Designing Visual Interactive Systems in the e-Government Domain

Daniela Fogli
Dipartimento di Ingegneria dell’Informazione, Università degli Studi di Brescia
Via Branze 38 - 25123 Brescia, Italy, +39 030 3715666
fogli@ing.unibs.it

ABSTRACT
This paper proposes an approach to improve the work practice currently adopted by a local municipality to create e-government services. The approach is based on meta-design and end-user development. These paradigms allow empowering domain experts, namely civil servants with competencies in government procedures, to design and develop e-government applications to be used by citizens and administrative employees. This requires, on the one hand, to pay attention on the visual interaction aspects of the tools designed to support domain experts, and, on the other hand, to create suitable mechanisms for the automatic generation of user interfaces, application logic and databases. These goals are achieved by conceiving meta-design as a participatory design activity aimed at creating meta-models of e-government applications and suitable tools exploiting such meta-models for code generation.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Presentation]: User Interfaces – Theory and Methods.

General Terms
Design, Human Factors.

Keywords
End-user development, e-government, meta-design.

1. INTRODUCTION
According to the European eGovernment Action Plan 2011-2015, information and communication technologies (ICT) “can help the public sector develop innovative ways of delivering its services to citizens while unleashing efficiencies and driving down costs” [4]. Indeed, many public administrations all over Europe are currently improving their relations with citizens and enterprises by providing them with online applications that can fasten bureaucratic duties and avoid people to visit the administration offices.

The Municipality of Brescia, in Italy, is implementing the European plan by promoting an ever-increasing use of web communication with citizens and enterprises. As far as Government-to-Citizen (G2C) services are concerned, citizens can access a variety of online applications on the municipality website to carry out different activities, such as enrolling children in schools or sport courses, obtaining certificates, paying local taxes, booking an appointment at a specific municipality counter. We are collaborating with the Municipality of Brescia since 2007 to improve the process of creating e-government applications. Particularly, we have developed an end-user development (EUD) environment to be used by domain experts (civil servants) to create e-government applications for citizens [6][7]. To this end, a meta-design approach [5] has been adopted, in order to define the characteristics of the EUD environment, including its look and feel and the output to be produced. The positive evaluation of the EUD environment by end users led us to deepen the approach and consider further aspects concerning the whole work practice related to the creation and use of G2C services usually adopted at the Municipality of Brescia. In fact, our previous work focused mainly on the creation of online applications on behalf of domain experts, but neglected the actual use of the generated applications and the management of the data acquired.

In this paper, we analyze this work practice and extend the approach to include the management of data provided by citizens through the online applications. Therefore, we consider the need for the generation of databases associated with applications used by citizens and of further applications to be used by employees that must actually manage citizens’ requests. The aim is to improve the whole work practice and obtain a framework that could be used in a variety of situations and government agencies.

2. CASE STUDY
E-government applications are usually implemented by adopting the multi-step wizard and form design patterns [12]. Also the e-government applications made available by the Municipality of Brescia are structured as sequences of steps, where each step is implemented as a web form [6]. To develop these applications, the following work practice is adopted. A domain expert, typically a head office who possesses the know-how about a specific G2C service, requires software developers to create a new application to provide citizens with the given service on the web. Software developers analyze at first the requirements of the new service, which refer to: 1) the data to be acquired from citizens; 2) the procedures necessary to validate the acquired data; 3) the data to be presented to the employee who will concretely manage the citizens’ requests; 4) the procedures adopted by the employee to manage the data acquired from citizens.
Requirements related to points (1) and (2) are crucial to design the web application that citizens will use to apply for the given G2C service. This application will be called front-end application in the following. Requirements referring to points (3) and (4) are useful to design the web application that administrative employees will use to actually provide the service. It will be called back-end application in the following.

Furthermore, software developers must design a database to store the data that citizens provide through the front-end application. Both the back-end and front-end applications should then be able to submit queries to this database.

In summary, the current work practice adopted for the provision of a new G2C service can be represented as shown in Figure 1. Four stakeholders are involved: 1) the software developers, who are in charge of directly developing the front-end application, the back-end application and the database; 2) the domain experts, who collaborate with software developers by analyzing application requirements, and who possibly validate the applications created by software developers; 3) the citizens, who will use the front-end application to submit their requests to the government agency; 4) the employees, who will use the back-end application to access citizens’ data and deal with their requests.

Given the recurrent structure of G2C services, software developers are supported by a homemade configuration tool able to generate automatically web forms, which must be subsequently adjusted by intervening on the underlying code to satisfy the specific requirements of the applications under development. Furthermore, the database often does not need to be developed from scratch, but an existing database can be used, and possibly extended. However, this work practice is still too inefficient. To correctly understand the requirements provided by domain experts and translate them into the necessary configuration parameters, several interactions between software developers and domain experts are usually needed. After the creation of the applications, an approval by domain experts is necessary, in order to evaluate if they satisfy the requirements and are judged adequate to citizens’ needs. Furthermore, administrative employees in charge of managing citizens’ requests are usually presented with non-flexible back-end applications, whose look and feel cannot be personalized according to their needs and preferences.

3. TOWARDS A NEW WORK PRACTICE

The idea proposed in this paper to overcome the above limitations is creating a meta-design team [2], including software developers, human-computer interaction (HCI) consultants, and domain experts, who must collaborate to the design and development of all the software tools necessary for an automatic generation of front-end and back-end applications and of databases. Furthermore, the approach aims at transferring to the domain experts the power and burden of defining the characteristics of the e-government applications through the use of an end-user development (EUD) environment [6][7]. Such EUD environment must be customized to the domain and the characteristics of its users, and thus be designed by the meta-design team in a participatory way. The output of the EUD environment should be conceived as a ‘model’ of the application(s) to be generated. Finally, proper interpreters of the resulting application models must be developed, in order to generate the actual applications and necessary databases to be used by citizens and employees.

The design of the visual interface of the applications to be automatically generated must be carried out again in a participatory way, by involving HCI consultants and domain experts. To clarify this process, we can distinguish three main phases of the approach:

1) the meta-design phase, where all the tools and (meta-)models necessary for the creation of the e-government applications are designed and developed;
2) the design phase, during which the domain experts can create the new applications by defining their models;
3) the use phase, where the generated applications are used and possibly personalized by their users.

3.1 Meta-designing e-government applications

During the meta-design phase, there is, first of all, the need to describe the characteristics of e-government applications in a non-ambiguous and shared way. Thus, the first step of this phase is to define a meta-model for a given class of e-government services, by involving the different members of the meta-design team, each one bringing his/her own point of view on the e-government applications to be created. This is actually a participatory design activity that can be carried out through design meetings, where the meta-model is progressively defined. The final meta-model is an abstract description of the applications, which encodes the different perspectives of the team members [6]. The meta-model must be represented through a structured, flexible and interoperable language. In our case, we have chosen to define an XML schema to describe services for counter reservations.

In a second step, software developers, in collaboration with the other members of the meta-design team, must develop a visual interactive system able to support a domain expert in creating instances of the XML schema representing the meta-model of a class of e-government services. In other words, this system will allow a domain expert to create an XML document describing the steps of the new front-end application and the data it deals with. This visual interactive system must be conceived as an EUD environment, in that it will be used by people who are not expert in information technology to actually create software artifacts [8]. Therefore, this environment must not force domain experts to write any XML code, neither to know the underlying meta-model.
The XML schema plays a crucial role also in the third step of the meta-design phase, which includes the development of the front-end generator, the back-end generator and the database generator. All of them must properly use the XML document created by the domain expert through the EUD environment.

More precisely, the front-end generator will exploit the information in the XML document to generate the pages to be used by the citizen to obtain the service from the government agency. The output that such generator should produce must be studied and defined again in a participatory way. HCI specialists should prepare mock-ups of the resulting web forms, in order to gather feedback about their usability from domain experts, and possibly from representative citizens. Software developers must then develop a software system able to generate web applications whose conceptual model recalls the decided mock-ups, and which can vary according to the content of the XML document.

In a similar way, the automatic generation of the back-end applications requires a preliminary study with the participation of HCI specialists and administrative employees, in order to decide the look and feel of the pages constituting the back-end applications. A set of mock-ups to be discussed with the different stakeholders can be used also in this case, and, on the basis of them, software developers can develop the back-end generator. This system must be able to generate different but consistent back-end applications, each one customized to the service at hand, according to the XML document.

Finally, software developers will create the database generator, which must be able to generate the database tables according to the needs of the corresponding front-end and back-end applications. Therefore, the database generator will not be based on a predefined and unique entity-relationship model, but will map the content of XML elements into tables of a database suitable to the case at hand. Queries to a generated database must then be managed by taking the XML document into account, in order to correctly associate each field with its meaning.

3.2 Designing a new e-government service

Domain experts can create e-government applications by using the EUD environment designed during the meta-design phase. This system must be easy to learn and to use. For example, the EUD environment we have developed for the case of online appointment reservation [7] allows domain experts to follow their traditional way of reasoning and operating when this kind of service had to be supplied to citizens. Indeed, this EUD environment requires managing and interacting with sets of forms to be composed and/or adapted according to the service requirements.

Figure 2 shows the new work practice that can be adopted by a government agency, whenever a domain expert would like to provide citizens with a new G2C service. The domain expert has now the responsibility to define the characteristics of the service without transferring his/her requests to software developers. The generators take care of the actual code generation by exploiting the XML schema designed for that specific class of services.

3.3 Using the generated applications

At use time, citizens will access the front-end applications to obtain e-government services; whilst administrative employees will use the back-end applications to access the data provided by citizens and deal with their requests.

Figure 2: Providing a G2C service: the work practice based on EUD and automatic code generation.

Thanks to the structure of the overall project there will be some flexibility degrees in the generated applications. Front-end applications could include sub-steps of main steps that are personalized at run time on the basis of the citizen’s input. For example, Figure 3 shows a screenshot of the housing service application, where the citizen is inserting data in sub-step 3.1, after she has declared in step 3 (Personal Data) that her family includes two members besides herself. The generated sub-step includes two parts, each one devoted to acquire data about a family member (“Dati personali del familiare” in Italian). This sub-step is not presented to a citizen that declares a number of family members equal to zero.

The back-end application allows the employee to manage appointments with citizens. The employee can visualize in a tabular way all data related with the appointments foreseen in a given day (see Figure 4). Employees suggested this kind of visualization, because it recalls a personal organizer. Selecting a row in the table allows accessing details about an appointment. The employee can also launch queries to the database to search for specific data, such as all the appointments managed in a certain time period or the appointments requested by a citizen in the last six months. The employee can finally personalize data visualization by accessing a configuration page, where s/he can find a set of check boxes that correspond to the data available in the database for each appointment. By default the selected check boxes correspond to the data defined as mandatory by the domain expert and thus indicated as mandatory also in the XML document. Modifying the selection of such check boxes allows the employee to change the structure of the table that visualizes the appointments.

4. CONCLUSION

This paper presents a new approach to the development of e-government applications. The three phases that characterize the approach recall the idea at the basis of the MAPS project [1]. This project aims at providing a simple, PDA-based device that prompts instructions to support a person with cognitive disabilities – end user – in the accomplishment of her/his daily tasks. This prompting system can be developed by caregivers – domain experts – by means of a simple editing tool, created by software developers.
We have tested the different applications involved in our approach, by obtaining encouraging results. Particularly, the EUD environment has been positively evaluated with six domain experts [7]. A positive feedback also came from a civil servant who was requested to test the back-end application associated with the front-end application she created through the EUD environment. Finally, four citizens have tested two front-end applications; they have been able to complete the tasks assigned without notifying any significant problems. Notwithstanding these promising results, a re-engineering activity and further experiments are necessary before a real deployment of the tools.

Technical aspects must be better investigated, such as the adoption of web services and semantic web techniques (see for example [3][9][10][11]). The effectiveness of XML-driven generation of databases must be carefully evaluated as well, since it leads to create redundant databases and does not take care of normalization needs; we will deepen this issue in the future.

5. REFERENCES


