

Serviceflow Management: Caring for the Citizen's Concern in Designing E-Government Transaction Processes

Ralf Klischewski, Ingrid Wetzel

*Hamburg University, Department for Informatics
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany*

{klischewski, wetzel}@informatik.uni-hamburg.de

Abstract

Many authorities have started to use web sites to provide services based on transactions. The guiding vision is often a process portal for citizens that provides access to the countless variations of administrative processes (transactions). But connecting the portal's functionality with internal IT supported processes of the various administrative units is not only a technical challenge. As expectations towards complex online services rise, innovations in process automation need to be combined with maintaining and/or enhancing service quality.

In this paper, we follow service as the guiding vision and try to explore what it needs to enable the service providers (governmental institutions and private partners) to fulfill their mission, i.e. the satisfaction of citizens' needs and concerns. Based on an e-service implementation in the city state of Hamburg, we suggest the serviceflow management approach to model patterns of consecutive service points as well as to represent individual process information using XML. Focussing on transparency for citizens, flexibility for staff, and synergy within provider networks, we claim that reaching a serviceflow management agreement improves the service quality beyond delivering e-government functions.

1. Service for Citizens in an E-Business World

Services based on transaction processes usually establish some sort of service relation beyond simply providing access to information. At first we analyze the special requirements for those services from the citizens' point of view. Then, taking the providers' perspective, we introduce the serviceflow management approach to meet those requirements. The approach suggests modeling patterns of consecutive service points as well as representing

individual process information using XML. Applying this approach, different actors may network to cooperatively provide and manage complex transaction based services. Based on experiences from an e-government project in Hamburg (Germany), we discuss the technical and organizational measures needed (as well as some limitations) in designing e-government transaction processes aimed at caring for the citizen's concern. In particular, we claim that reaching a serviceflow management agreement – centered around the issues of transparency for citizens, flexibility for staff, and synergy within provider networks – leads to an environment where all actors involved have more options to improve the quality of e-government services.

Service is aimed at the satisfaction of needs. Whether the relation between government administration and governed citizens should be described in terms of service is subject to discussion. One can observe aspects of a provider–consumer relation just as in other services, but there is usually no market to choose a provider, and in many cases citizens may not choose at all since they are forced by law to call on administrative services. However, e-government agendas frequently emphasize (besides efficiency or other money saving goals) the improvement of service quality. Taking this seriously, we follow service as the guiding vision and try to explore what it needs to enable the service provider (authorities and private partners) to fulfill their mission, i.e. the satisfaction of citizens' needs based on caring for the citizens' concern.

Accepted definitions of the notion of service ([5],[6]) frequently emphasize, besides the satisfaction of needs, the substantial amount of work involved, the simultaneity of production and consumption and/or the lack of permanence (particularly in contrast with the provision of material goods). The person or organization requesting the service may be involved in the process personally (e.g., as a patient) or as the owner of objects belonging to him (e.g., goods in transit).

The introduction of e-government reduces the necessity for citizens to meet face to face with specialized employees in the administration. As the high number of visits on governmental sites indicates, many citizens appreciate this freedom from physical restrictions (places to go, opening hours) as an improvement in service quality. However, in introducing e-government transaction we can not presuppose that this positive attitude will prevail as we move on to provide more complex services (such as transactions) for citizen via the internet. Research on e-business reveals that trust and trustworthiness is a major success factor. Therefore, caring for the citizen's concern is an issue which needs to be addressed in the design and management of transaction processes.

With e-business invading the service sector there is a shift in the quality of service. Speaking in terms introduced by B. Gutek [8], web site visitors experience service as an encounter, whereas service framed by a personal relationship seems to be left behind – or is simulated by sophisticated customer relation management technology. This applies also to e-government web portals that follow the vision of “all at one-stop anywhere any time”. From the use perspective, this works well for providing general information on administrative issues or access information (places to go, opening hours etc.).

However, in designing e-government transaction processes we go beyond offering services which only need one situated encounter of citizen and service provider. With transactions – e.g. parking permission, postal vote application, birth registration, employment agency, tax return – a person-to-authority relationship starts to unfold where the service quality is determined by the extent to which the provider is able to recognize and address the situated (possibly changing) client's needs (cf. [12]). With an increase in process complexity and/or transaction value, transaction processes become more personalized, involve more organizational units, require more cooperation and expertise, make staying in touch with the client (the citizen) more difficult, and may cause more damage in case of a processing breakdown. For those services citizens usually seek to identify an employee in the administration (at least at the front desk) who (potentially) will look after their concern. But with e-government nobody is there anymore. Or there is only someone to assist your application (or another kind of service demand) but with no personal ties to the actors in charge of processing.

During transaction processing, i.e. while the particular service is carried out, appropriate technical and organizational support must meet requirements from at least two perspectives (here, we exclude privacy and other aspects):

1. The citizen (or his/her agent) expects recognition of his/her concern and expects accountability (i.e. trans-

parency of authorities in charge) throughout the chain of activities/operations.

2. The authorities (and maybe additional, private service providers) try to address situated/changing needs of citizens (in order to enhance service quality and/or to fulfill the legally granted rights of citizens) as well as to efficiently manage the cooperation between the subsequent organizational units involved.

In short, what is needed is a combination of efficient process management (in accordance with metaphors such as business process or workflow management) and customer relation management, suitable to be applied across the organizational boundaries of the public and private service providers involved. And in the end it should be the client who decides whether his needs have been satisfied.

2. Serviceflow Management for E-Government Transaction Processes

Given these requirements, an e-government transaction model of web-based self-service points with the actual operation carried out somewhere in the “back office” does not serve as a suitable guiding vision. Instead, as accountability and flexibility throughout the process are success factors for relationship based services, we introduce serviceflow management (SFM) as a generic concept to compensate for increasing anonymity and to care for the citizen's concern in the redesign of government transaction processes. The concept has been developed especially to meet the requirements of public service domains such as e-government and e-health, but it can also be applied to other service domains such as education or tourism (see [13], [14], [15], [16], [11]).

Where the client naturally follows her/his individual concerns, the service provider applies a professional rationale. E.g., service is regarded as a piece of work or a performance by a business organization, the net value of which is based on the recognition and satisfaction of customer needs. To this end, standard processes are, wherever possible, adapted to the requirements of each service situation [13]. Within e-government, the company's profit or shareholder value are no essential factors. However, there are many interests driving in the same direction, and authorities and their private partners have to cope with the fields of tension between routinization and personalization as well as between standardized process patterns and situated process execution.

The notion of *serviceflow* is meant to pick up both of these perspectives:

- From the *customer's perspective*, a serviceflow gives a customer the feeling of being embedded in a coherent ‘flow of service’ taken care of by the service

organization(s) where the service provided ‘follows’, ‘accompanies’ or ‘precedes’ the customer as she/he moves through time and space.

- From the *service provider’s perspective*, the emphasis is on the integration and coherence of all situated sub-services across temporal, spatial and team boundaries, which are combined to form a continuous and complex overall service to satisfy the client’s need (based on standard processes).

On the one hand, supporting cross-organizational process management in the service domain requires new approaches. On the other hand, any new approach must relate to the ideas already implemented in existing organizations and technology, e.g. workflow management (see also below: 2.3, 2.4) or business networking (cf. [21]). In the following, we will draw on these approaches, but we need to go beyond them to address the specifics of e-government as one of the major public service domains.

The SFM approach has been applied to designing and implementing e-government transaction processes in the city state of Hamburg (Germany). The pilot project covers the postal vote application through the city’s web portal. In our case, the serviceflow management approach is the adopted basis for IT application and cooperation between the different service providers: the commercial portal provider of www.hamburg.de, the city’s central department for application programming (“Senatsamt für Bezirksangelegenheiten”) and the city’s voting department responsible for the temporary voting offices. Service delivery based on SFM first started in August 2000. As expected, the majority of the personalized serviceflows follows the designed serviceflow pattern (which, at first, seems to be much like a workflow). However, a number of variations, uncertainties, exceptions and failures frequently occur (e.g., a citizen moves to a new address before the voting offices could start processing his/her application), forcing the actors involved to react flexibly based on the situated evaluation of the citizen’s concern.

The SFM approach includes a modeling and an implementation strategy. We explain both with respect to this case and to other related approaches published lately in the area of workflow management and e-government.

2.1. Modeling Serviceflows

Serviceflow modeling picks up on modeling techniques and approaches which have evolved through the 90’s in the area of process modeling (workflow, business processes) and object modeling (mainly UML). We try to incorporate these approaches as much as possible, striving for integration and trying to keep specific supplements to a minimum. Our modeling approach has also been inspired by recent work in the field of computer supported

cooperative work (CSCW). Being concerned with social relations and interactions (service, work cooperation), we try to introduce modeling as a tool to bring out the different actors’ perspectives and keep them involved while envisioning change and deciding about the design and use of information technology within the service performance.

The modeling is based on object oriented, workflow and user oriented modeling techniques. We model serviceflow patterns by identifying cross-organizational sequences of service points, each capturing the specific service tasks and their respective pre- and postconditions from the provider’s point of view. Based on these serviceflow patterns, the organizational units involved can (1) share domain knowledge, establish new norms for transaction and make them public (cf.[24]) as well as (2) pass on information (also to the citizen, if appropriate) about the state of each individual and personalized transaction process which, of course, may deviate from the modeled serviceflow pattern.

The notion *serviceflow* indicates the interrelation of all subservices whereas what actually flows is (1) the customer’s concern (which may evolve over time) related to a service agreement and his/her accumulated service experience (often supported by the customer’s physical or virtual presence) as well as (2) the documented plan and history of each individual sequence of service tasks. The aim of serviceflow modeling is to provide a process representation, which may serve as a basis for

- cooperation agreements between the service providers involved in a serviceflow (process patterns for standardized serviceflows)
- service agreements between client and customer (personalized process patterns)
- individual process documentation to be passed on between the service providers.

Also, modeling must allow for flexibility and decentralized control of what service tasks are to be carried out and what should be the schedule for service tasks to follow.

To simplify matters and to enable structuring from the provider’s point of view, we define serviceflow in terms of service points. A service always creates some social situation, it needs a “place” [9] which frames the situation where service tasks are carried out, e.g.

- service staff evaluating the client’s concern and serving his/her needs (in these situations the client’s presence may vary from being present, being present through telecommunication, or being virtually present through one of his objects or through a representation of his concerns)
- the client is served by some automatic device (e.g. a web portal) on behalf of the service provider.

We call these places service points, and the sequence of a number of service points is a serviceflow (e.g. figure 1). Each service point captures specific service tasks to be carried out and their respective pre- and postconditions from the provider's point of view.

Within the postal vote project (started in fall 2000, see also [14], [16]), we have modeled the future process of citizens applying for postal vote through the web portal of the city state of Hamburg (Germany). Based on interviews with administration staff we developed a serviceflow pattern (figure 1) as well as pattern models for the first three service points, each with lists of pre- and postconditions and of service tasks to be carried out. All tasks are linked to text descriptions (scenarios, role descriptions) which also refer to a glossary. All documents were linked and exported into HTML which allowed for convenient reviewing by the interview partners.

Conceptualizing the citizens' application for postal vote through the web portal we identified four service points with their respective tasks in parentheses:

1. providing application assistance for citizens visiting www.hamburg.de, the city's web portal (opening application, assistance in personalization, on-site evaluation of the application, confirming receipt, serviceflow preview, offering/registering personal reporting channel)
2. inspecting the application at "Senatsamt für Bezirksangelegenheiten", the city's central administration for IT procedures (automatic validity check including selecting the voting office in charge; or exception handling: generating a rejection message and moving directly to service point 4 in case of invalid application)
3. processing the application by the respective voting office (validity check with up-to-date preconditions, preparing personal postal ballot paper, notification for the electoral register, preparing postal ballot paper for delivery, personalized exception handling if necessary)
4. reporting on process through the web portal provider (delivering messages to inform the applicant about the

state of the process, providing information about what to do next and/or whom to contact) through the channel the applicant has selected (web page, email, SMS, etc.).

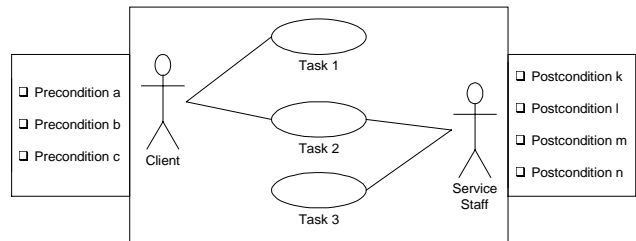


Fig. 2. Service point meta model with service tasks and respective pre- and postconditions

The service tasks are modeled as use cases are in UML (see figure 2). Jacobsen ([10], p. 129) defines a use case as a "behaviorally related sequence of transactions in a dialogue with the system." Within serviceflow modeling, these service tasks either refer to a set of activities carried out by service staff with support from an IT system (indicated in the model by connecting tasks and service staff) or to a set of operations carried out by the IT system, possibly in dialogue with the client (indicated in the model by connecting tasks and the client).

The pre- and postconditions represent the contract for interrelating the service points. Note: there is no centralized serviceflow control assumed, neither within any nor across organization(s). During serviceflow implementation the serviceflow model is only a process pattern, i.e. the flow logic may or may not be executed.

Based on the serviceflow modeling appropriate IT support can allow for flexible performance at each service point: e.g. tasks may be carried out although some preconditions are not true, tasks may differ from the pattern although all preconditions are true, or postconditions may not be achieved although tasks have been carried out in accordance with the pattern (or the other way around).

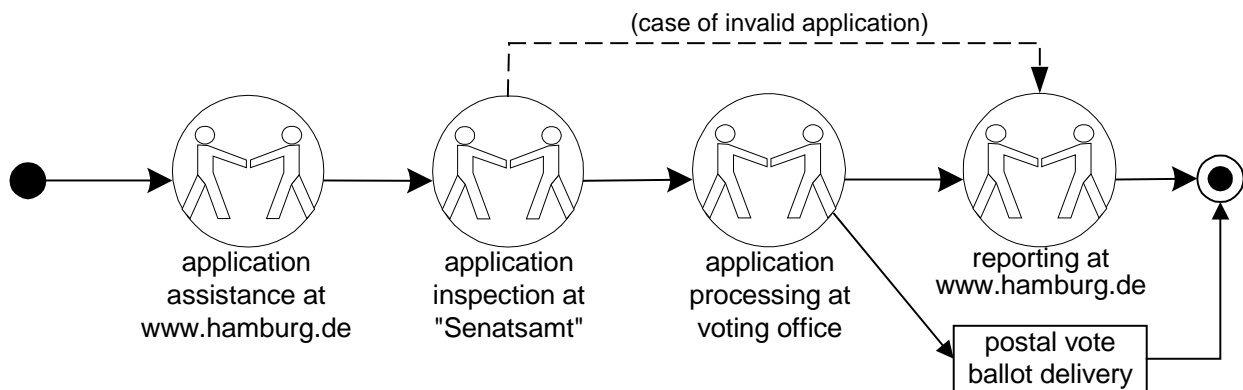


Fig. 1. Serviceflow pattern for the postal vote application starting at www.hamburg.de

The interrelation of service points is subject to possible changes: whereas the serviceflow history (the sequence of service points passed) is, of course, not changeable, the serviceflow schedule, i.e. the part of the serviceflow pattern with service points not visited yet, may be manipulated by deleting or adding service points or changing their order.

However, based on the serviceflow pattern, a documentation of the individual serviceflow can be initiated and updated at each service point. This documentation, i.e. the process representation, includes the individual schedule, the serviceflow history (especially the accumulated postconditions). At the next service point, this process representation can be (automatically) interpreted based on the general serviceflow pattern shared by all actors involved.

For modeling purposes, each service point activity should be linked to a rich description (e.g. a scenario) and a glossary. Cooperation pictures ([17], [18]) can also augment the serviceflow representation to explicate recurrent cooperation relations among the actors involved. To construct an enriched serviceflow model the resulting documents of all these techniques should be related for display, e.g. by interlinking HTML versions of the different documents.

2.2. XML-based representation for e-government serviceflows

Process management across organizational units must cross IT systems' boundaries unless there is a strong force to provide and put through a technical integration of, for example, a central database, file server, workflow engine and a web server. In the e-government domain we cannot presuppose an integrated technology infrastructure. Among the various reasons (due to the limited IT capacities of the public administration) the most significant is that web service providers (which are, e.g., hosting the city's or state's web portal) usually run their own secured IT environment which is significantly different from the main frame oriented IT landscape of the public administration. At the same time we find similar requirements for IT support as in any kind of service organization in terms of customer relation and process management.

Given these requirements, IT support for serviceflow management must be based on the following assumptions:

- In the chain of subsequent service points there is always exactly one service point in charge after process initialization and before ending (the concept for handling parallel, concurrent threads within serviceflows cannot be covered in this paper).

- Each service point in charge has full control of the process (within an agreed technical and organizational frame), there is no central instance (necessary).
- It must be possible to handle a (large) number of individual serviceflows at the same time.
- It must be possible to handle a (large) number of different kinds of serviceflow at the same time (i.e. based on different serviceflow patterns, which may change over time).
- All process information to be communicated between servicepoints must be persistent and portable.

To meet these conditions, process representation for serviceflow management is organized around sending a service float from service point to service point (see also figure 5). Comparable to other approaches seeking to connect sub-processes (see below), we have chosen to implement the process representation in XML documents, i.e. each customer-related serviceflow is represented by one service float implemented in XML.

When starting a personalized serviceflow a service float is created by personalizing a copy of the respective serviceflow master which includes the schedule of service points for this process. At each service point carrying out the activities for the individual client starts by personalizing a copy of the respective service point script master. After all service point activities have been addressed and the postconditions are documented, the updated service float is sent to next scheduled service point. The personalized service point script may then be used for documenting the service activities at this service point. However, the detailed information about service point activities remains with the respective service provider.

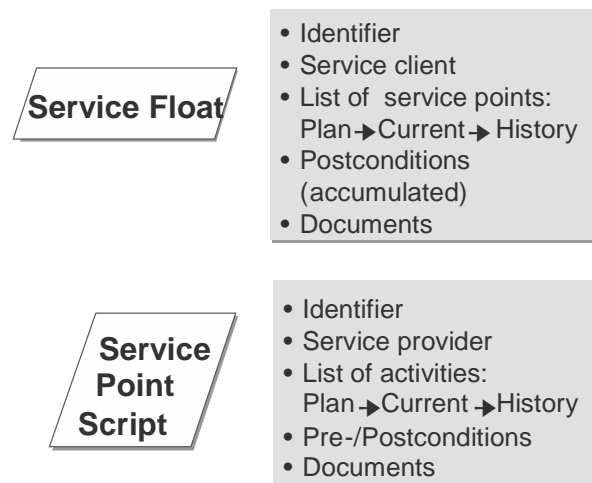


Fig. 3: Structure of the XML documents representing service floats and service point scripts

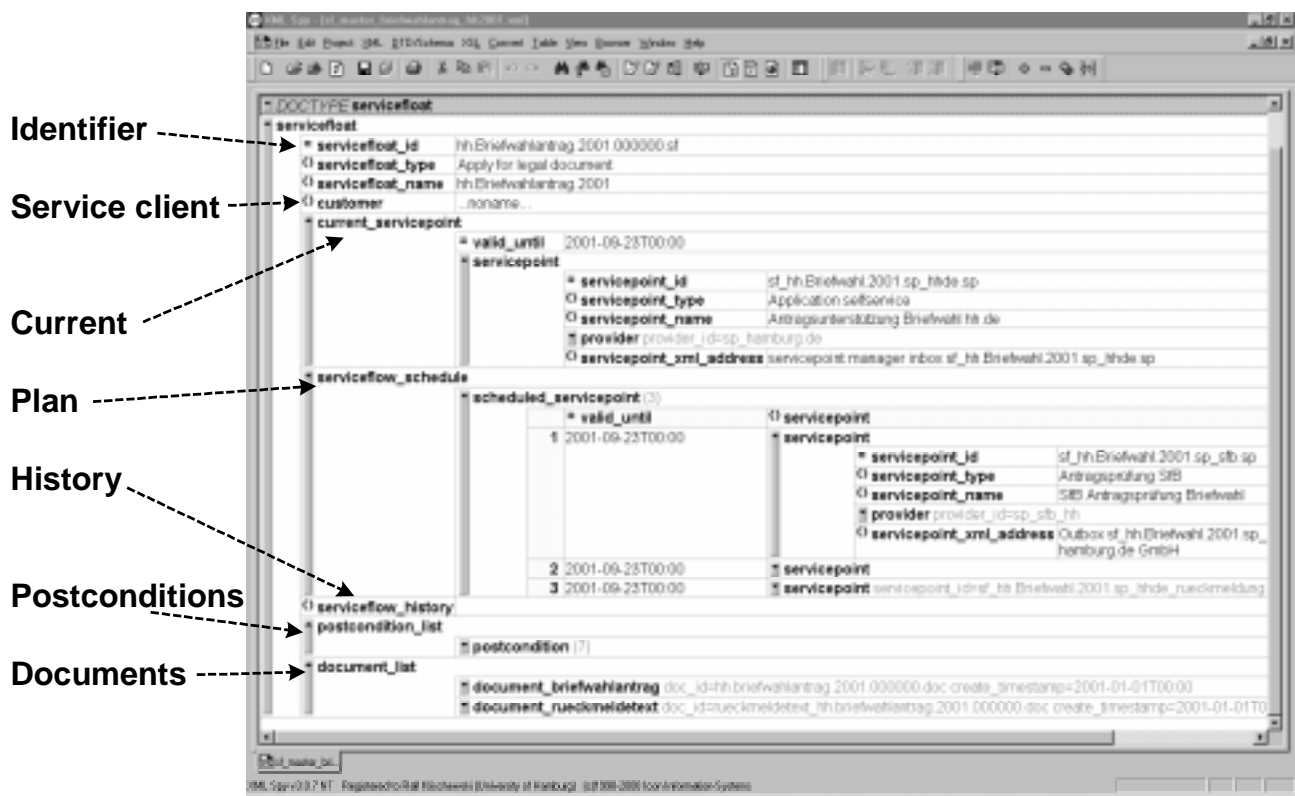


Fig. 4. Editor view of an XML service float ‘master’

The service float contains the following elements (see figures 3 and 4):

- identifier for individual serviceflow (based on serviceflow type/variation)
- basic information on serviceflow client (with possible reference to comprehensive client data)
- current service point (service points are described by identifier, name, type, provider, address)
- list of scheduled service points
- list of passed service points
- list of accumulated postconditions
- 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 service point script prescribing the activities at the ‘current service point’ (see figure 3):

- identifier for individual service point (based on service point type/variation)
- basic information on service point provider (with possible reference to comprehensive provider data)
- current activity (activities are described by identifier, name, type, task; the activity’s attribute list may contain provider id, employee/operator id, document id, time stamp, and more)
- list of scheduled activities

- list of passed activities
- list of preconditions for the set of activities of this service point
- list of postconditions for the set of activities of this service point
- list of documents, i.e. short message texts (for display) or references to full documents (e.g. forms) or document folders (local or as pointed at in serviceflow).

The use of XML is only just spreading and standards in the e-service and/or e-government domain are still being developed (e.g. the “Governmental Markup Language”, see www.egov-project.org). Therefore we had to develop our own framework for a XML-based process representation for serviceflow management (see also 2.3 and 2.4 below). Before starting service operation, the framework must include:

- XML document type definitions (DTD) or schema for service float and service point script and other shared data structures (e.g. postal vote application)
- a set of XML ‘master’-documents for service float and service point script according to the different types and variants of serviceflows.

While using service point scripts is not required for SFM, they can represent the specifics on how to carry out the service activities within a certain type or variant of serviceflow. Thus, using service point scripts allows using

the same technology for supporting a variety of serviceflows. At the web portal, e.g., there will be a set of service point scripts for application assistance (one for postal vote, one for income tax documents, ...), a set for making payments, and other sets.

For dynamic serviceflow management all participating service point providers must agree to follow the process pattern as indicated by the type/variant of serviceflow model and to use the related process representation by

1. trying to carry out activities according to the negotiated serviceflow model and/or as specified in the service point script
2. transferring their own 'current service point' into the list of passed service points while at the same time supplementing the list of accumulated postconditions with the postconditions achieved at this service point
3. extracting the first from the list of scheduled service points to replace the current service point
4. evaluating the address of the new current service point and sending the service float to this address.

All in all, the organizations involved need to agree on the following process-oriented framework for managing serviceflows across organizational units and across the systems' boundaries:

- a set of serviceflow models as a basis for cooperative process management
- a set of XML-DTD/schema and XML 'master'-documents
- a set of rules how to manipulate and share those XML documents (see above, 1.-4.)
- a procedure how to administrate and develop SFM in the given environment (in particular, who looks after the creation, distribution and update of serviceflow models, XML-DTD and XML 'master'-documents).

Note: SFM does not presuppose any special kind of cross-organizational IT infrastructure except the processing and exchange of XML documents. Thus, any kind of organization (public or private) can easily join in the cooperative serviceflow management, and it may independently look after its own IT support as long as it keeps up to the mutual agreement. However, there are IT solutions at hand (see [14], [15]): we recommend a 4-tier architecture (see figure 5) with only one layer encapsulating the serviceflow application logic, which may also be adapted to domain specific requirements.

Summing up, the implementation aims at exchanging XML-based process representation of personalized serviceflows. It is based on an agreement between the organizational units on the modeled serviceflow patterns to be applied, a related set of XML-DTD/schemas and XML 'master'-documents, and a set of rules on how to manipulate and share those XML documents. Each provider involved may use available components (based on a

four-layered IT architecture) or choose (in addition) to develop his own IT support. Instead of realizing one new IT system for all, this approach leaves the freedom for relating the serviceflow-based transactions to the existing IT infrastructure and to realize (or simply maintain) a citizen-oriented flexibility from the specific provider's point of view.

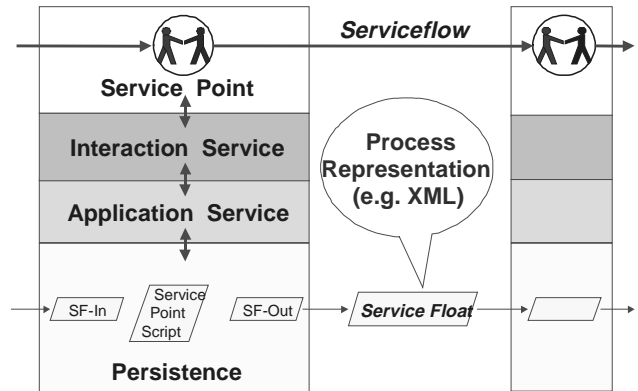


Fig. 5. Layered architecture to process XML-documents representing the serviceflow

2.3. Related work: workflow management

Comparing the SFM approach with related work in the area of workflow management and e-government, we try to identify the unique contribution SFM makes in improving the service quality through designing e-government transaction processes. That means examining in which way recent approaches are suitable in that they allow for accountability throughout the process and support the authorities in dealing with changing customer needs and efficient service delivery.

In addressing inter-organizational processes from a workflow management background, related approaches correspond in aiming at overcoming limitations of current workflow technology such as inflexibility [1] and the assumption of a central engine for control. As in SFM, they apply internet technology to provide technologies that go beyond mere interoperability of workflow systems:

- A comparable approach of Kumar et al. ([19], [2]) uses XRL, an XML-based extensible routing language, in order to capture process knowledge. XRL provides XML elements for each usual workflow modeling construct together with a transformation into petri-net representations. The overall idea is to attach XRL-based routing slips to documents and exchange them between organizations that are involved in the inter-organizational process. A decentralized architecture

with swap-in and swap-out components for receiving and sending the routing slips is given that enables interoperability between heterogeneous system platforms. Flexibility is enabled by instance/case-based routing slips being modifiable at each receiver's site. However, within organizations it is assumed that the routing slip will be transformed into a petri-net which then directs the execution of workflow systems (and may significantly restrict the flexibility of operation at service points).

- Shegalov et al. ([22]) also combine workflow and internet technology. Their approach supposes a centralized architecture with an XML mediator and exchanging instance-based XML messages between clients and servers in the organizations. Referring to the Workflow Reference Model Diagram defined by the Workflow Management Coalition [25], they aim at making the interfaces 2 (to the clients), 3 (to the applications) and 4 (to other workflow systems) uniform and at hiding any details regarding interoperability across system platforms within one component. Similarly to [2] they assume workflow servers within the organizations to control the corresponding sub-processes.

Other approaches address an even wider spectrum of support to provide e-service platforms for managing the dynamic configuration of e-services in inter-organizational processes. Aspects taken into account are the provision of templates [4], the distinction between service and method nodes [3], the support of contract completion [7] and the support of catalogues, guarantees, measurements, multimedia cooperation [20].

Reviewing the latest workflow research, it may be noted that most related approaches now presuppose (as we do in SFM) instance-based XML-represented process information to be exchanged, allowing for ad-hoc changes to the process and enabling the interoperability required. However, in contrast to approaches such as [2] and [22], which assume workflow management systems controlling the process within the organizations, we emphasize the providers taking care of the client's concern at service points where flexibility is required to follow the situated needs of citizens. Accordingly, information technology has to *support* (and empower) service workers instead of *prescribing* and *controlling* their action. SFM takes this into account by applying XML-based process representations to the service points (i.e. service point scripts) as well, so that the predefined patterns of service activities are 'only' plans that serve as resources for situated action (cf. [23]).

With a priority on service orientation, we argue for a different view of cross-organizational process interoperability. We believe that improving service quality is not limited to improving the efficiency of the underlying

processes. Rather, transactions need to be considered as part of the service relation. SFM addresses this aspect by providing a conceptual distinction between serviceflows (processes to take care of the client's concern) and administrative processes in the background or, in general, workflows insensitive to the situated needs of the client. The identification of service points within these serviceflows then helps all actors involved to shape transparent, accountable, and efficient processes.

2.4. Related work: e-government

Lately, transaction management has become a hot topic in e-government research. Ambitious international endeavors such as the eGOV project (see www.egov-project.org) or FASME (Facilitating Administrative Services for Mobile Europeans, see www.fasme.org) focus on inter-organizational workflows as well as improving service quality for the citizens. At the same time, there are countless projects on the local level trying to address the same aspects on a smaller scale.

The most common approach to this challenge is envisioning a central portal for citizens that gives access to a number of standardized workflow processes which are then called 'services'. From the citizen's point of view this may indeed be an improvement in service quality (as it may save the time and effort it would take to personally go and meet the authority). However, from the provider's point of view it should be clear that 'service' in that case means nothing more than delivering functionality (similar to the requested response of a software component) suitable to complete a service encounter (cf. [8], see also section 1). But usually there is no attempt to contribute to service as a *relation* between citizen and authority during the delivery of a particular service and beyond.

The argument of this paper is: as internet applications are decreasing the possibilities of personally serving the citizen, we need conceptual approaches and adequate technology to take the service idea beyond just delivering anonymous functionality through an inflexible machine interface. Service in e-government means taking care of the citizen's concern – and SFM is an approach which provides a modeling and implementation strategy to address this challenge.

3. Designing Transaction Based Services within an E-Government Network

At this point, the reader might be convinced that SFM is suitable to support cross-organizational transaction processing. But does implementing SFM automatically lead to care for the citizens' concerns? Care is something

that unfolds within human relations, which no serviceflow model, XML representation nor its automatic execution will substitute. But the SFM approach changes the mode of designing transaction processes. It brings the service idea to all actors involved – the officials in charge, public managers, software developers, screen designers etc. Applying this approach helps to identify obstacles on the way towards a service-oriented e-government. In our view, SFM frees the actors involved from the restrictions of workflow and back office systems, it is a necessary prerequisite to the authorities' (and partners') carrying out services aimed at caring for the citizens' concerns.

As our focus is on services provided across organizational borders, the design of transaction based services must seek to establish a cooperation of service provider units. Within this network, each contribution to the process may be a service point, thus raising the question of how to care for the citizens' concerns at this point. As modeling serviceflows seeks to divide the overall service into manageable parts with transparent relations between them, it breaks it down into the various issues to be solved in designing transaction processes. It puts a number of questions on the agenda, which must be answered to achieve a cooperation agreement for the management of the serviceflows in focus (for the purpose of illustration we provide some example options also discussed in the postal vote project):

- **Transparency for citizens:** The citizen/client (or his/her agent) expects recognition of his/her concern and looks for accountability throughout the chain of activities/operations: to what extent should the transaction process be visible to citizens? What are the service points and respective tasks within a serviceflow? Should only the serviceflow pattern be public, or may each citizen look at how his/her personal concern is taken care of? Where are the places to "meet" the provider in cyberspace?

Example: Having applied for postal vote via the internet, the citizen receives an identifier to track the processing of his/her application online.

- **Flexibility for staff:** Officials in charge must follow operational guidelines based on laws and management directives, but at the same time they need flexibility to manage the circumstances of individual cases at each service point: which of the service tasks can/should be automated, which ones should not? To what extent may officials 'deviate' from the modeled serviceflow pattern, and how should that be accounted for? What are the options and (dis-)advantages of turning the back office into a (virtual) front office with sophisticated means to communicate with the citizens?

Example: the official at the voting office recognizes an incomplete forwarding address and is supported by the

IT system by automatically calling the citizen on the phone or sending an email or SMS to retrieve the missing piece of information.

- **Synergy within provider network:** As each service provider within the network applies his own rationale, they need to negotiate a balance of costs and benefits: how can the various interpretations of and attitudes towards service be combined in one coherent serviceflow? Where are the resources needed to enhance service quality (i.e. to care for the citizens' concerns) and how can they be allocated? What kind of responsibilities need to be assigned to maintain the serviceflow management network (e.g. responsibilities for creating/negotiating new serviceflow patterns, exception handling, archiving, caring for privacy issues, ...)?

Example: To reduce the amount of exception handling during application inspection and processing, citizens can call on personal assistance for filing the postal vote application online in case they consider the automatic help provided insufficient.

Putting these (and many more detailed) questions on the agenda for designing transaction based services will certainly have an impact on any e-government project. Summing up, serviceflow management means modeling serviceflow patterns and providing an infrastructure for exchanging and manipulating individual XML-based process representations. It enables efficient transaction processing in the fields of tension between routinization and personalization as well as between standardized process patterns and situated process execution. The approach offers the potential for (re-)designing and managing e-government transaction processes across organizational boundaries with respect to flexibility and accountability in the course of each personalized serviceflow. Taking the satisfaction of citizens' needs seriously, it allows all actors involved (authorities, private partners, and the citizens) to care for and keep track of the citizen's concern throughout the transaction process. In this way, authorities (and private partners) can proceed to fulfill their mission, i.e. being at the service of the citizens.

4. Acknowledgement

The authors would like to thank the anonymous reviewers for helpful comments and Gudrun Parsons for polishing the language. This research has been supported by hamburg.de and the city state government of Hamburg.

5. References

- [1] W.M.P. van der Aalst and S. Jablonski, "Dealing with Workflow change: Identification of Issues and Solutions",

- International Journal of Computer Systems, Science, and Engineering* 15 (5), 2000, pp. 267-276.
- [2] W.M.P. van der Aalst and A. Kumar, "XML Based Schema Definition for Support of Inter-organizational Workflow", *University of Colorado and Eindhoven University of Technology Report*, <http://spot.colorado.edu/~akhil/pubs.html>, August 2001.
- [3] F. Casati, S. Ilnicki, L.-J. Jin, V. Krishnamoorthy, and M.-C. Shan, "eFlow: a Platform for Developing and Managing Composite e-Services", *Hewlett Packard Laboratories, Technical Report 2000-36*, www.hpl.hp.com/techreports/2000/HPL-2000-36.html, March 2000.
- [4] V. Christophide, R. Hull, A. Kumar, and J. Simeon, "Workflow Mediation using VortexXML", *Data Engineering* 24 (1), March 2001.
- [5] Deutsches Institut für Normung DIN (German Institute for Standardization) (ed.), *Service Engineering*, Beuth, Berlin, 1998.
- [6] *Encyklopaedia Britannica Online* (<http://www.eb.com>). © 1994-1999 Encyklopaedia Britannica, Inc.
- [7] P. Grefen, K. Aberer, H. Ludwig, and Y. Hoffner, "CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises", *Data Engineering* 24 (1), March 2001.
- [8] Gutek, B., *The Dynamics of Service*, Jossey-Bass Publishers, San Francisco, 1995.
- [9] S. Harrison and P. Dourish, "Re-Place-ing Space: The Roles of Place and Space in Collaborative Systems", *Proceedings CSCW'96*, ACM, New York, 1996, pp. 67-76.
- [10] Jacobson, I., et al., *Object Oriented Software Engineering. A Use Case driven Approach*, Addison-Wesley, Wokingham, 1992.
- [11] R. Kaschek, R. Klischewski, and I. Wetzel, "A Virtual Debate on Serviceflows", *EMISA Forum*, 1/2001, pp. 16-23
- [12] R. Klischewski, "Abstrakte Bedürfnisse und konkrete Beziehungen – oder: Wie man Services (nicht) modelliert", *Proceedings Modellierung 2000*, Fölbach, Koblenz, 2000, pp. 19-26.
- [13] R. Klischewski and I. Wetzel, "Serviceflow Management" *Informatik Spektrum*, 23 (1), February 2000, pp. 38-46.
- [14] R. Klischewski, I. Wetzel, and A. Bahrami, "Modeling Serviceflow", *Information Systems Technology and its Applications. Proceedings ISTA 2001*. German Informatics Society, Bonn, 2001, pp. 261-272.
- [15] R. Klischewski and I. Wetzel, "Serviceflow Management for Health Provider Networks", *Information Age Economy. Proceedings 5th International Conference Wirtschaftsinformatik (Business Information Systems)*, Physica, Heidelberg, 2001, pp. 161-174.
- [16] R. Klischewski and I. Wetzel, "XML-based Process Representation for e-Government Serviceflows", *Proceedings of 1st IFIP Conference on E-commerce, E-business, and E-government*, IFIP, 2001.
- [17] A. Krabbel, I. Wetzel, and S. Ratuski, "Participation of Heterogeneous User Groups: Providing an Integrated Hospital Information System", *Proceedings of the Participatory Design Conference (PDC'96)*, CPSR, Palo Alto, CA, 1996, pp. 241-250.
- [18] A. Krabbel and I. Wetzel, "Designing Hospital Information Systems: Handling Complexity via a User-Oriented Document-Based Approach", in: Armoni, A. (ed.), *Healthcare Information Systems: Challenges of the New Millennium*, Idea Group Publishing, Hershey, PA, 2000, pp. 1-26.
- [19] A. Kumar and J.L. Zhao, "Workflow Support for Electronic Commerce Applications", <http://spot.colorado.edu/~akhil/>, 1999.
- [20] A. Lazcano, H. Schuldt, G. Alonso, and H.-J. Schek, "WISE: Process based E-Commerce", *IEEE Data Engineering Bulletin*, Special Issue on Infrastructure for Advanced E-Services, 24 (1), March 2001.
- [21] Österle, H., E. Fleisch, and R. Alt, *Business Networking. Shaping Enterprise Relationships on the Internet*, Springer, Berlin, 2000.
- [22] G. Shegalov, M. Gillmann, and G. Weikum, "XML-enabled Workflow Management for E-Services across Heterogeneous Platforms", *VLDB Journal* 10 (1), 2001, pp. 91-103.
- [23] Suchman, L., *Plans and Situated Actions. The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge, NY, 1987
- [24] M. Wimmer, R. Traunmüller, and K. Lenk, "Electronic Business Invading the Public Sector: Considerations on Change and Design", *Proceedings HICSS-34*, IEEE, 2001
- [25] "Workflow Reference Model Diagram", *Workflow Management Coalition*, <http://www.wfmc.org/standards/model.htm>, last view: Sep. 2001