

XML-based Process Representation for e-Government Serviceflows

Ralf Klischewski, Ingrid Wetzel

University of Hamburg, Department for Informatics/Software Engineering

Abstract: Addressing new public challenges such as the one-stop government and improved service quality, we introduce serviceflow management as a generic concept to coordinate cross-organizational e-government processes. Aiming at a serviceflow management infrastructure for networked service providers we present an XML-based process representation of serviceflows as well as a four layered IT architecture for realizing serviceflow related applications.

The research presented is exemplified by the case of citizens applying for postal vote through the web portal of the city state of Hamburg (Germany). The discussion includes how XML-based process representations for e-government serviceflows enable governmental actors to enter the network age without high investment burdens, but with many options for creating the service-oriented government of the future.

1. THE PUBLIC CHALLENGE: ONE STOP – MANY SERVICE PROVIDERS

We are used to speak of “government” or “administration” in the singular even though we know from experience that there is a multitude (if not a “jungle”) of institutions and hosts of employees each responsible for one small portion of the governmental/administrative “business”. Now, with e-government the singular notion has received new emphasis:

- citizens are looking for a one-stop government with single entry points for personalized services (authority-to-citizen, A2C)
- government departments as customers and commercial providers strive to align through centralized web-based market places (business-to-authority, B2A)

- co-operation among government departments and among governments (authority-to-authority, A2A) needs to establish channels and network structures corresponding to a clear understanding of what are central nodes of the network by which one can reach members of the respective subnets.

In this paper, we concentrate on citizens' expectations to contact the relevant government unit through one single point of entry or access (one stop) which will enable the fulfillment of whatever concern he or she might have. With regard to content, the administrative challenge⁷⁴ then is to realize those "single points" which

- provide the unmeasurable amount of (context dependent) information on government and administrative issues (*information*)
- offer to make contact with the vast number of actors and institutions which are all competent authorities (*communication*)
- let citizen enter in the countless variations of administrative processes (*transaction*)

This challenge is multiplied by the need for many "single points" of access with respect to the government's layered structure (e.g. federal/state/municipal or international/national/local level) as well as for a division of competence according to geography or function. In addition, these "single points" must support access through different channels such as web portals, call centers, face-to-face service counters or even one central postal address.

Meanwhile, nearly all governments are publishing information online, including contact information. Also, there are services⁷⁵ that select relevant information and/or competent authorities according to the citizen's personal needs (owing to e.g. time restrictions, disabilities). Many authorities have started to provide transaction services (e.g. filing tax returns). However, up to now there is a lack of organizational concepts and system architectures matching the special requirements of governmental transaction services as well as the restrictions of governmental service providers.

Addressing these requirements we introduce *serviceflow management* as a universal concept to coordinate cross-organizational e-government processes, present an XML-based process representation to support those serviceflows, and discuss IT architectures for service points suitable to realizing serviceflow related applications. The research presented is exemplified by the case of citizens applying for postal vote through the web portal of the city state of Hamburg (Germany). In summing up, we will consider how the concepts and solutions introduced here apply to e-government challenges in general.

⁷⁴ E.g. KPMG (2000, 21) states "in the public sector (...) the challenge is to manage customer relationships through a single channel".

⁷⁵ E.g. the "direct citizen information service" of Hamburg, DIBIS, <http://dibis.dufa.de>

2. SERVICEFLOW MANAGEMENT TO COORDINATE CROSS-ORGANIZATIONAL E-GOVERNMENT PROCESSES

E-business “invading the public sector” (Wimmer et al. 2001) poses technical as well as organizational challenges. Governmental institutions must move to catch up with the rise of the network society, but at the same time they cannot simply copy the concepts applied in commercial domains. From the numerous differences (see e.g. Wimmer et al. 2001) the most significant is that governments do not sell products to customers in a competitive environment. Rather, they provide a vast variety of informational and document-oriented services where, in most cases,⁷⁶ the client-citizen is forced (by laws and regulations) to demand a service from a monopolistic provider. Specifically, much of governmental administration consists of law-based, well-defined processes that take into account the citizens’ concerns and personal situations and produce specific documents and records as well as a number of intended side effects.

Following the vision of a one-stop government, these processes should be accessible through one single entry point, i.e. process management must reach beyond the borders of the competent authority in charge. For developing and using IT support (esp. internet technology) for cross-organizational process management we need an appropriate organizational model. Basically, we may draw on two backgrounds:

- *Workflow management* originally focuses on the flow of worked-on documents as well as on the interrelation of workplaces. Meanwhile, flexibility and cross-border process management have become an issue, but (based on a back office orientation) customer relation management or service quality are usually not taken into account. Technical development is aiming at integrated workflow management systems for modeling and supporting the whole process.
- *Business networking* (e.g. Österle et al. 2000), based on business process analysis/engineering, is a rather comprehensive approach meant to identify and support processes for increasing accountable value in business cooperation and/or for external clients. It provides a guiding vision and a strategy framework for how to achieve cross-organizational integration in e-commerce, supply chain management, or customer relationship management through the extensive use of internet applications (e.g. customer process portals). But this top level approach focuses primarily on the interrelation of business units, not on the workplace or personal interaction level. It may lead to a comprehensive IT strategy, but does not include concepts detailed enough to bring out new application modeling and respective IT architectures.

⁷⁶ One of the notable exceptions are weddings where administration now often engages in marketing activities to attract couples as customers to their wedding sites.

In the following, we will draw on these approaches to organize and manage processes, but they do not sufficiently address the special needs and circumstances of e-government. There we need process models which allow for

- portal access, i.e. one-stop for all services, at the same time
- multi-channel access to services at different entry points or directly at the authority in charge, and
- accountability, i.e. transparency (from the citizens' point of view) of authorities in charge throughout the whole chain of activities
- flexibility to address situated/changing needs of citizens and to enhance service quality to fulfill the legally granted rights of citizens

Given these requirements, the model of self-service points with the actual operation carried out somewhere in the back office does not serve as a suitable guiding vision. Service is process – while ongoing, the service quality is determined by the extent to which the service provider is able to recognize and address the situated (changing) needs of the client, based on an (implicit) agreement (cf. Klischewski 2000). We have therefore introduced the concept of serviceflow management (Klischewski/Wetzel 2000) to meet the special requirements of public service domains such as e-government and e-health (see also Klischewski/Wetzel 2001), but the same applies to other service domains, such as education or tourism. In these domains the challenge is to manage personalized sequences of interrelated activities/operations carried out by actors (humans and non-humans) of different organizational units which, from the clients' point of view, sum up to a personalized service. Serviceflow management is aiming at:

- improved service quality by customer relation management throughout the whole serviceflow,
- resource efficiency in the fields of tension between routinization and personalization as well as between standardized process patterns and situated process execution
- connectivity and suitable computer support for each service point

Based on object oriented, workflow and user oriented modeling techniques, we model serviceflow patterns by identifying sequences of service points, each capturing the specific service tasks and their respective pre- and postconditions from the provider's point of view (Klischewski/Wetzel/Bahrami 2001). In contrast to workflow approaches, serviceflow modeling implies that

- each work-place is a place of service (*service point*)
- flowing data represents customer relations (not the 'products' to work on)
- all process models are resources for personalization
- process governance is decentralized (no central flow engine)

In principle, serviceflow management enables any process to continue individually according to the accumulated postconditions as well as the requested preconditions and situated process planning at each service point. Thus, each service

provider must decide to what extent the respective work organization and IT support will allow for variations of or deviations from the predefined standard processes.

2.1 Case: postal vote application at www.hamburg.de

In our case – citizens applying for postal vote through the web portal of the city state of Hamburg (Germany) – all parties involved have acknowledged that the underlying concept of serviceflow management applies a general perspective and that the selected process of applying for postal vote is only one first example to demonstrate the city’s new capabilities and to learn how to manage the organizational and technical aspects of e-government transaction services. In evaluating and redesigning the service process, we have identified four service points with respective activities/operations in parentheses (see figure 1):

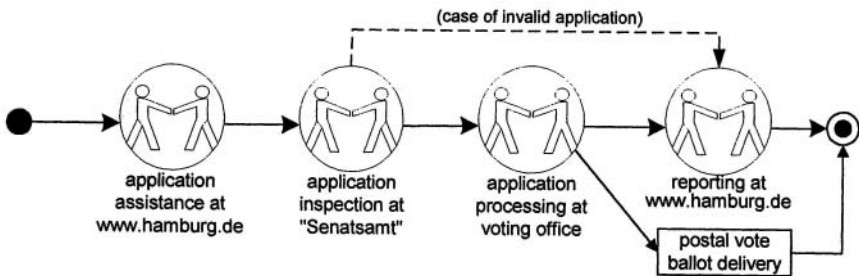


Figure 1. Serviceflow model for a postal vote application

1. providing assistance with the application for citizens at the city’s web portal www.hamburg.de (opening application, automatic assistance in personalization, on-site evaluation, confirming reception, serviceflow preview, offering/registering personal reporting channel, optional: saving application)
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: selecting the voting office in charge if application processing seems possible – or moving directly to service point 4 in case of invalid application)
3. processing of the application by the respective voting office (validity check with up to date preconditions, preparing personal postal vote ballot, notification of the electoral register, preparing postal vote ballot for dispatch, personalized exception handling if necessary)
4. reporting on process by 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 before (web page, email, SMS, etc.)

Other activities/operations not focussing on or reflecting the citizen's personal/situated need are considered support processes, in this case the delivery of the postal vote ballot by regular mail.

Modeling the postal vote application as a serviceflow helps (1) to identify standardized portions (service points) of the overall service, (2) to allocate responsibility for each service point, and thus (3) supports cooperation across organizations and/or organizational units (cf. Klischewski/Wetzel/Bahrami 2001). While the process described above seems pretty straightforward (at least simple enough for prototyping purposes), a number of variations, uncertainties, possible exceptions and failures may occur. Situated needs to be addressed include a citizen's

- moving to a new address before the voting office starts processing his/her application (voting offices open only a few weeks before the election date)
- having lost the postal vote ballot and needing a new one
- not needing to use postal vote after all and wanting to vote at the polling station

However, the administration expects the majority of the personalized serviceflows to follow the designed pattern. In our case, the above serviceflow model is the adopted basis for 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 election department responsible for the temporary voting offices. The city's finance department, responsible for e-government strategy, is also involved in the background.

Given the successful realization of the postal vote application service (which is to support the city state government election in September 2001), the established teamwork will continue to pave the ground for the multitude of future e-government services. Lately, negotiations have started with other administrative bodies in Northern Germany who also want to adopt this new approach. Whereas each citizen process portal and the management of all accessible serviceflows need a negotiated organizational model with specific actors in charge, the IT architecture and technology to support serviceflow management are of general use. In the next sections, we will continue documenting this case to explain about the underlying general concepts for the use of XML and the system architecture for service points.

3. XML-BASED PROCESS REPRESENTATION TO CROSS SYSTEM BARRIERS

Process management across organizational units must cross IT system barriers unless there is a strong force to provide and put through a technical integration (e.g. a central database, file server, web portal, workflow engine). In the e-government domain we cannot presuppose an integrated technology infrastructure. Among the various reasons (which are mainly due to the limited IT capabilities of the public

administration) the most significant is that commercial application and service providers (which, e.g., host the city’s or state’s web portal) run their own secured IT environments which are usually significantly different from the mainframe oriented IT landscapes of the public administration. At the same time, we find similar requirements for IT support in any kind of service organization in terms of customer relation and process management.

Given these requirements, IT support for serviceflow management relies 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 ending⁷⁷
- 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 (great) number of individual serviceflows at the same time
- it must be possible to handle a (great) number of different kinds of serviceflows at the same time (i.e. based on different serviceflow patterns, which may change over time)
- all process information that needs to be communicated between service points must be persistent and portable

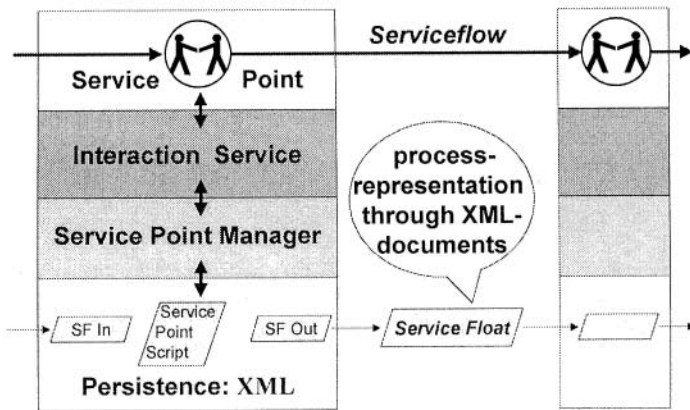


Figure 2. Context of serviceflow process representation

To meet these presuppositions process representation for serviceflow management is organized around sending a service float from service point to service point (see figure 2). The service float contains the following elements:

- identifier for individual serviceflow (based on serviceflow type/variation)

⁷⁷ except for concurrency; the concept of handling parallel threads within serviceflows cannot be covered in this paper

- 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)
- lists of scheduled and 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 related service point script prescribing the activities at the ‘current service point’:

- 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)
- lists of scheduled and passed activities
- lists of pre- and 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

Note: using service point scripts is not required for implementing IT support for service points as described in the following section 4. However, service point scripts represent the specifics of how to carry out the service activities at this point of 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, for example, there will be a set of service point scripts for application assistance (for postal vote, income tax cards, ...). 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 by type/variant of serviceflow model and to use the respective process representation by

1. carrying out activities/operations 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

As XML is used more and more for exchanging structured data between organizations, the documents for serviceflow representation are implemented in XML, i.e. each customer-related serviceflow is represented by one service float in

XML. Meanwhile, research not only addresses the mapping of data structures by using XML, but also to support process management. For example, Lenz/Oberweis (2001) have introduced XML Nets for modeling interorganizational workflows, intended to be executed by a workflow engine. For e-government (assumed to be structurally similar to eBusiness) Greunz et al. (2001) suggest secure XML document containers to support processes aiming at electronic contracting.

When starting a personalized serviceflow a service float will be created by personalizing a copy of the respective serviceflow “master” (which represents the standard process for the type or variant of serviceflow in question) including the schedule of service points for this process. At each service point the activities for the individual client will be started off by personalizing a copy of the respective service point script master (manipulation of this personalized service point script may be used for documenting the service activities at this service point). After all service point activities have been carried out the updated service float will be sent to next scheduled service point.

To enable cross-organizational serviceflow management, the organizations involved need to agree on the following:

- a set of serviceflow models as a basis for cooperative process management
- a set of XML DTD and XML “master” -documents
- a set of rules on how to manipulate and share the XML documents (see above)

Additionally, the actors involved need to agree how to cope with overarching issues such as service quality (e.g. overall response time), creation, distribution and update of serviceflow models and XML documents, privacy and security issues.

4. IT ARCHITECTURES FOR SERVICE POINTS

The approach to serviceflow management as introduced above does not presuppose any kind of shared IT infrastructure except the processing and exchange of XML documents. Thus, any kind of organization (public or private) can easily join the cooperative serviceflow management, and it may independently look after its own IT support as long as it keeps to the mutual agreement. However, there are IT solutions to hand, and we recommend a four layer client-server architecture:

1. Front-end: client to present the user interface
2. Interaction: server layer to organize the user dialogue
3. Serviceflow application: server layer to realize the XML document processing for process representation
4. Persistence: the server’s file system or data base for saving and retrieving XML documents

The IT architecture for a web-based service point as suggested for the postal vote application assistance is shown in figure 3. The user dialogue is organized by a set of templates created in the web content management system (WCMS) of a high-

performance environment for e-business and customer relation management (in this case based on Vignette® and WebLogic® products). These templates include Java method calls addressing the public interface of the serviceflow application layer implemented in Java.

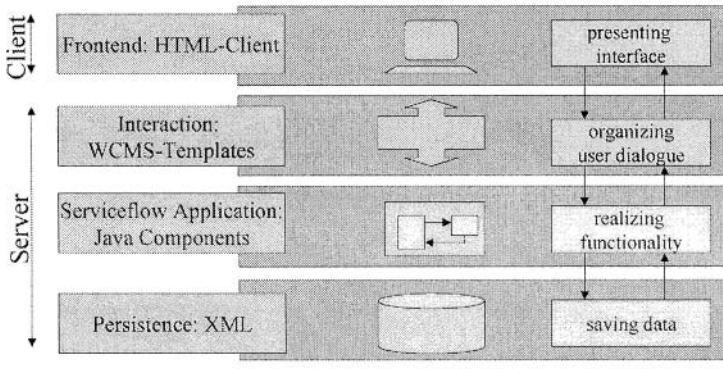


Figure 3. IT architecture for a web-based service point

The components of the serviceflow application layer⁷⁸ – created dynamically for the time it takes to carry out the service point’s activities for a specific customer – encapsulate the processing of the XML documents related to serviceflow management (see figure 4):

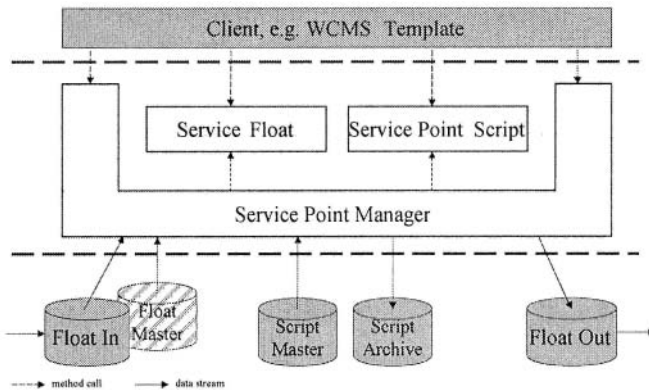


Figure 4. Java components for service point management

- the service point manager includes methods for retrieving the relevant XML files, creating Document Object Models (DOM) of service float and service

⁷⁸ The description is based on the prototype developed at Hamburg University as presented at the end of January 2001. The final solution in this case may differ in some aspects.

point script for a specific customer, saving the manipulated DOMs in XML files and preparing the service float for dispatch

- service float and service point script each include a variety of get- and set-methods (according to the usage of serviceflow process representation, see above) to be called on through the public interface for manipulating the respective XML DOM

Note: This architecture is generic, and layers are interchangeable. The interaction layer of a web-based service point, for example, may be implemented without web content management systems but with Java Server Pages or other web technology, and/or the serviceflow application layer to realize the XML document processing may draw on other than the developed components as long as it meets the agreement on how to use the XML-based process representation.

The same applies to service points without web clients, but with clients based on, for example, mainframes (as in the case of the postal vote application processing). The layered architecture may include the available Java components, but the serviceflow application layer may be implemented in a different technology to provide a suitable interface to existing mainframe applications (in our case, given a Microsoft environment, the developers chose Visual Basic®).

Note: The Java components may be used for all types of serviceflows, no re-implementation is necessary. At the interaction layer each type of serviceflow may require additional programming work. However, a sophisticated architecture for this level may allow the reuse of all components by requesting the relevant data for the specific serviceflow type or variant from the service point script.

In our case the architecture described is leading the current implementation work which aims at realizing the postal vote application through www.hamburg.de and, at the same time, at providing a technical basis for a great variety of e-government processes in the city state of Hamburg. Other challenges to be addressed (beyond the scope of serviceflow management) include scalability and reliability, as well as security and privacy. Most of these are solved within the web portal's IT infrastructure designed for high performance (including secure operations such as web mail), and encryption is used for all file transfers between service points of different organizations. Additionally, some legal, organizational and technical measures need to be implemented to secure citizens' privacy as the commercial portal provider is processing personal data on behalf of the city's authorities.

5. E-GOVERNMENT: TOWARDS MANAGING A SERVICEFLOW INFRASTRUCTURE

In this article we have introduced serviceflow management as a universal means to connect and coordinate cross-organizational e-government processes. We identify serviceflow patterns by modeling sequences of service points, each capturing the

specific service tasks and their respective pre- and postconditions from the provider's point of view. In contrast to the workflow approach, serviceflow management allows for analyzing, modeling, supporting and managing those processes as being related to the citizens' concerns all the way through the chain of service points. This approach meets the unique requirements of e-government services, i.e. portal access (one stop), multi-channel access, accountability/transparency, and flexibility. With serviceflow management it is possible to dissect complex routine processes and delegate parts of the process (e.g. assisting and accepting formal applications) to actors outside the administration (e.g. a web portal provider), whereas the responsibility for the key activities (eg. processing a citizen's application) – as can easily be demonstrate to citizens and all service providers involved – rests with governmental institutions.

To support serviceflow management we have presented an XML-based process representation, which is the only technology necessary to be shared among the network of service providers. The integrating framework for cooperative process management consists of a set of serviceflow models as a basis for cooperative process management, a set of XML documents, a set of rules on how to manipulate and share those XML documents, and an agreement on how to administrate and develop serviceflow management. Organizational and technical integration does not need more than (ongoing) coordination between the service providers involved, shared serviceflow modeling, and the exchange of XML-based process representations! To enable a serviceflow management mainly requires organizational and administrative expertise, but (compared to other cross-organizational process management solutions) only little technical know-how. Thus, governmental institutions in charge can easily take the lead in coordinating all service providers involved.

Already, IT support is available to realize serviceflow related applications. We have developed a generic four-layered service point architecture which allows the exchange of single components as well as a certain freedom of choice in developing and using individual solutions. The feasibility of this approach has been proved as it has guided the development the first e-government transaction service available at the web portal of the city state of Hamburg (Germany). The research and development going on is aiming paving the ground for creating a variety of e-government services and managing a serviceflow infrastructure with as little IT investment as possible.

In this contribution we have concentrated on the challenge of providing single entry points for citizens who want/need to engage with the authorities through personalized transactions which will lead to the fulfillment of whatever concern he or she may have (A2C). However, the concepts and solutions introduced here can be applied to other e-government challenges in the fields of B2A and A2A as well. To give an example, serviceflow management provides a process management solution for government departments aligning with providers to fulfill the service needs of the administration. Cooperation among government departments is of course a

central element of this approach. Adopting the same serviceflow infrastructure (i.e. the integrating framework) easily enrolls new members in the process management network. But even cooperation between such networks (each with their own sets of models and XML documents) will be no problem since already available XML mapping tools are likely to be part of the e-government's IT infrastructure.

In short, XML-based process representation for e-government serviceflows is a promising application-oriented approach which enables governmental actors to enter the network age without high investment burdens, but with many options for creating the service-oriented government of the future.

6. ACKNOWLEDGEMENTS

We wish to thank the student members of the postal vote project, especially Timmy Blank and Nol Shala for implementing and sharing the DTDs and Java components. The authors acknowledge support by hamburg.de GmbH.

7. REFERENCES

- Greunz, M., Haes, J., Schopp, B., Stanoevska, K. Integrating e-government Infrastructures through Secure XML Document Containers. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*. IEEE, 2001
- Klischewski, R. Abstrakte Bedürfnisse und konkrete Beziehungen - oder: Wie man Services (nicht) modelliert, In *Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik. Proceedings Modellierung 2000* (St. Goar, 5.-7.4.), Ebert, J., Frank, U., ed. Koblenz: Fölbach, pp. 19-26
- Klischewski, R., Wetzel, I. Serviceflow Management. *Informatik Spektrum* 23:1, February 2000, pp. 38-46
- Klischewski, R., Wetzel, I. Serviceflow Management for Health Provider Networks. In *Information Age Economy – Innovations, Methods and Applications for Electronic Commerce. Proceedings 5th International Conference Wirtschaftsinformatik (Business Information Systems)*, Augsburg, Sep. 19-21, 2001
- Klischewski, R., Wetzel, I., Bahrami, A. Modeling Serviceflow. In *Information Systems Technology and its Applications. Proceedings ISTA 2001* (June 13-15, 2001, Kharkiv, Ukraine), Godlevsky, M., Mayr, H., ed. Bonn: German Informatics Society, Lecture Notes in Informatics, 2001, pp. 261-272
- KPMG. Leading the Transformation to e-Government. Seven Things You Need to Know. Oct. 2000, www.us.kpmg.com/RutUS_prod/Documents/12/Trans3_Lite.pdf (download: 2001-2-15)
- Lenz, K., Oberweis, A. Modeling Interorganizational Workflows with XML Nets. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*. IEEE, 2001
- Österle, H., Fleisch, E., Alt, R. *Business networking. Shaping enterprise relationships on the Internet*. Berlin: Springer, 2000

Wimmer, M., Traummüller, R., Lenk, K. Electronic Business Invading the Public Sector: Considerations on Change and Design. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*. IEEE, 2001