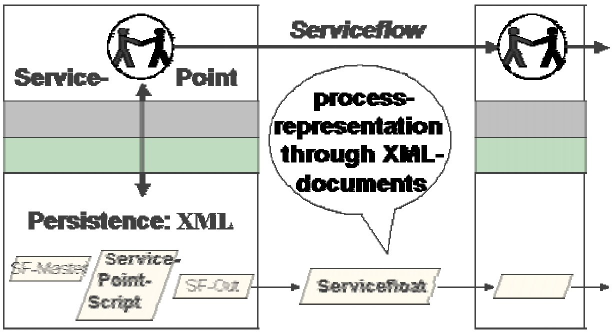

Figure 4: Service floats and service point scripts

Overall, the approach bears the following potential:

- Initializing a service float by copying (and possibly adapting) a standard serviceflow pattern guides each provider as to how to deliver the service.
- Enabling providers to access and update the process representations (as material) allows for flexibility and instant realization of changes.
- Documenting “the history” enables a service provider to be informed about deviations from the standard and their reasons.
- The update of the current and setting of the next service points forms a basis for automating the delivery of service floats to the next provider.
- The design of service points with flexible support of customer interaction and service delivery.

Note: Serviceflow Management requires agreements on the content of the modeled serviceflow pattern and on the handling of these representations during exchange.

4 Realization of Serviceflow Management

In the following, we describe the realization of serviceflow management in four stages. We start with a basic form of interoperability support which can be achieved by specifying XML-based process representations that are exchanged on a case-basis. Further stages will provide generic components in order to support the exchange of process information, to perform service tasks at service points and to provide suitable interfaces. The components are designed in such a way that they are configurable into decentralized, half-centralized and centralized architectures.

4.1 XML Representation

Basic interoperability can be achieved by exchanging process representations between providers on the condition that providers are able to interpret these representations in a uniform manner and guarantee certain rules for their handling. Following major improvements in standardization, we represent serviceflows technically as XML documents, called service floats.

In detail, a service float's XML representation contains the following elements: an identifier for the individual serviceflow, basic information on the serviceflow client, the current service point (service points are described by identifier, name, type, provider, address), a list of scheduled service points, a list of service points passed, a list of accumulated postconditions, a list of documents, i.e. short message texts or references to full documents or document folders.

At each service point, the service float is evaluated according to the respective XML service point script that prescribes the activities at the 'current service point', which is an identifier for the individual service point, basic information on the service point provider, current activity (activities are described by identifier, name, type, task), a list of scheduled activities, a list of passed activities, a list of preconditions for the set of activities at this service point, and a list of documents.

To use XML-based service floats for interoperability requires from

- the provider network
  - to distribute the XML-DTD/schema for service floats and service point scripts and other shared data structures (e.g. forms, patient data, patient record),
  - to agree upon XML "master"-documents for service floats and service point scripts according to different serviceflow types,
- each provider
  - to meet a set of rules on how to manipulate and share those XML documents
  - to provide a technical infrastructure for enabling the rules.

Summing up, this "stage 0" realization is based on process representations and methods for their handling. Although it enables interoperability between heterogeneous systems of different organizations it leaves open how each provider enables the actual system support. (This proved to be successful in providing the postal vote application service through www.hamburg.de [14][16]).

4.2 Component-Based Architectures

Beyond basic interoperability, the serviceflow management approach aims at providing suitable components which may be distributed to different sites in order to ease the effort of joining networks. We distinguish three stages which provide additional components for handling the service float, the servicepointscript and the interaction (s. figure 5).

- Stage 1 offers a component which enables the swap-in and swap-out of service floats including the routing and the update.
- Stage 2 provides additional support for processing the tasks at each servicepoint (s. below).
- Stage 3 builds on the underlying stages and provides a web-interface for service providers or customers to be integrated in a process portal.

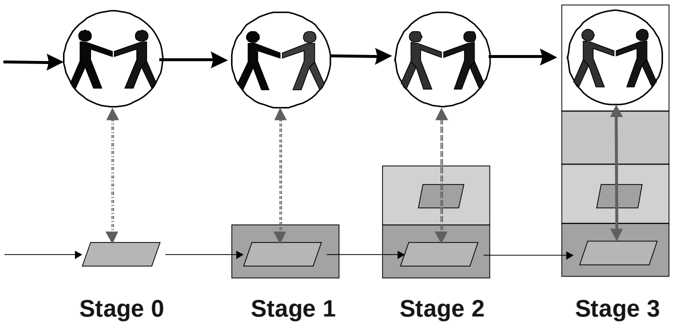

Figure 5: Stages 0 to 3 in serviceflow management support

Note that the same components can be configured in different architectures ranging from a decentralized, to a partly centralized, to a fully centralized architecture as discussed in the following.

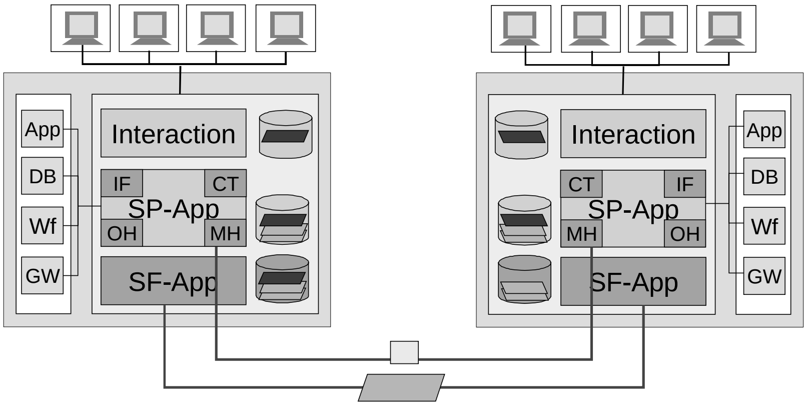

Figure 6: A decentralized architecture for serviceflow management

The *decentralized architecture* is given in figure 6 and shows two organizations both of which apply to stage 3. The exchange of service floats is accomplished by the service float application (SF-App) at each site, which apart from the routing support offers methods to update and query a service float. Furthermore, it comprises a storage component where the service float masters are stored (at the starting node in the network) and each incoming and outgoing service float can be documented for statistical reasons or for error handling. In this sense, the architecture is completely decentralized.

The service point application (SP-App) captures at least four subcomponents, (1) a message handler (MH), which receives incoming or delivers outgoing messages carrying information, e.g. about the completion of a background process or the availability of further information, (2) a condition trigger (CT), which compares messages with preconditions of a service point and informs workers about a changed status if necessary, (3) the object handler (OH), which manages documents or other objects related to the case, and (4) the integration facilitator (IF), which serves to integrate the service point tasks with applications (App) available in the organization, such as legacy systems, data bases (DB), workflow (Wf), or groupware systems (GW). Similarly to the service float application, the service point application comprises a database component for storing service point script masters and documenting their execution.

Finally, the interaction component serves for providing html-based client interfaces and can be structured on the basis of standard templates.

A decentralized architecture suffices for processes with "chained execution" [1], where each of the tasks at one provider's site is completed before the next provider is in charge, i.e. where no parallel execution is required. Now consider our example case. A patient may be part of two serviceflows in parallel, in which case different providers may share documents across serviceflows. Likewise, a specialist may need to add a delayed result of an examination to a service float which has already been sent to the next provider. Here, a partially centralized architecture may be more suitable in order to provide the required infrastructure.

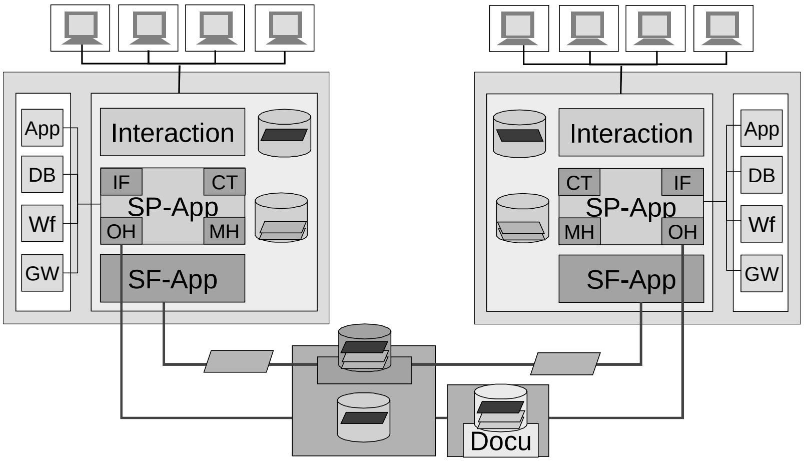

Figure 7: A partly centralized architecture for serviceflow management

Figure 7 indicates a partially centralized architecture with a central server for documents, assuming that the documents are no longer part of the service floats. In this case they will be sharable among different serviceflows and updateable after a service point has ceased to be active (in which case an additional message has to be sent in order to provide awareness information to the next provider or the whole network, signalizing that additional information is available and should be considered immediately).

In this architecture, a second central server can be installed. It may provide a centralized routing support, which eases the serviceflow application in that each provider must send service floats to this centralized server only and routing information is kept in one hand. Additionally, service float masters can be stored at this server. Furthermore, the centralized documentation of the in- and out-serviceflows at each provider's site allows for comprehensive queries regarding the actual state of a particular flow, statistical evaluation, or error handling. Finally, the centralized storage of service point script masters provides the means for standardization of service delivery at service points across different providers in the same network.

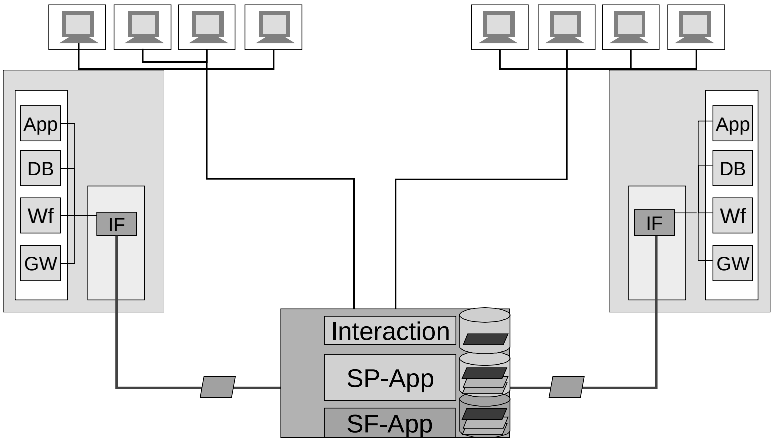

Figure 8: A centralized architecture for serviceflow management

A further step is made if all architectural serviceflow components are united at one central provider/server leading to a fully centralized architecture, possibly managed by an application service provider in charge. This scenario, as indicated in figure 8, has the advantage of providing direct Web connection to clients/service points for service workers and the customers themselves without requiring any extra software equipment at the provider's site. However, the main difficulty with such a scenario lies in the integration effort "behind" the central component. As discussed in [23] the handling might be very complex. Still, we depart from their approach in that we assign (again) an integration facilitator within each provider organization for hiding details of server connections. This allows for a further level of abstraction which is necessary both because changes in a provider's infrastructure should not jeopardize the functioning of the provider network and because dynamic joining in and out of the provider network should be as easy as possible.

5 Discussion: Serviceflow beyond Workflow

In the following, we compare our approach with other approaches and e-platforms in order to underline and summarize its specifics along the three dimensions given in section 2.

5.1 Flexibility and Interoperability

Like other recent approaches [17] [3] [23] which aim at combining workflow and internet technology in order to support interorganizational processes, we use XML based process representations that are exchanged among providers. As we have seen, this allows for (1) interoperability by using XML standard and Internet-based communication and (2) flexibility of service delivery on the basis of *instance-based* process "execution".

Further, we have to mention that the underlying extensible routing language XRL [3] is in some aspects more sophisticated than our routing language as it provides XML elements for each workflow modeling construct (and thus concurrent branches). Furthermore, by defining transformations from each construct into petri-net representations it even provides a formal semantics. On the other hand, it may not offer as much attributes as our approach in order to capture the history of a process instance.

However, an interesting difference arises at the "entering point" of process descriptions into organizations. Within organizations it is assumed in [3] that the routing slip gets transformed into a Petri-net which then directs the execution of workflow systems. It is at this point – with regard to the inner-organizational processes – that flexibility seems to be no longer guaranteed. Whereas these approaches up to now fail to explain how the required overall flexibility may be accomplished if the organization-internal work is accomplished with inflexible workflow systems, serviceflow management differs inasmuch as it applies the same technology – XML-based process representations – at service points. This allows to provide flexible support *also* or *mainly* within the organizations at the servicepoints where customers and service workers meet and thus individual customer care takes place.

At the same time, applying XML-based process representations at service points bears the potential of standardizing these subprocesses either within the organization or across service providers. The latter also contributes to the dynamic configuration and contracting of service processes which, on the basis of these templates, eases switching in of providers into the network and enables the implementation of future service and process brokers. Admittedly, this requires a quite sophisticated service point application, which is still under development.

Brief mention should be made of further work on e-service platforms for managing the dynamic configuration of e-services in inter-organization processes, which addresses a wide spectrum of support. Aspects taken are e.g. the provision of templates [6], the distinction between service and method nodes, and the introduction of migration support [5], the support of matchmaking and contract completion based on service monitoring, flexible change control, and transaction models [8], and the support of catalogues, guarantees, measurements and multimedia cooperation [18].

Apart from the correlation discussed above to other current approaches which already go beyond typical workflow management in allowing ad-hoc flexibility and interoperability, we want to underline another general difference from (former) workflow approaches we perceive. This relates to the exchange of service history (as captured by service floats) among providers and potential consequences.

For this let us briefly consider typical execution environments in workflow management systems. Originating in the area of mass production, these approaches saw an advantage in equipping a worker with only the one particular bit of the overall process knowledge which seemed sufficient for him/her to perform the specialized task at hand. The traditional division in analysis and production (Taylorism) made analysts responsible for knowing and (re)designing efficient processes that were carried out by a highly specialized work force [20]. Accordingly, neither in workflow management literature nor in workflow products was the user interface of much interests, and it seldom provided more than a simple list of work tasks at a first glance.

Instead, serviceflow management aims at explicitly equipping the service workers involved with access to process patterns (standard plans) and individual histories (based on enhanced user interfaces) with possibly far reaching consequences. As pointed out in [10] service delivery requires "empowered" employees responsible for both decision making and execution. Improved support in information technology may, thus, contribute to lessening the de-skilling tendency of encounter based service delivery in general.

Or to put it this way, comprehensive service understanding at delivery time goes beyond workflow assumptions of separating plans and actions in order to certainly improve situatedness and decision making [24], adding to an overall service quality.

5.2 Customer Orientation

Apart from supporting the aspects of flexibility/situatedness and a sharing of service history, each of which already contributes to quality in service delivery, with the concept of serviceflow management we pursue a further step in customer orientation. An example will make this clear.

In his article "Process-oriented Architectures for Electronic Commerce and Interorganizational Workflow" [1], van der Aalst distinguishes different kinds of interorganizational processes in order to select suitable architectures[1]. For a process type called "loosely coupled", which allows a process to be split up into parts and be executed in parallel, he gives the following example of an ordering process, s. figure 9. Briefly, the customer orders an item which causes the producer to order two sub-

[1] Referring to the work of the WfMC van der Aalst argues: "These interoperability standards address the technical issues but not the content of the coordination structure. It is our belief that possible conceptual architectures for truly supporting interorganizational workflows should be explored before solving the technical issues (mainly syntactical)".

items from two suppliers, then the producer assembles the product and delivers it. The figure is presented in the usual way, exhibiting an *overall process* with *subprocesses*.

Now consider the following in the context of this sample process: in serviceflow management we are mainly interested in those parts of the process which interact with the customer. Accordingly, we distinguish conceptually between customer contact and support processes as indicated in figure 9.

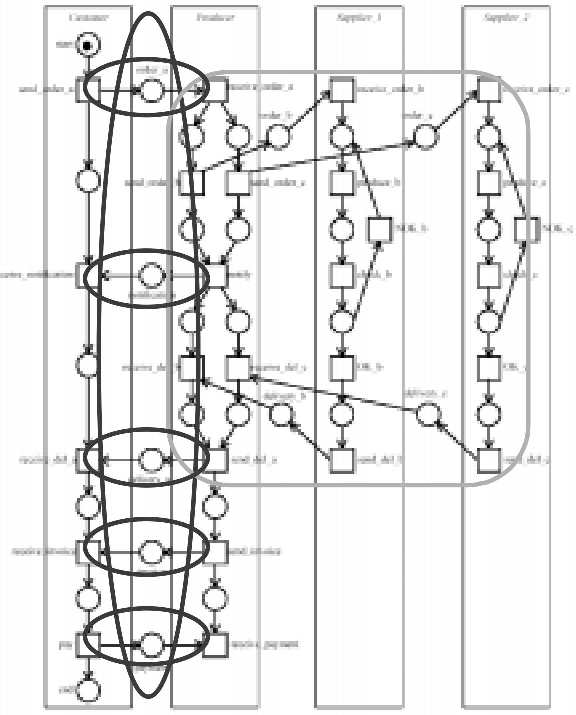

Figure 9: A conceptual distinction supporting customer orientation (process example from [1])

This allows for differences in their support and has several consequences:

1. It allows for prioritizing the serviceflow over subprocesses. In their book “The Brave New Service Strategy” [10] the authors discuss an actual case which can be seen as an instance of the given example. Under the perspective of service quality they summarize the following process situation. A customer orders a computer on the Internet. However, shortly afterwards he or she has to cancel the order for whatever reason. Although trying different media to reach the provider – e-mail, phone, fax – the customer fails to succeed. Some days later he/she finally gets some response indicating that the best way of handling the situation is to “simply” have the computer delivered as planned and then send it right back. Two things are remarkable in our context. Obviously, a service point for canceling an order was missing – either intentionally or unintentionally. Secondly, the background process seems to direct the interaction with the customer. Thus, with *prioritizing* the

serviceflow over the background processes we argue *for carefully designing service points* as well as their interaction with background processes. This leads to the next aspect.

2. With conceptually distinguishing a serviceflow from other processes we revalorize customer contact in service points. As an important by-product, the simple concept of service points brings the customer orientation to each actor/stakeholder involved by providing a concept and language. More often than not, organizations *get stuck* in focusing on their *internal processes* instead of *designing customer contact points*. Furthermore, the metaphor may also be applied in order to redesign or design new electronic services by carefully deciding which of the service points may get transferred into e-service points. This structured approaching was applied in an e-government project in the city state of Hamburg (Germany) supporting the postal vote application through the city's web portal on the basis on serviceflow management (see [14][16]). Considering the healthcare example, it seems difficult to impossible to introduce e-services as most of the time the patient has to be physically present to undergo examinations, care or treatment. However, the first service point may be a candidate for a virtual consultation. The patient could describe the problems and receive a referral slip while being helped to choose from different specialists to consult. Also, an additional e-service point may be added to the existing scenario. For choosing among hospitals, the patient may consider detailed information about possible hospitals over the Internet at home and then again consult a specialist in order to make the decision.
3. The conceptual distinction also helps to discern differences in the software support of serviceflow and background processes according to *their distinctive nature in cooperation*. E.g. coordinating the work of specialists seems to require much more flexibility than the ordering of an implant which can be performed by workflow support.

6 Summary and Further Work

With serviceflow management we contribute to supporting inter-organizational service processes and applications. Having exemplified typical requirements for a healthcare service, we state three main requirements: flexibility in process execution, interoperability, and customer orientation.

The concept of serviceflow management corresponds with recent research in inter-organizational processes in that we apply XML-process representations in order to exchange case/instance based process knowledge among providers. This enables flexibility as well as interoperability and thus *goes beyond* former "bulk"-based *workflow* execution. Sharing process history even allows for empowerment of highly functional specialized work places in service delivery. Applying the same technology (XML-process representation) in order to allow for flexible process execution at service points, we contribute to a set of aspects such as flexible service delivery within organizations, standardization, future template-based contracting and dynamic configuration of services within a provider network.

Furthermore, the conceptual distinction between a serviceflow and background processes together with a revalorization of service points and their design contribute to a *customer orientation beyond a mere process orientation.* The latter is still found in several recent publications and can be traced to business process redesign where efficiency aspects were considered sufficient tokens of customer orientation.

Future work needs to address several aspects. Applying serviceflow management to different service processes will provide further insight into what are suitable architectures. Exploring spreading technology will lead to the rebuilding and extending of existing components, in respect to stage 0: extending the serviceflow representation, to stages 1-3: redesigning existing components, and to stage 3: extending generic interfaces for service points.

To sum up: if inter-organizational processes are designed to converge e-commerce, customer relationship management and supply chain management and thus to enable comprehensive service processes, we argue for serviceflow management for clearly focusing on the customer and his/her concern. This asks for metaphors, concepts and architectures beyond traditional workflow management as presented to serve an (e-) service-based economy.

References

1. van der Aalst, W.M.P.: Process-oriented Architectures for Electronic Commerce and Interorganizational Workflow. In: Information Systems, 24 (8), (2000) pp. 639-671
2. van der Aalst, W.M.P., Jablonski, S.: Dealing with Workflow Change: Identification of Issues and Solutions. International Journal of Computer Systems, Science, and Engineering 15 (5), (2000) pp. 267-276
3. van der Aalst, W.M.P. and Kumar, A.: XML Based Schema Definition for Support of Inter-organizational Workflow. University of Colorado and Eindhoven University of Technology Report. (2001) http://spot.colorado.edu/~akhil/pubs.html
4. Alt, R., Puschmann, T., Reichmayr, C.: Strategies for Business Networking. In: Hubert Österle, Elgar Fleisch, Rainer Alt (eds): Business Networking – Schaping Collaboration Between Enterprises, Springer-Verlag. (2001) pp. 89-11
5. Casati, F., Ilnicki, S., Jin, L.-J., Krishnamoorthy, V., Shan, M.-C.: eFlow: a Platform for Developing and Managing Composite e-Services. HP Labs Technical Reports HPL-2000-36 (2000) www.hpl.hp.com/techreports/2000
6. Christophide, V., Hull, R., Kumar, A., Simeon, J.: Workflow Mediation using VorteXML, Data Engineering 24 (1), March (2001)
7. Gabbot, M., Hogg, G.: Consumers and Services, John Wiley & Sons, Chicester, (1998)
8. Grefen, P., Aberer, K., Ludwig, H., Hoffner, Y.: CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises. Data Engineering, Vol. 24, No. 1, March (2001)
9. Gutek, B., The Dynamics of Service, Jossey-Bass Publishers, San Francisco (1995)
10. Gutek, B., Welsh, T.: The Brave New Service Strategy: Aligning Customer Relationships, Market Strategies, and Business Structures, Amacom, (2000)
11. Harrison, S., Dourish, P.: Re-Place-ing Space: The Roles of Place and Space in Collaborative Systems, Proceedings CSCW'96, pp. 67-76 (1996)
12. Hoque, F.: e-Enterprise – Business Models, Architecture, and Components. Cambridge University Press (2001)

13. Klischewski, R., Wetzel, I.: Serviceflow Management for Health Provider Networks. Information Age Economy. Proceedings 5th International Conference Wirtschaftsinformatik (Business Information Systems). Physica, Heidelberg, (2001) pp. 161-174
14. Klischewski, Ralf; Wetzel, Ingrid, XML-based Process Representation for e-Government Serviceflows, Schmid, B., et al. (ed.): Towards the E-Society: E-commerce, E-business, and E-government (I3E 2001, IFIP). Dordrecht: Kluwer, 2001, pp. 789-802, (2001)
15. Klischewski, R., Wetzel, I., Bahrami, A.: Modeling Serviceflow. In print: Proc. of the 1st International Conference on Information Systems Technology and its Applications (ISTA). Kharkiv, Ukraine, June, (2001)
16. Klischewski, R., Wetzel, I.: Serviceflow Management: Caring for the Citizen’s Concern in Designing E-Government Transaction Processes, to appear in HICSS ‘35 (Hawaii International Conference on Systems Sciences), (2002)
17. Kumar, A., Zhao, J.L.: Workflow Support for Electronic Commerce Applications, http://spot.colorado.edu/˜akhil/ (1999)
18. Lazcano, A., Alonso, G., Schuldt, H., Schuler, C.: The WISE approach to Electronic Commerce. International Journal of Computer Systems Science & Engineering, special issue on Flexible Workflow Technology Driving the Networked Economy, Vol. 15, No. 5, September (2000)
19. Merz, M.: E- Commerce und E- Business. Marktmodelle, Anwendungen und Technologien, Springer Verlag, (1999)
20. Mintzberg, H.: The Structuring of Organizations, Prentice-Hall, (1979)
21. Österle, H., Fleisch, E., Alt, R. (eds): Business Networking – Shaping Collaboration Between Enterprises. Springer-Verlag (2001)
22. Schmid, R.E., Bach, V., Österle, H.: Mit customer Relationship Management zum Prozessportal. In: Bach, V., Österle, H. (eds), Customer Relationship Management in der Praxis. Springer (2000)
23. Shegalov, G., Gillmann, M., Weikum, G.: XML-enabled Workflow Management for E-Services across Heterogeneous Platforms. VLDB Journal 10 (1), (2001) pp. 91-103
24. Suchman, L.: Plans and Situated Actions. The Problem of Human-Machine Communication. Cambridge (NY), Cambridge University Press (1987)