Stages of Growth in e-Government: An Architectural Approach

Marijn Janssen and Anne Fleur van Veenstra
Faculty of Technology, Policy and Management, Delft University of Technology
MarijnJ@tbm.tudelft.nl
AnneFleur@gmail.com

Abstract: Governmental agencies from all over the world are in various stages of development to migrate their traditional systems architectures to more horizontally and vertically integrated architectures. In this paper a stages of growth model for the development of information architectures for local governmental agencies is presented. By analyzing discontinuities in the architectures coordinating back and front office applications five stages are derived. The five-stage model consists of 1) no integration, 2) one-to-one messaging, 3) warehouse, 4) broker and 5) orchestrated broker architecture. Public decision-makers can use these stages as a guidance and direction in architecture development, to reduce the complexity of the progression of e-government initiatives, to communicate changes to the rest of the organization and to provide milestones to evaluate and control cost of architecture development.

Keywords: Information architecture, local government, stage models, coordination, information broker, web service orchestration

1. Introduction

Citizens in all countries are calling for better services at lower costs and for more responsiveness in a dynamic and continuously changing environment. The information architectures need to be in pace with these development. Initiatives are confronted with a highly fragmented information architecture that has often been vertically organized around departments. In general, in the current situation each department has developed its own information systems in relative isolation, and for each product or service a separate information system exists (Janssen, Wagenaar and Beerens, 2003). As such there is a need for more horizontally and vertically integrated architectures addressing the communication between systems within and between departments and organizations. Most public agencies have felt the need for more integrated architectures, and in one way or another, often questions concerning the management and development of these architectures have arisen.

The development of information architecture can evolve through a number of phases or stages of growth. Stages of growth or evolution models are popular in organizational research and information systems and have been applied in various domains (e.g. Greiner 1972, Nolan 1979, Layne and Lee 2001, Grover and Segars 2005). Nolan (1979) made the stages of growth model popular due to the intuitive appeal of his model. Stage models aims at deconstructing information systems development into a series of stages and development goes from one stage to another stage. Although of importance, stage models do no not focus on change management and organizational development strategies. This implies that stage models need always be accompanied by a careful chosen change management strategy. Change management strategies requires the addressing of all kind of process management issues, including the creation of shared vision, motivation of stakeholders, dealing with resistance to changes, obtaining political support, intervention strategies, and so on (e.g. Bruijn, Heuvelhof and Veld, 2002).

The evolution of government efforts to provide electronic services in this stage-by-stage manner has already been described by a number of authors (e.g. Layne and Lee 2001, Moon 2002, Rao, Metts and Monge 2003, Ghasemzadeh and Sahafi, 2003). A model that is widely referred to is that of Layne and Lee (2001), which focuses mainly on the service provision by municipalities to citizens and business. They distinguish four stages: 1) cataloguing, 2) transaction, 3) vertical integration and 4) horizontal integration. The stages are explained in terms of different levels of integration and involved complexity. Reddick (2004) describes the e-government development of municipalities in the United States in a two-stage model, which is similar to the first two stages of cataloguing and transaction. He distinguishes between G2C (government to citizen), G2G (intra-governmental) and G2B (government to business). He concludes that in the field of G2C, the services are largely in the cataloguing category, in G2G, they are in the transactional stage and for G2B they are most advanced. Moon (2002) proposed a five-stage model, with stages named 1) information dissemination/ cataloguing, 2) two-way communication, 3) service and financial transactions, 4) vertical and horizontal integration, and 5) political participation. The fifth stage thus adds the political dimension of e-government. Rao, Mets and Monge (2003) have developed a

Reference this paper as:
Janssen M and van Veenstra AF (2005) "Stages of Growth in e-Government: An Architectural Approach" *The Electronic Journal of e-Government* Volume 3 Issue 4, pp 193-200 , available online at www.ejeg.com

model to assess the nature of the provided service through the development of e-commerce in SME's. They distinguish four stages 1) presence, 2) portals, 3) transactions integration and 4) enterprises integration and identify the organizational characteristics that facilitate their development and the external characteristics that function as barriers. All of these models can be criticised for offering no guidance to actually addressing technology aspects.

A basic premise underlying stage models is that descriptive stages can potentially be used in a prescriptive manner. Nolan (1979) suggests that the model can be viewed as a learning model where stage adoption is influenced by the environment and the adaptation to the environment. The applicability of stage models in general is described by Prenanto, McKay and Marshall (2002, 2003). They have identified four useful applications of stage models for e-business: 1) guidance and direction in architecture development, 2) reduce complexity of the progression of e-government initiatives, 3) communicate changes to the rest of the organization and 4) provide milestones to evaluate and control cost of architecture development. Descriptively, stage models may help to describe and evaluate the organization's maturity and sophistication in its use and management of the IT resource, for the purposes of enhanced and shared understanding. Prescriptively, it may assist the management in formulating an appropriate strategy to pursue their organization's e-business objectives (Prenanto, McKay and Marshall 2002).

Stage models are premised on the idea that organizations pass through the notional stages of maturity or sophistication. The existing growth models are primarily focused on service provision to citizens and business, and not on helping to structure the information architecture's maturity process, i.e. how the architecture should be arranged and evolve over time. As such, public decision-makers and managers are in need for support concerning the evolution of their information architecture. Hence, in this paper we address this need by deriving a stage model for the development of the architecture for e-government activities.

2. Architectural stage model development

The term information architecture, or architecture for short, lacks a universally accepted definition (Ross, 2003). Architecture is often referred to as a kind of city plan, containing detailing policies and standards for the design of architecture layers. The idea of breaking up the system into subsystems, layers and connections that is central to architecture, makes it therefore possible to design, construct and maintain a system by more than one person separately and to re-use useful parts of the architecture. This city plan concept has therefore given birth to a breed of IT architects who develop detailed drawings of the interaction between systems. A definition of architecture proposed by IEEE is: "the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution" (Hillard, 2000). The purpose of architecture is to effectively align the organizational strategies with their business processes, information systems and the coordination of their resources. Appropriate architectures can facilitate the changes required to progress to a modern service-led organization.

Architectures are typically classified into stages to distinguish various phases of development (Prenanto, Marshall and McKay 2002, Layne and Lee, 2001). *Classification* is the separation of things into groups and is often considered useful because it improves the ability to communicate or, in other words, to process data (Cook, 1996). A fundamental concept for classification is *discontinuity* (Cook 1996). Discontinuity helps to find the boundaries for groups of things that are alike. Criteria for classification depend on what you are classifying. We wanted to classify various architectures found in local governments that are primarily shaped by the coordination of the interactions between information systems residing in the back and front office. The front office (FO) comprises business processes of an organization that interact with outside entities, i.e. citizens and businesses. The back office (BO) comprises all business processes that do not directly involve customer-interactions, often settlement and related processes. We found discontinuities in the governmental architectures by locating various forms of coordination between back and front office systems, i.e. by analyzing how the interdependencies between information systems in front office and back office are managed. We found that the architecture necessary for service delivery to citizens and business evolves in five stages.

3. Architecture stage model

3.1 Stage 1: No integration

Before the advent of the Internet the information systems of most governmental agencies were hardly integrated. The first step that organizations take in order to develop online services to citizens, is the development of a website. In the beginning this site is usually set up by and the responsibility of the communications department. Layne and Lee (2001) describe the services delivered in this step in their stage model as the 'cataloguing' services, where this website comprises an overview of useful information.

From a quantitative perspective, a lot of the services towards citizens concern the delivery of information. The 'business case' of electronic business here is the replacement of personal contact at the desk or telephone by information delivery through the Internet. Practically, this means that an electronic coupling between information systems is not relevant yet. The web applications and data are 'stand-alone' applications and there is no need for exchange of data.

Bovens (2003) and Layne and Lee (2001) recall the governmental duty for *universal access*, which means that the public services should also be available through more 'old-fashioned' channels for citizens that are not yet online. A first step in delivering services online in addition to other channels is through copying the data one-by-one from one system to the other manually, thus without integration of systems. This should not be a problem for small quantities of data, but when the services are expanded, this practice cannot be scaled to accommodate larger quantities.

3.2 Stage 2: One-to-one integration architecture

A second discontinuity concerns the creation of a one-to-one messaging architecture as schematically depicted in figure 1. Data stored in back office systems are automatically published on the website and e-mail or web forms are used for communication. Although the technical demands in this stage are not yet high, this architecture does have important implications for the business practice and the responsibility for the information. Technical issues that surface are the location of resources, maintenance and how to assure privacy; organizational issues that should be addressed in this stage are also the coordination and planning of resources, and the answering of e-mails (Layne and Lee 2001). Thus, already at this stage, a deliberate policy for electronic service delivery needs to be developed.

On the technical level, message adapters are necessary for getting the data out of the applications and routing it to another application. *Adapters* are layers between the message broker, the middleware and the application, hiding the complexities of the interface (Linthicum, 2004). When software systems are becoming increasingly complex, more than one developer will develop the systems and different programming languages, applicatons, protocols and other standards might be used. Furthermore, when two systems need to 'talk' to each other, a common language is necessary, a protocol for communication and a means of delivery of the message. To enable communication between information systems middleware is used. *Middleware* is aimed at encapsulating the implementation details and supporting the smooth integration of systems (Linthicum, 2004). For this purpose middleware provides generic services such as naming and directory service, message transmission and transaction processing monitoring (e.g. Fan, Stallaert and Whinston 2000).

One-to-one messaging usually starts on a small scale. First one or a couple of services require the integration of information systems using middleware technology. A direct link to one or more BO applications is created to enable online transactional services. Newly added online services are added to the system as a direct link between FO and BO systems. Step by step, more applications are added to the online one-to-one service provision. After a while the increasing number of connections creates a complex system of applications and middleware solutions. Especially when transactions need to be facilitated by the systems implemented, the organizational and technical complexity increase at an accelerating rate (Rao, Metts and Monge, 2003). Also, in this step, the question of who owns the systems and the information in the system becomes more important and, the question of who is responsible for the information and the information quality becomes crucial.

The maintenance of this 'spaghetti' of systems is expensive, as too many solutions become into existance and often systems' knowledge gets lost over time. As a result, the information architecture becomes less and less transparent. It is no longer clear what the impact of a modification in one system will be on the other systems, as all applications are linked. One bug might even disrupt the whole system. The middleware solutions have become the legacy systems of the future. When the amount of data that needs to be exchanged increases and integration of the many existing systems becomes too complicated, a new stage of growth enabling the decrease of the number of links is called for.

 ISSN 1479-439X

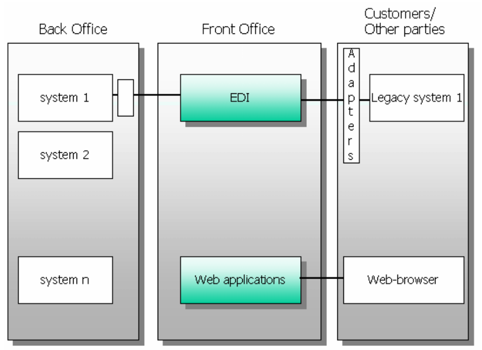

Figure 1: One-to-one integration architecture

3.3 Stage 3: Warehouse architecture

The number of connections between systems has increased and cannot be easily maintained anymore. Therefore, organizations are looking for new types of coordination. The Internet is an additional channel and creates the need for multi-channel approaches. These approaches create the need for collecting and storing integral customer information that can be used in various channels to ensure coherent customer responses (Diepen, 2000; Simons & Bouwman, 2005). The discontinuity we found is that information is decoupled using a data warehouse. Data coming from various sytems is imported into a data warehouse that provides an overview of and single point-of-access for all data, as schematically shown in figure 2.

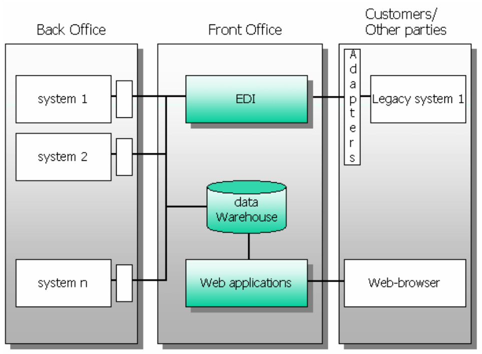

Figure 2: Warehouse architecture

The concept of using a *data warehouse* for electronic service delivery concerns the design of a dedicated database that is installed in the FO, which is filled with copies of relevant data from other databases. This database is thus maintained separately from the organization's operational databases and it stores data only for a specific duration. These data are often only used to support tasks performed in the FO. The BO applications retain ownership of the data that cannot be mutated directly from the FO. This makes the system very robust and viruses cannot attack the BO in this way. Moreover, vulnerability is reduced, as data is stored redundantly. It is stored in at least two different places. To ensure that the data are accurate and correct, changes to the data should be made in all places, preferably at the same time.

One important advantage of a data warehouse is that it facilitates the processing of large queries, because the system is not used by other applications at the same time and long response times will not block user queries. The copying of data from the BO is usually done outside working hours in order to relieve the system from large queries that need to be performed during daytime. In this way, more contact with clients can be processed directly by the FO, which performs its tasks more demand-driven than before when one-to-one messaging was in place.

Another advantage is that the exchange of data takes place on the level of databases. Many databases have standard support for making connections with other databases. This ensures that only specific fields of the databases need to be connected and that data is exchanged automatically. Finally, internal employees often welcome the installation of a central database, as it can be used as the one-stop source of information.

Data warehouses are often criticized for containing incorrect data. Organizational decisions have to be made: who will be responsible for collecting and maintaining chunks of data. For example, multiple systems contain name and address information that might not correspond. It should now be made clear which system holds the correct information, i.e. is appointed a vital record registration and will be used as the source for the data warehouse, and who will be responsible for the information and information quality and maintaining the systems.

Customer requests are processed using the data stored in the data warehouse. There are two different ways to deal with changes in data. First, customer interactions are stored within a separate transaction database and transported to the BO on a periodically basis. When other organizations request information, the information might not be up-to-date, e.g. a customer address change is only updated the next morning. As a result, this type of architecture solution is often an in-between solution as it can be created quickly and the drawbacks result in the next discontinuity.

3.4 Stage 4: Broker architecture

The more information is exchanged between agencies, the more real-time information exchange becomes necessary. The 24-hours-7-days-a-week economy requires also the continuous updating and availability of systems. The need for real-time information exchange results in a *broker architecture* were all nodes are directly linked to the *message broker* that takes care of the translation of messages, routing them to their destination and so on. This is schematically depicted in figure 3. All applications are thus connected to one central broker. The broker is used as a *one-stop shop* for all connections with information systems inside and outside the organization.

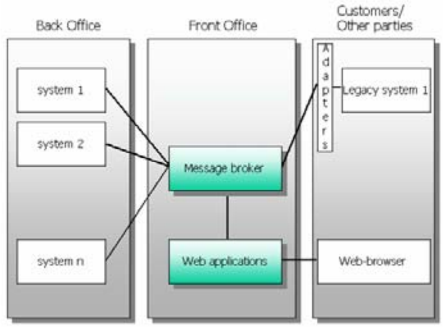

Figure 3: Broker architecture

Broker architectures fulfill the business process re-engineeering (BPR) principle of entering data only once at the source, as proposed by Hammer (1990). The most efficient way to connect information systems is through a central point, the broker. In this way, each information system needs only one connection to that central point. Thus, when a network consists of n participants in total, only n relationships have to be managed. Without a central broker, each system has to manage (n-1) relationships and n*(n-1) relationships have to be managed in the entire network, thus increasing the complexity (e.g. Janssen, 2004). When looking at the management of the number of relationships, it is clear that the use of a central broker is preferable.

A central point has advantages for the speed and costs of implementation of an information system, for the easiness and flexibility of modifying connections and for the costs of maintaining it, as only one connection needs to be implemented. When more connections with information systems are made, a central point can profit from the economics of scale principle and the experience to ensure secure, reliable and fast transmission of data. A major disadvantage is the creation of a single point-of-failure in the information architecture. This can be overcome by having redundant versions of the broker, preferably on physically separated locations in order to use different network connections.

A message broker supports different integration methods and consists of a combination of several integration technologies. Adapters can be part of the broker or part of the application connected to the broker and is therefore not depicted in figure 3. For a lot of programs, like SAP or Oracle, standard adapters are readily available.

More and more broker functionality and information can be packaged up and accessed using web services protocols (Janssen & Wagenaar, 2003). Web services enable a Service-Oriented Architecture (SOA), an architectural paradigm according to which application functionality is not provided by large monolithic information systems, but by means of web services (Linthicum, 2004). The services-oriented paradigm offers many benefits to enterprises, and the creation of a class of enterprise services allows us to create services that are modular, accessible, well-described, implementation-independent and interoperable (Fremantle et al., 2002). The combination of the broker architecture enabling a real-time exchange of information and the use of service-oriented architecture results in the next growth stage.

3.5 Stage 5: Orchestrated broker architecture

Currently, pleas have been made for more open, flexible and adaptive architectures constructed of relatively small components, which can be configured to support a limited number of functions (e.g. Fan, Stallaert and Whinston 2000; Fremantle et al., 2002). Over time, more and more functionality has been added to message broker architectures, including component and web service invocations and routing of messages; even complicated workflow rules are added. This results in a discontinuity where not only information exchange between systems, but also the invocation of information system functionality and management of the sequence of invocations become part of the broker architecture. As such, the paradigms of enterprise application integration (EAI) and workflow management (WfM) merge slowly (e.g. Linthicum, 2004; Gortmaker, Janssen & Wagenaar, 2005). This depicts a trend to include business logic as part of the architecture. This is a natural extension of the broker architecture enabled by developments in service-oriented architectures and services technology. The broker architecture becomes gradually an orchestrated system of both technical and business functions. This architecture, called orchestrated broker architecture, is schematically visualized in figure 4. Business or workflow rules are depicted in a repository, as this enables easy maintenance and re-use

Web service orchestration is enabled by developments in the web services technology stack (http://www.w3c.org). The de facto standard for web service orchestration is the Business

Process Execution Language for Web Services (BPEL4WS), or BPEL for short. Orchestration using BPEL aims at coordinating the time-dependent sequence of single web service invocations. In this way a series of complex business processes can be created. Web service orchestration contains the business logic for managing the sequence of service invocations and requests. The orchestrator can also guides the business process or workflow of users, i.e. which screen will appear in the web browser of a user and which functional components and information sources need to be accessed.

Service-oriented architectures enable the on-demand composition of new business processes using already existing web services. Internal departments or external service providers can provide web services. This enables the connection to outside systems, to access information or invoke functionality. This the architecture has gradually expanded from coordinating back and front office applications to coordinating business processes and interactions with external systems.

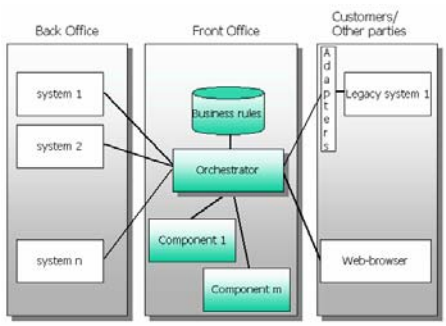

Figure 4: Orchestrated broker architecture

This architecture is based on high-levels of standardization of interfaces, business rules and protocols. It provides many benefits, including the re-use of components as services, the easy construction of new processes and might even enable the re-use of complete processes by other public organizations. In the higher stages the emphasis gradually shifts from technology to organizational processes, structures, and the socio-political issues. The higher stages need business process engineering and structural transformation.

4. Conclusions

In this paper we have presented an information systems management model based on stages of growth. We derived a five-stage growth model based on the fundamental concept of classification; discontinuity. We looked at the discontinuities in the information architectures connecting the information systems in the back and front offices of local governmental organizations. Our stage model consists of the following five stages: 1) no integration, 2) one-to-one messaging, 3) warehouse, 4) broker and 5) orchestrated broker architecture. In the first stage no integration exists and information is copied manually. When the number of services increases, the process of copying manually becomes too time-consuming and electronic links are set up. In the second stage an electronic connection or coupling is set up for each service to be delivered separately. Automating many services results in'spaghetti' of links, which become difficult to maintain when the number of connections between systems increases. When the spaghetti of systems becomes too intransparent, the third stage can be found; the data warehouse architecture. A data warehouse is a database containing information that is copied from various systems. The information in the warehouse is sometimes incorrect and not up-to-date as information is periodically imported. Therefore, the need for real-time data exchange appears, resulting in the fourth stage: the broker architecture. A broker is a central point for information exchange that passes on information between the different information systems in real-time. Over time, the information broker not only handles information, but also starts invoking other types of technical services. In that case, the last growth stage, the orchestrated broker architecture, is entered. Business logic is included in the information broker to create workflows and even complete business processes. Our growth stage model enables the gradual expansion from no integration architecture, via an architecture coordinating back and front office applications, to an architecture coordinating complete business processes and interactions with external systems. In the higher stages the emphasis gradually shifts from technology to organizational processes and structures. The number of stages is limited by the current practices we have found at local governments. Moreover, we found that some organization have concatenated two stages into one stage. The investigation of other practices might result in other stages; moreover, in the future the number of stages might be expanded based on technology developments. New developments might result in discontinuities, which in turn result in new growth stages. Current trends that might be included are the use of ontologies, semantic web services, software agents and peer-to-peer architectures.

Stage models are built on the assumption that developments of IT systems evolve through a number of stages of growth. As an organization becomes more familiar with the use of technologies it advances to a higher stage. Technologies in themselves will not likely cause changes to happen. This implies the need for a certain amount of organizational change strategies addressing issues like creation of shared vision, motivation of stakeholders, dealing with resistance to changes, obtaining political support, planning intervention strategies, reallocation of responsibilities and so on. Our stage model does no address these issues but can be used to plan for change to establish goals and determine progress towards accomplishing these goals. The model can be used descriptively to assess the maturity and sophistication of current architectures. Prescriptively, it may assist public managers in formulating an appropriate strategy to pursue their organization's objectives. Moreover, the stages can be used by public decision-makers as a guidance and direction for architecture development, to reduce the complexity of the progression of e-government initiatives, to communicate changes to the rest of the organization and to provide milestones to evaluate and control cost of architecture development.

References

Bovens, M. (2002) De Digitale Republiek: Democratie en Rechtstaat in de Informatie-maatschappij. Amsterdam University Press, Amsterdam, The Netherlands.

Bruijn, H. de, Heuvelhof, E. ten & Veld, R. in 't (2002) Process Management. Why Project management Fails in Complex Decision Making Processes, Kluwer, Boston/Dordrecht, The Netherlands.

Cook, M.A. (1996) Building Enterprise Information Architectures. Reengineering Information Systems. Prentice Hall, New Jersey, USA.

Diepen A.M. van (2000) "Multi-channel distribution in financial services: Impact of electronic distribution channels on the internal organization", Trends in Communication, vol. 6, pp. 37-60.

Fan, M., Stallaert, J. & Whinston, A.B. (2000) "The adoption and design methodologies of component-based enterprise systems", European Journal of Information systems, vol. 9, pp. 25-35.

Fremantle, P., Weerawarana, S. & Khalaf, R. (2002) "Enterprise services. Examine the emerging files of web services and how it is intergrated into existing enterprise infrastructures", Communications of the ACM, vol. 45, no. 20, pp. 77-82.

Ghasemzadeh, F. & Sahafi, L. (2003) "E-Commerce Adoption: A Two Dimensional Maturity Model" 4th World Congress on the Management of Electronic Business, McMaster University, Hamilton, On., Canada. January 15 – 17.

Gortmaker, J., Janssen, M. & Wagenaar, R.W. (2005). Towards Requirements for a Reference Model for Process Orchestration in e-Government. TED Conference on e-Government, TCGOV2005, Springer Verlag LNCS 3416, pp. 169-180.

Greiner, L.E. (1972) "Evolution and Revolution as Organisations Grow", Harvard Business Review, vol. 50, mo. 4, pp. 37-46.

Grover, V. & Segars, A.H. (2005) "An Empirical Evalution of Stages of Strategic Information Systems Planning: Patters of process design and effectiveness", Information & Management, vol. 42, pp. 761-779.

Hammer, M. (1990) "Reengineering Work: Don't Automate, Obliterate", Harvard Business Review, vol. 68, no.4, pp. 104-112.

Hiller, J. & Bélanger, F. (2001) Privacy Strategies for Electronic Government. E-government Services, Arlington, VA, USA.

Hillard, R. (2000) IEEE-std-1471-2000: Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE, http://standards.ieee.org/.

Janssen, M. (2004) "Insights from the Introduction of a Supply Chain Coordinator", Business Process Management, vol. 10, no. 3, pp. 300-310.

Janssen, M. and Wagenaar, R. (2003). From Legacy to Modularity. A roadmap towards modular architectures using web services technology in e-government. Second International Conference, EGOV 2003, Prague, Springer Verlag, LCNS 2739, pp. 95-100.

Janssen, M., Wagenaar, R. & Beerens, J. (2003) Towards a flexible ICT-architecture for Multi-channel Service Provisioning. Hawaii International Conference on System Sciences (HICSS-36) Hilton Waikoloa Village, Big Island, January 6-9.

Layne, K.J.L. & Lee, J. (2001) "Developing fully functional E-government: A four stage model", Government Information Quarterly, vol. 18, no. 2, pp.122-136.

Linthicum, D.S. (2004) Next Generation Application Integration: From Simple Information to Web Services, Addison-Wesley, Reading, Massachusetts, USA.

Moon, M.J. (2002) "The Evolution of E-Government Among Municipalities; Rhetoric or reality?", Public Administration Review, vol. 62, no. 4, pp. 424-433.

Nolan, R.L. (1979) "Managing the crisis in data processing", Harvard Business Review, vol. 57, no. 2, pp. 115-126.

Rao, S. S., Metts, G. & Monge, C. A. M. (2003) "Electronic commerce development in small and medium sized enterprises: A stage model and its implications", Business Process Management, vol. 9, no. 1, pp. 11-32.

Reddick, C.G. (2004) "A two-stage model for e-government growth: Theories and empirical evidence for U.S. cities", Government Information Quarterly, vol. 21, pp. 51-64.

Simons, L. & Bouwman, H. (2005). Multi-channel service design process. Challenges and solutions. International Journal of E-business. Vol.3, No.1, pp. 50-67.

Zhao, J.L. & Cheng, H.K. (2005) "Web services and process management: A union of convenience or a new area of research?", Decision Support Systems, vol. 20 No. l, pp. 1-8.