Towards a Flexible ICT-Architecture for Multi-Channel E-Government Service Provisioning

Marijn Janssen and René Wagenaar
School of Technology, Policy and Management,
Delft University of Technology,
Jaffalaan 5, NL-2600 GA, Delft, The Netherlands,
Tel. +31 (15) 278 1140/8077, Fax. +31 15 278 3741,
E-mail: MarijnJ@tbm.tudelft.nl, Renew@tbm.tudelft.nl

Jaap Beerens
D3K Simulations and consultancy,
Delftechpark 26, NL-2628 XH Delft, The Netherlands,
Tel. +31 (15) 2600952; e-mail: Jaap@d3k.nl

ABSTRACT

The planning and subsequent nationwide implementation of E-government service provisioning faces a number of challenges at the level of municipalities in the Netherlands. Initiatives are confronted with a highly fragmented ICT-architecture that has been vertically organized around departments and with hardly any common horizontal functionality. This situation is even further enforced by a defacto duopoly on the software market of information systems used by municipalities. The provision of services over web-based channels leads to a need for a more flexible, open ICT-architecture based on standardized elements.

The goal of the research presented in this paper is to determine the feasibility of a component-based approach to meet the aforementioned challenge for a more flexible, open ICT architecture. The research consisted of two parts (1) the identification of opportunities for generic components in the ICT-architecture of municipalities and (2) supporting the evaluation of these opportunities using simulation.

1. Introduction

Citizens in all countries are calling for better services at lower cost and for more responsiveness in a dynamic and continuously changing political, economic, societal and technological (PEST) environment. Public administrations should stay closer to citizens' every-day life, and act more proactively rather than reactively [19]. Governmental organizations are challenged to provide more customer-oriented products and services. Customers can be targeted through multiple channels, such as web-based, call centers and physical offices in the municipal hall. In order to exploit these channels in a coherent, efficient and effective way, the need to restructure administrative functions and processes is clearly felt to support coordination and cooperation between different departments. Legacy information systems within governmental organizations, however, often restrict the development towards new customer-oriented processes.

In general, the current situation is such that each governmental organization has developed its own information systems in rather isolation, and that for each product or service a separate information systems exists. The information systems are often monolithic packages. No generic architecture is available that enables communication between front-office and back-office applications, between back-office applications or with systems outside the own organization.

Beneath being monolithic packages enterprise information systems have been criticized that they often impose their own logic or business process view on a organization and lack flexibility and adaptability in today's dynamic environment [9]. Currently, pleas have been made for more open, flexible architectures constructed of relatively small components, which can be configured to support a limited number of functions [9]. Although various private companies already use such component-based systems, governmental organizations are relatively slow in adapting such approaches as they lack sufficient insight into the pros and cons of such an approach. So, much effort has to be put in communication

Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)
0-7695-1874-5/03 $17.00 © 2002 IEEE

of the implications in order to get management commitment and to support decision-making. Gaining commitment is further complicated due to the large number of stakeholders that are involved such as politicians, process owners, information managers, ICT-departments, administrative departments etc. Some stakeholders might have a natural resistance against or not trust new initiatives, some might have too limited knowledge for decision-making or lack experience with ICT and stakeholders might be interested in different aspects like advantages for the operational processes or for the customers or in ICT aspects.

In this paper we will investigate the opportunities for a component-based approach for E-Government service provisioning and support the evaluation of such an approach by decision-makers. We will choose simulation as a communication vehicle for evaluation such an approach, as it allows to understand the essence of business systems, to identify opportunities for change, and to evaluate the effect of proposed changes on key performance indicators [16].

The *scientific relevance* of this research is embedded in the issue of fit and alignment between an organizational context and information systems. The fit between an IS and the organizational context is believed to be critical to the success of the business [10]. Simply aligning IS to fit the existing organization is a too narrow view, as innovation in ICT might result in new organization structures. For example the Internet enables new interactive, customized distribution channels. Fan, et al. [9] argue that measuring the business value and analyzing variations in organizational performance from the interactions of IS and organizational contexts will make significant contributions to IS theory as well as practitioners. Using simulation we support the alignment of applications with business processes and provide insight into the potential benefits and drawbacks.

The structure of the paper is as follows. First, the problem domain is explored and discussed in the context of a project conducted for the information-managers council of the group of Dutch municipalities with over 100.000 inhabitants. Next, we discuss new developments in ICT architectures and the role of component based developments therein. Based on this discussion, opportunities of using generic ICT components are identified as a result of a thorough analysis of the process of handling requests for driver license renewal and construction building allowances within municipalities. Section five elaborates on the use of simulation as a tool to provide insight into the pros and cons of a component-based approach to decision-makers. Simulation was used as a vehicle to visualize both the dynamics of business processes, and the interactions between components, legacy systems and users. Models of the existing 'as is' situation and 'to be' situation with components were developed in order to provide detailed insight into the consequences of a new supporting ICT architecture. In sections six, the evaluation of the architecture by the information managers' council using the simulation models developed is discussed. Finally, section seven summarizes the main conclusions.

2. Case: AnalysePilot

The Dutch municipalities are free to design their information architecture and to choose appropriate software vendors. Often there is no central management and departments can buy their own applications for each process. As a result, municipalities have a highly fragmented ICT-architecture, consisting of legacy systems for each product they offer. These systems are often monolithic packages and thus extremely difficult to reconfigure and to integrate with applications developed by other vendors and to provide access to new distribution channels. In short, there is an *interoperability* problem between applications within a municipality, but also between municipalities.

The VNG (in Dutch: Vereniging van Nederlands Gemeenten), the Dutch association of municipalities has launched a number of initiatives to develop *communication standards*. The most important initiative in this respect is the creation of standards for the GBA (in Dutch: Gemeentelijk Basis Administratie). The GBA is the Dutch authentic registration of data about all residents living in a particular city or village. Many applications, such as the passport or drivers license renewal request, and address change need to use data of the GBA and consequently need to communicate with the GBA.

Apart from the interoperability problem two companies having sector-specific knowledge dominate the software market for municipalities. The software market for municipalities can as such be characterized by lock-in by this duopoly. These companies possess specialized and unique knowledge about business processes at municipalities, have developed dedicated software to support these processes and are involved in many ICT implementation projects. *Lock-in* focuses on the role of switching costs as a source of sustained competitive advantage for firms selling ICT applications [3]. *Switching costs* are created when organizations make investments that are specific to a particular seller of ICT. This can include sunk investments in hardware, software, user training and organizational changes, as well as non-technology barriers such as trust in organizational partners or long-term contracts, which are valuable as long as an organization continues to use a particular vendor. Just currently they initiated changes in their software packages to enable communication with other information systems.

Due to this dependency, municipalities have either to invest heavily for developing their own customized applications or pray that one of these companies will ship out software or customize an existing software package that will serve their needs. The duopoly tends to impose its own logic or business process view on the large municipalities. But, this view is inevitably limited as they are not forced by competition to look for better long-term solutions and thus benefit from staying close to the status quo.

To counter this lock-in situation, the information managers of the Dutch cities with more than 100,000 inhabitants have joined forces in a cross-municipality information-management council under supervision of the VNG. The goal of this council is to search for more open, flexible architectures that can support multiple municipalities. They are facing the challenge set by the Ministry of Internal Affairs to provide at least 25% of the services over the Internet by the end of 2002 [7].

The current services provision over the Internet can be positioned in the catalogues and transaction phase of Layne [15]. Overall, municipalities have a web-presence containing product information, there are downloadable forms for a limited number of products and for some products it is possible to conduct online transactions. In the latter case, most transactions are performed without any direct integration of front- and back-office applications. The stages of horizontal and vertical integration [11], characterized by integration of information systems across different functions and departments, are still far away.

The cross-municipality information managers council initiated the "*AnalysePilot*" aimed at developing a reference architecture that should provide guidance for the development towards a component-based architecture. Such an architecture should not only bring online the 290 products currently provided in the municipalities' portfolio, but also support existing distribution channels. One of the goals of this project was to support management in their decision-making about the potential of a component-based architecture. When this decision-making turned out to be positive, a follow-up project should to be started for developing a prototype of such an architecture to prove the concept in practice and to develop standards for ensuring interoperability between components and legacy systems.

3. ICT architecture and components

The existence of isolated, overlapping in function and content, highly fragmented and unrelated computerized applications within the same public organization has resulted in a major *interoperability problem* and has led to 'isolated islands of technology' while information systems were viewed as being internal to the public organizations [23]. As technology continues to evolve at an accelerating rate, nontrivial hard- and software will remain diverse and heterogeneous. The various legacy applications for the 290 products offered by the municipalities have grown over time, meaning that hardware and applications purchased years ago *must* interoperate with those purchased today and tomorrow. Driving factors such as multi-channel support, the need for communication between back- and front-office and with external systems, and the call for a more customer-oriented organization explain for a large part the existence of such a heterogeneous ICT-architecture. There is a need for a more interoperable and flexible ICT-architecture that can more efficiently and effectively meet these future requirements.

ICT-Architecture is the description of the set of components and the relationships between them 0. Architecture can be viewed at various levels, including hardware, network, system, application and enterprise level. This research focuses on the application (software) architecture level where a component-based software architecture is positioned. The relationships with the other architectural levels should also be taken considered for a complete architectural picture. This will be done at a higher level of abstraction, with the exception of the enterprise level since the relationship between components and business processes is crucial and therefore needs a more detailed analysis.

Component-based system development is rooted in the object-oriented modeling approach [9]. Objects are straightforward abstractions of real-world entities and have properties such as encapsulation, inheritance, and polymorphism. Objects greatly increase software reusability and simplify the software development process [9]. The component model, on the other hand, focuses on building information systems by combining and matching pre-developed software objects, i.e. components. The focus is not on the properties of objects but on the combination and integration of different software components. The manageability increases as large components can thus be constructed from smaller components. In essence, a complex problem is split up into smaller problems, which can be solved independently. Each single component can be replaced by another component without affecting the others. Components can run from different computer platforms and interact with each other through standardized interfaces. The main characteristics of components for our purposes are the following:

- Each component consists of a bounded and unique functionality;
- Each logical component should be translatable into a software component;

- A component is always triggered by a request and always sends a response;
- A component has a standard interface and the implementation is encapsulated;
- A component should be reusable in various processes that provide a specific product or service and be sufficiently generic to support various Dutch municipalities.

Components can communicate with each other directly or through *middleware*. The advantages of using middleware over direct interaction are that fewer connections need to be established and maintained and that changes need only be made at one place in the overall ICT-architecture [12]. Middleware already exists for some years in various forms such as message queuing systems, Common Object Request Broker Architecture (CORBA), Microsoft's Component Object model (COM), Java 2 enterprise Edition (J2EE), and recently web-services. A *Web service* is a software application identified by a URI (Uniform Resouce Indentifier), which interfaces and binding are capable of being defined, described and discovered by XML artifacts, and that supports direct interactions with other software applications using XML based messages via internet-based protocols [27].

The argument for middleware is that by enabling interoperability and portability between components, it prevents lock-in and allows users to select components based on other criteria such as quality, costs, functionality etc. The basic idea is that municipalities can purchase components from different vendors or develop more specialized components reusable by other municipalities in order to assemble a highly customized information system. The development of such specialized components could be coordinated centrally by the VNG, the Dutch association of municipalities. Ideally, it will stimulate more specialization and thus more competition in the software market. Municipalities can then buy components from the market that best fist their needs and customize them if necessary to fit seamlessly within their processes. In practice, this ideal situation might not occur when vendors pay lips service to standards while still hooking customers into lock-in. Even vendors that adhere faithfully to standards are often forced to introduce proprietary features to cover areas the standards do not address [24]. In this respect there might be a huge role for the VNG.

Reducing information complexity is a significant challenge for a component-based approach. At the price of loss of flexibility, component designers need to establish reasonable default settings that make reconfiguration unnecessary for most cases. As hardware performance has increased, so has the capacity to tune middleware through configuration rather than through programming [24]. By separating development from deployment issues, maintenance costs can be lowered and the need for specific knowledge at municipalities.

Commercial Off-the Shelf (COTS) components can be used to reduce software development and maintenance costs, as well as reducing software development time by bringing the system to market as early as possible [14]. COTS components replace the traditional scenario of developing unique system components with the promise of a fast, efficient acquisition of cheaper components and implementations.

In a component-based approach existing information systems can be wrapped so they behave like components. *Wrapped* means that existing systems are surrounded with new interfaces [5]. The wrapped systems act as a server to perform functions required by an external client, often middleware, which does not need to know how the system implements the service. Over the years, the legacy system might be replaced in its entirety by a component using the same interface and request-response protocol. In this way a smooth transition to a complete component-based architecture might be made.

4. Opportunities for components

Process analysis of the existing processes at a number of municipalities was used to determine which type of components could be suggested that are necessary to support these processes. We focused our process analysis by modeling the tasks for one particular process, the renewal of driver licenses. Component functionality can then be derived from those tasks by investigating how one or more tasks could be supported. We start with describing the current physical and future on-line business processes involved in driver license requests. Thereafter we will look how ICT can support this process by mapping components on these processes. Finally we will discuss the functionality of the components.

A *business process* is a number of interdependent tasks and decisions and can be described using various modeling techniques [6], [26]. *Process modeling* stresses the tasks of actors acting in the business process under consideration [25]. This kind of model emphasizes the actor's coordination of tasks. The description of the current tasks involved in the request for drivers' licenses is shown on the left side of figure 1. Note that this is an *abstraction* of the drivers' license renewal process performed at various municipalities. Each municipalities has it's own logic to perform this process. The process is aborted when citizens do not have the required official forms with them, such as the old driver license and passport photos in case of renewal, or the license is denied due to a number of offences. A detailed

description of the processes performed at various municipalities can be found in Beerens [4].

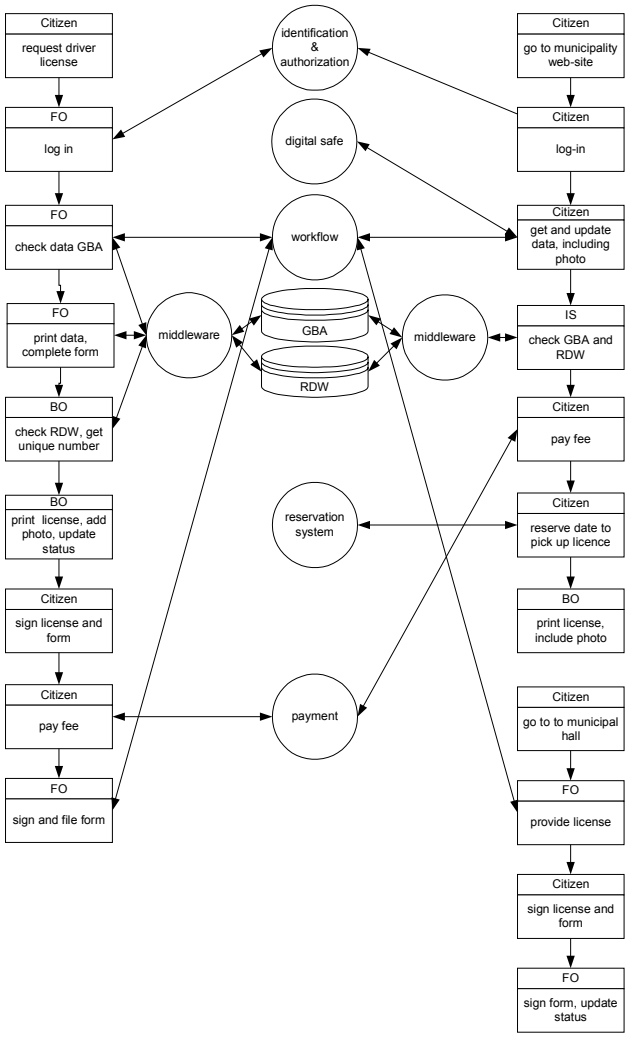

Figure 1. Mapping of components on processes

In the current process a citizen goes to the counter of the town hall and requests a (new) driver's license. The front-office employee, abbreviated as FO, asks for the old drivers license and other official papers and logs-in on the legacy information system. The FO requests data about the citizen from the GBA. The GBA contains registration data such as name, address, date of birth, sex, nationality etc. The FO checks the information in the old driver license or other official papers with the information in the GBA. When the information is found to be valid the FO prints the data on paper and fills in additional information from the existing driver's license. The FO gives this form to the back-office employee, abbreviated as BO. The BO enters the data on a terminal with is connected to the information system of the RDW (in Dutch: RijksDienst voor Wegverkeer). The RDW is responsible for the central nationwide registration of drivers' information of persons living in the Netherlands. The RDW checks the data with the data in its own system. When approved the RDW sends a message to the BO containing a new unique drivers license number. The BO can use this unique number to print the information on a pre-printed drivers license. After printing the BO adds the passport photo on the license and brings the license to the FO. The FO employee checks the information again, asks the citizen to sign for acknowledge of receipt, asks for payment and after payment signs the delivery receipt and files the information.

The CB architecture under study should align the structure of the municipalities' processes to the demand for services for current ('as is') as well as potential future ('to be') situations. The future situation should allow for the on-line request for a drivers' license using the Internet. The tasks for the on-line request handling are shown on the right hand side of the figure. A citizen goes to the municipality web-site and he/she logs-in and enters data, including digital photo, or gets the data from a digital safe. The information system (IS) automatically checks the entered data with the GBA and RDW. When the request is accepted the citizen is asked to pay the fee and to reserve a date to pick up the license. The BO gets all the necessary data, prepares the license and stores the license so it is available for the FO. On the reserved date and time-slot the citizen arrives at the FO, signs the license and takes it with him/her.

Next, the processes in the 'as is' and 'to be' situation were further analyzed to identify components. In literature a number of component-based development methods can be found [8][11]. These methods did not fit our purposes of identifying best-of-breed component as they (1) view components as a piece of software code, (2) do not provide guidelines for determining the granularity of components, (3) do not take the complexity of the existing situation sufficiently into account and (4) are too complex to be suitable for communication to stakeholders. Present methods treat components mainly as implementation and deployment artifacts, rather then linking them to the business processes and taking COTS components as a starting point. The communication aspect was of vital importance as, decision-makers often lacking ICT knowledge wanted to stay involved in the project and understand the steps taken due to the possible high impact on the ICT-architecture of municipalities in the Netherlands. As such we had to make the relation directly to the task diagrams already derived, as the actors involved had already accepted these task diagrams. The consequence of this starting point is the interference between the process and component view as shown in figure 1. Although this leads to a less formal representation, we found it to be easily understandable by the participants.

At each task we looked how components could support this task and which functionality was necessary. To identify components the characteristics of component discussed before were used as requirements and a number of intuitive appealing design guidelines were used. Design guidelines include (1) information should be captured only once at the source and re-used by other components, (2) communication between components should be kept to a minimum and (3) there should be a central process control component integrating business process steps with functionality provided by components [12]. Below, we give a brief overview of the components we found to be relevant and sufficiently generic to support other services.

1. Identification and authorization

With the identification component citizens or employees in the BO and FO can identify themselves in order to get access to the other components. Identification can be implemented using a user name and password, but also using a chip card or biometrical method. Other components can ask the identification component for *authorization* data. The *authorization*, the provision of access to the functionality of a component, stays within each component itself. E.g. when a user wants to get data from his digital safe, the digital safe component sends a request for information about the identification to the identification and authorization component. The digital safe component decides to permit access based on the information received from the identification and authorization component. The advantage of this construction is that the identification and authorization component does not need to store all data about authorization rules per component. In this way cumbersome maintenance and control procedures can be avoided. This follows our design guideline to store data at the source and re-use it in other processes.

Authorization is not only dependent on the role of a person, but might also be dependent on the level of identification. For example, somebody identified using biometrics might get more access rights (e.g. update GBA) than somebody identified using only a username and password.

2. Digital safe

The digital safe is a component for citizens to store and reuse personnel and private data. The citizen has authorized access to the data in this component and can provide permission to others to get (some) data stored in the safe. The digital safe might contain all kinds of data, such as medical information, insurance data etc. In this way, data have to be entered once and can be reused by authoring access.

3. Workflow

The workflow is responsible for the scheduling sequences of tasks and for controlling the use of other components. It is for example responsible for first requesting an identification before proceeding to a next step and to show screens one-after the other. Moreover, this component signals delays, sends alerts and should offer other standard workflow functionality. It should also provide information about which documents are necessary for which tasks.

4. Middleware

Middleware is responsible for the communication between humans and information systems and between information systems. Middleware can be implemented using messaging, web-servers and web-services. In a general sense, legacy information systems can communicate with other systems using messaging. The communication with state-of-the-art components can be done using web-services. Human-to-computer (information system) interaction can be implemented using a web-server.

Messaging exchanges data between information systems such as the legacy systems and components without any human intervention. A messaging application translates data formats into other data formats and asynchronously exchanges data based on message queuing. When an information system is not available the messaging component can queue messages and submit the message at another time when the receiving component is available. The data formats are stored in the messaging application, so that control and maintenance of these formats can be done at one (central) location.

Human-to-computer communication proceeds using web-servers. The human interacts with the server using a client application, such as a web-browser based on a request-response protocol.

Web-services are 'middleware agnostic': rather than replacing existing middleware solutions, web-services integrate and expand the capabilities of middleware [24]. The use of Web Services is expanding rapidly as the need for application-to-application interoperability grows [27]. Web services can provide a standard means of communication among different components involved in processing dynamic context-driven information. Web-services are based on a request-response model, components can function both as a client and a server.

5. Reservation system

A reservation system can be used to let citizens schedule an appointment with a government representative. Employees can provide data about their availability and citizens can make appointments. Long queues before the offices at the municipality hall can be avoided by reserving time-slots.

6. Payment

The payment component is responsible for financial settlement of the service provision. Third parties like Bibit and BNG in the Netherlands offer various forms of payment. They can take over the full responsibility for financial settlement after obtaining the necessary data.

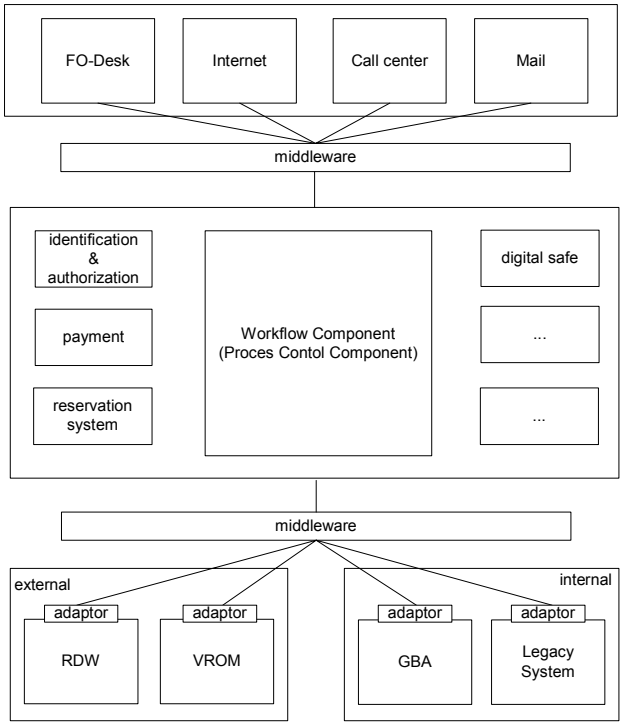

Figure 2. Application architecture

Management of shared information is an important task in component-based system development since the access control mechanism may be distributed in different systems. The relationship between information systems, including components within the applications architecture is shown in figure 2. This figure makes the distinction between back-, mid- and front-office. At the top of the figure the multi-channel approach in the front office is shown. The mid-office contains the components that can be used in various processes. The back-office contains the internal and external legacy systems that need to be accessed using adaptors. *Adaptors* are layers between the message broker middleware and the application, in this way hiding the complexities of the interface [17]. Examples of adaptors are the before discussed wrappers surrounding existing systems with new interfaces.

5. Simulation and visualization

Presenting a strong business case is necessary for gaining management support [14]. A business case should provide insight into the benefits, but also into the drawbacks and implementation pitfalls. The real importance lies in controlling expectations, focus on real design issues 0. Modeling can play an essential part in communication of the implications and benefits to stakeholders. Two main stakeholders for communication can be distinguished, the information managers and the decision-makers such as mayor and aldermen of the municipalities with more than 100.000 citizens. Communication to these two groups should be focused on providing insight into the current as well 'to be' situation with components in order to support decision-making. The requirements on modeling can be summarized as follows.

- Various architecture levels such as organization, business process, application and network should be visualized;
- Interactions and dynamic behavior between and within various levels of architecture should be visualized. Especially the interactions between components and the relationship with the business processes should be visualized;
- Provide insight into the current 'as is' as well as potential future 'to be' situations;
- Should support stakeholders to draw conclusions about the benefits and disadvantages based on the insight gained.

The *key issue* in modeling is the choice of phenomena to include and to omit [12]. Modeling should capture the requirements discussed above, but avoid presenting so much detail that it requires too much effort and scarce time of decision-makers to understand the models. This aspect is complicated, as we had to deal with multiple stakeholders having various background and kinds of knowledge. Consequently they might need different level of detail.

Simulation of business processes constitutes one of the most widely used applications of operations research, as it allows us to understand the essence of business systems, to identify opportunities for change, and to evaluate the effect of proposed changes on key performance indicators [16]. *Simulation* of tasks means that the time aspects of a sequence of tasks are modeled [18]. Essential is that simulation can be used to understand the behavior of a concrete system, to evaluate various strategies for the operation of the system, and to study the impact of scenarios representing a particular path to a hypothetic future situation [20]. The philosophy behind simulation is to develop a dynamic model of the problem situation, experiment with this model, and experiment with alternatives for the problem situation [21]. Simulation can be used to asses process design options [13]. One of the advantages of simulation is that what-if analysis can be committed without changing reality at lower costs.

Visualization is often a standard feature of a simulation language. Pegden, et al. [18] describe *animation* as a dynamic display of graphical objects that change position, shape or color on a static background. Animation of time-ordered sequences of tasks can take place against a static background, which provides the layout of an organization.

Simulation can be used as a communication instrument to stakeholders, and thus satisfied the requirements for our study.

There are a large number of simulation packages available [22]. Arena was selected as simulation package, as it matches the requirements described above and was readily available. Using Arena a simulation model of the existing situation was build. A screenshot of this model displayed in the development environment of Arena is shown in figure 3. The model shows the employees and systems in the back-office at the top of the model. The front office employees and information systems are shown in the middle, below the front office. Citizens waiting in the municipality hall are shown at the bottom. On the left of the model the existing external systems, RDW and GBA and the relationship to the FO and BO applications are shown.

When playing the simulation the interactions, such as messages send between the systems are shown. Communication time between information systems is too short to be visible in a simulation, therefore a time delay is used to visualize the movement of messages and data packages between information systems.

The following step was to build a 'to be' model including a component-based architecture and multiple channels for service provisioning. A screenshot of this model is shown in figure 4. The 'as is' model is expanded towards an open, flexible ICT-architecture for multi-channel service provisioning. The new channels are shown on the left side of the figure and the ICT-architecture based on components is shown on the middle of the figure.

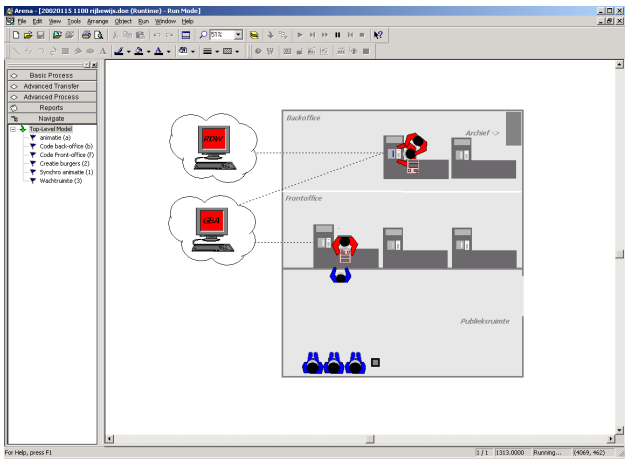

Figure 3. Screenshot of simulation model of the current situation

The simulation is aimed at showing the business case for an open, flexible component-based ICT architecture supporting multi-channel service provisioning. The simulation model shows that components and legacy applications developed at different time periods and by different software vendors can work together. So, investments in legacy systems can be protected. By just plugging in new components changes can be made more easily.

The functions of the components were simulated to ascertain that the model was independent of specific implementations. Municipalities should be able to replace any component with better ones when necessary, consequently only the functions of the components were simulated.

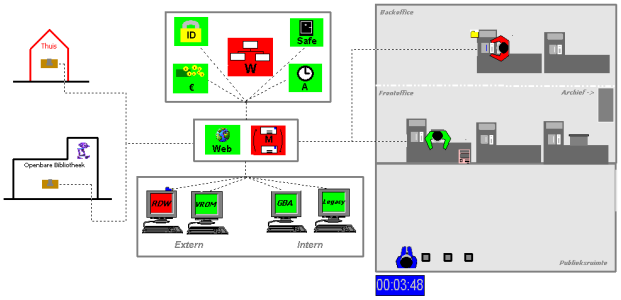

Figure 4. Screenshot of Simulation model including components

The result is that the system will always support the business processes of a municipality, rather than the other way around. The market for municipalities will be more open leading to higher competition, better products, and lower costs. In practice this should mean that components provided by vendors as Microsoft, IBM, Oracle etc. can be chosen and configured for the particular situation of a municipality. In this way the duopoly can be broken down.

Two new channels were added to the simulation, citizens using a computer at home and connected to the Internet requesting a driver's license and citizens using a terminal in the library to request a driver's license. The simulation shows the interactions over time to and from these channels using a component-based architecture including wrapped legacy systems. It shows that on a conceptual level a component-based approach is flexible and open enough to support a multi-channel approach. Probably even mobile channels can be supported using this approach.

6. Evaluation

The simulation models were presented to the information managers and management of the Dutch municipalities involved in this project. Using the simulation model they recognized the benefits of such an approach. They agreed that a component-based approach

created an open, flexible architecture because new components can be added to and existing components can be replaced in such architectures and the current process and a potential future process for renewal a driver's license using multiple channels could be supported with the components identified.

The information managers' council became also aware that it is important that a component's interface to the outside world should be clearly defined and decoupled from its implementation. Coordination of the development of standards and architecture are important aspects to facilitate such an approach. They agreed that such an approach might create a market with more competition than the current duopoly. The best-of-breed products among all vendors in the world might be chosen and ideally integrated in a plug-and play manner. There might still be a dependence on the duopoly as systems integrators with their existing systems, however, the dependence would likely be less. The information managers also became aware that making reuse cost-effective could be difficult and would be dependent on the translation of the functions to software components and the fit of components within the architecture.

The simulation models made not only plausible that a component-based architecture would be more open and flexible, the models also supported the identification of a number of *drawbacks* that would be encountered when implementing such an approach.

The *first* drawback is the wrapping of legacy system. Wrapping seems to significantly extend the life cycle of existing information systems and support the integration in the ICT-architecture. Wrapping could, however, mean that man-months are necessary to wrap single applications. Some applications might even have to be rebuilt from scratch. The costs involved to achieve this are not estimated and might differ largely per municipality.

A *second* drawback seems to be the maintenance of the ICT-architecture. Small municipalities depending on a limited number of information systems do not need such complicated ICT-architectures and have fewer problems realizing a multi-channel service provisioning approach as they only need to rebuild a limited number of systems that are available off the shelf. The maintenance gets complicated, as knowledge of more vendors per component is necessary.

The *third* drawback seems to be the deployment and configuration of components. Detailed knowledge about functionality and knowledge for configuration components for business processes and services are necessary. An unfortunate side effect of increasing the configurability of applications, however, might be that configuration can become nearly as complicated as programming [24].

These drawbacks lead to the insight that a central control and maintenance organization might be useful to bundle the knowledge about ICT-architectures, components and processes. It was decided to start an investigation in this direction. Moreover, the cross-municipality information management council in the Netherlands decided to formulate a pilot project to test the technical feasibility of a component-based approach and to estimate the costs and benefits using *prototyping*. This would result in detailed insight into data formats currently used and how to design connections between legacy applications and components. This pilot project would take place within a number of municipalities having different characteristics in order to evaluate the efforts in various situations. Decision-making about the go/no go has not taken place yet.

Important *limitation*s of the research are that we did not test the approach in practice, but limited the research to building simulation models to prove the concept. Further, we focused at the level of ICT-architecture; however, organizations should be able to respond to changing conditions with capabilities such as the right organization structure, training of employees, spotting new technology, defining transitions and implementation paths etc. We did not look at the organizational readiness for a CB approach nor did we make an analysis of municipalities as political entities to identify possible resistance against change.

7. Conclusions and future research

The goal of the research presented in this paper was to determine the feasibility of a component-based approach to meet the challenge for a more flexible, open ICT architecture for multi-channel service provisioning. We used simulation to make plausible that such an approach would be better than building large new systems and replacing old systems. More flexibility was created as (1) new components can be added to and (2) existing components can be replaced in such architectures; the simulation showed that (3) the current process and (4) a potential future process for requesting a driver's license using multiple channels could be supported with generic components. The flexibility of the architecture helps to better align business processes and ICT.

The simulation models were also used to communicate to stakeholders that a component-based architecture for multi-channel service provisioning could allow for a more flexible, multi-channel customer driven service provisioning at a functional perspective. This helped to inform and support the cross-municipality information management council in the Netherlands to formulate feasibility requirements to initiate pilot projects. A main

value of the study lies in the creation of realistic expectations and the focus on the real design issues.

Involvement of and communications of concepts to stakeholders was essential for success. This, however, had the disadvantage that no existing methods for identifying components were usable for our purposes. Instead this research depended heavily on the knowledge and judgment about component granularity, functionality and availability in the market of experts and the use of task analysis. This research can provide input for future research focused on formal and systematic methods for designing and developing component-based architectures.

8. References

[1] S. Andriole, "The Politics of Requirements Management", *IEEE software*, vol. 15, no. 6, pp. 82-84, 1998.

[2] F.J. Armour, S.H. Kaisler, and S.Y. Liu, "A big-picture look at Enterprise Architectures", *IEEE IT Professional*, vol. 1, no.1, pp. 35-42, 1999.

[3] J.Y. Bakos, "A Strategic Analysis of Electronic Marketplaces", *MIS Quarterly*, vol. 15, no. 3, pp. 294-310, 1991.

[4] Beerens, J., *Analysepilot, Verslag werkbezoeken*, Rapport Delft University of Technology and D3k, Delft, The Netherlands, 2002 (in Dutch).

[5] J. Bisbal et al., "Legacy Information Systems: Issues and Directions", *IEEE Software,* vol. 16, no.5, 1999, pp. 103-111.

[6] Bosman, A. *Modellen van en in een Organisatie*, Lansa Publishing, Leiderdorp, The Netherlands, 1996 (in Dutch).

[7] Boxtel R. van, *Actieprogamma Elektronische Overheid*, Ministerie van Binnenlandse Zaken en Koninkrijksrelaties, The Hague, The Netherlands, 1999 (in Dutch).

[8] A.W. Brown and K.C. Wallnau, "The Current State of Component-Based Software Engineering", *IEEE Software*, vol. 15, no.5 ,1998, pp. .37-46.

[9] M. Fan, J. Stallaert, and A.B. Whinston, "The adoption and design methodologies of component-based enterprise systems", *European Journal of Information Systems*, vol. 9, no. 1, pp. 25-35, 2000.

[10] D.L. Goodhue, and R.L. Thompson, "Task-technology fit and individual performance", *MIS Quarterly,* vol. 19, no. 2, pp. 213-236, 1995.

[11] Jacobson, I., G. Booch, and J. Rumbaugh, *The Unified Software Development Process*, Addison-Wesley, Reading, 1999.

[12] Janssen, M.F.W.H.A., *Designing Electronic Intermediaries. An agent-based approach for designing interorganizational coordination mechanisms*, Doctoral Dissertation Delft University of Technology Delft, The Netherlands, 2001.

[13] Kettinger, W.J., Teng, J.T.C., and Guha, S., "Business Process Change: A study of methodologies, techniques, and tools", *MIS Quarterly,* vol. 21, no. 1, pp. 55-79, 1997.

[14] D. Kunda, and L. Brook, "Identifying and classifying processes (traditional and soft factors) that support COTS component selection: a case study", *European Journal of Information Systems*, vol. 9, no. 4, pp. 226-234, 2000.

[15] K.J.L. Layne, "Developing fully functional E-government: A four stage model", *Government Information Quarterly*, vol. 18, pp. 122-136, 2001.

[16] Law, A.M., and D.W. Kelton, *Simulation Modeling and Analysis*, McGraw-Hill, New York, 1991.

[17] Linthicum, D.S., *Enterprise Application Integration*, Addison-Wesley Information Technology Series, Boston 2000.

[18] Pegden, C.D., R.E. Shannon, and R.P. Sadowski, *Introduction to Simulation Using SIMAN*, McGraw-Hill, Blacklick, 1994.

[19] V. Peristeras, and K. Tarabanis, "Towards an enterprise architecture for public administration using a top-down approach", *European Journal of Information systems*, vol. 9, no. 4, pp. 252-260, 2000.

[20] R.E. Shannon, *Systems Simulation: The art and science*, Prentice Hall, Englewood Cliffs, New Jersey, 1975.

[21] Sol H.G., *Simulation in Information Systems Development*, Doctoral Dissertation, University of Groningen, Groningen, The Netherlands 1982.

[22] Swain, J.W. "Imagine New Worlds. 1999 OR/MS Today Simulation Software Survey", *OR/MS Today*, pp. 38-51, 1999.

[23] Tapscott, D. and A. Caston. *Paradigm Shift: The new promise of information technology,* New York, McGraw-Hill, 1993.

[24] S. Vinoski, "Where is middleware?", *IEEE Internet Computing*, vol.6, no.2, 2002, pp. 83-85.

[25] Vreede G.J. de, *Facilitating Organizational Change: The participative application of dynamic modeling,* Doctoral dissertation Delft University of Technology, Delft, The Netherlands 1995.

[26] G.J. de Vreede and D.T.T. van Eijck, "Modeling and Simulating Organizational Coordination", *Simulation & Gaming*, vol. 29, no. 1, pp. 60-87, 1998.

[27] W3C, *Web Services Architecture Requirements*, http://www.w3.org/ W3C Working Draft, 2002.