Metadesigning e-Government Services: 
A Case Study in a Local Agency

Daniela Fogli, Loredana Parasiliti Provenza
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Brescia
Via Branze 38 - 25123 Brescia, Italy
{fogli, loredana.parasiliti}@ing.unibs.it

Sergio Colosio
Comune di Brescia
Servizio Intranet-Intranet, Direzione Generale
Piazza Loggia 3 - 25121 Brescia, Italy
+s9 030 2977210
scolosio@comune.brescia.it

ABSTRACT
This paper describes the ongoing collaboration between the Department of Information Engineering of the University of Brescia and a local government agency. The goal is to create the socio-technical conditions for transferring the ability to develop web-based government-to-citizen services from software professionals to administrative personnel. To this end, proper end-user development techniques should be defined by capturing and exploiting employees’ best practices and expertise. With reference to a typical government service, the paper reports summary results from a study of administrative personnel work practices and explores employees’ motivations to become end-user software developers. It then outlines the design pattern language we have been defining for the considered domain to represent, from the employees’ point of view, the problems encountered while creating government services together with their possible solutions. This is the starting point for identifying a further language that addresses the problem of end-user development of e-government services from an interaction-design point of view. These pattern languages can be regarded as metadesign tools for end-user development whose final aim is to support, and even improve, existing best practices in service creation, thus increasing employees’ motivation and engagement in end-user development activities.

Keywords
E-government, end-user development, metadesign, design pattern.

1. INTRODUCTION
The term “e-government” is usually adopted to denote web-based services provided by agencies of local, state and federal governments. Such services can be supplied to several entities, from citizens and business enterprises to government agencies, thus being classified into Government-to-Citizen (G2C), Government-to-Business (G2B), and Government-to-Government (G2G) services [7]. The Department of Information Engineering of the University of Brescia is involved in an ongoing collaboration with an Italian government agency, the Brescia Municipality, for designing and developing G2C services. This collaboration has started in 2007 in the context of e-government web content creation and accessibility [4][5]. During this work we extended the content management system used by administrative personnel for web publishing with End-User Development (EUD) functionalities for generating accessible content.

Within this collaboration, a further and more ambitious need emerged from software developers working at the Brescia Municipality: empowering administrative employees to create e-government G2C services, such as online tax payment, enrollment in schools, front office reservation, etc. Developing such services is currently at the hands of the Computer Science department of the Brescia Municipality given that such activity requires programming skills that can be hardly found outside this department. However, only administrative employees, as experts of government procedures, possess the know-how to structure and characterize a given service. To develop e-government services, several interactions are thus needed between software developers and administrative personnel, not rarely affected by misunderstandings and ambiguities. Additionally, software developers consider the programming activity for creating e-government services very boring and repetitive, since service structures are usually characterized by recurring patterns, which lead to implement very similar programs. Actually, when a new e-government service has to be supplied, software developers usually proceed by replicating and adapting the code of another service, according to the new requirements. On the other hand, transferring this programming activity to administrative personnel is not so obvious: although these personnel possess the knowledge about the services offered by the municipality, they are not interested in contributing to online service creation if this means learning new software tools and acquiring new practices that are alien to their daily work and expertise.

To deal with these opposite needs, we are investigating a possible solution to the problem by adopting a metadesign perspective [3] that capitalizes on concepts and ideas deriving from the EUD area [6][9]. Our aim is to create the socio-technical conditions that allow employees of government agencies to become e-government service developers. The focus here is on designing a software environment that could foster employees’ participation in service development, without asking them to acquire additional competencies in information technologies.

The rest of the paper describes the ongoing work with the Brescia Municipality about the metadesign of G2C services and outlines the research line we are following.
2. E-GOVERNMENT SERVICES

Let us first analyze the online services made available to Brescia citizens through the municipality website (http://www.comune.brescia.it). The Brescia Municipality currently provides several G2C services classified as follows: 1) front office reservation; 2) tax payment; 3) document request; 4) document submission; 5) enrollment in courses or schools. All these services have been implemented according to a form-based metaphor, which reminds the approach citizens adopt in exploiting traditional government services. In the real world, citizens are usually given a sequence of paper forms to be filled in and submitted to a counter for receiving a utility. Similarly, in the virtual world, e-government services have been implemented as a set of web pages, organized as virtual forms, to be filled in and submitted when all data are provided and the necessary operations carried out.

For example, the service for the enrollment in a nursery school is implemented in the municipality website as a 9-step process: 1) citizen identification; 2) input of pupil’s data; 3) input of parents’ (or guardians’) data; 4) specification of parents’ work activity; 5) specification of pupil’s needs (e.g. specific diet, impairments); 6) choice of school services (full-time or part-time activities); 7) selection of critical issues for the access list to full-time services; 8) religion choice (whether attending Catholic religion courses or not); 9) summary of inserted data and confirmation. In each step, citizens are provided with a form, which contains a limited number of widget types for data insertion (text fields, radio buttons, combo boxes and check boxes). In each form, a section shows the steps performed, the step under execution, and the remaining steps. Figure 1 shows a portion of the web page during the enrollment in a nursery school while step 5 is under execution.

The form-based metaphor designed for e-government services has proved to be adequate to the heterogeneous population of potential users, due to its low cognitive burden and because it mimics the traditional interaction with paper-based forms. At present, the available online services are daily used by many citizens living in Brescia, even though alternative ways to access the same services are foreseen, e.g. phone calls or direct access to the competent office.1

3. END-USER DEVELOPERS IN THE E-GOVERNMENT DOMAIN

The service illustrated in Section 2 is a typical example of a complex G2C service whose creation requires the knowledge possessed by administrative personnel as domain experts. To understand how to support service creation, we have investigated, through unstructured interviews, employees’ profile and how they operate when supplying a service [4][5].

Government agency employees are expert in different administration issues, thus having different competencies, skills, and cultural background. Most of them are female, their age ranges in a wide spectrum and they do not hold a high education degree. Sometimes, they complain when they are charged by software systems with housekeeping activities. Finally, a key characteristic we have to take into account carefully is the low motivation of administrative employees in participating in web content authoring, which is perceived as an activity alien to their daily work. Particularly, they are willing to contribute to online service creation only if they are not forced to learn new software tools or to change their work practices radically.

Government agencies traditionally follow well-defined procedures to supply citizens with a given government service. More concretely, administrative employees usually have a set of drafted forms (paper or electronic-based) they adapt and combine to provide the service applicant with the forms to be filled in with the required information. Therefore, we can observe that the activity performed by agency employees requires again managing and interacting with sets of forms to be composed and/or adapted according to the service requirements.

4. METADESIGNING G2C SERVICES

The analysis of existing services and the employees’ best practices in the government domain helps us to (meta)design the most suitable EUD techniques allowing administrative employees to act as unwitting service developers [2]. The aim is to support these professionals to become end-user developers of e-government services and possibly improve existing work practices.

To this aim, the first step is to capture and represent the recurring behavior and practices of administrative employees in supplying a service. To represent such a knowledge, we have adopted an approach based on design pattern languages [1], by specifying, first, the pattern language for the application domain. Figure 2 (a) shows its graph structure. Each node represents a design pattern describing, at a certain level of abstraction, a problem in the domain of government service creation and its solution expressed in the employee-oriented language. For example, the pattern Arrange how to apply for a service, at a high level of abstraction, the problem of creating a new service; while the pattern Adapt/compose a form specifically addresses the following problem: “You must create a form for supplying a new service” and provides the solution “You can either select an already prepared form and decide to adapt it, or compose the

---

1 According to municipality studies 70% of enrollments in schools and courses in 2009 were carried out through online services.
form yourself, by creating its lists of options and blanks, or adding a procedure”. The solution is then refined by referring to lower-level design patterns Add a procedure, Create a list of choices and Create a list of blanks, which address, respectively, the problem of adding a new procedure to the forms (e.g. the procedure for identifying a citizen), the problem of creating a list of options for exclusive or multiple choices, and the problem of creating a list of text fields for data insertion.

This language is the basis for creating the Human-Computer Interaction (HCI) design pattern language, which addresses the problems encountered in the end-user development of e-government services from an interaction design point of view. Figure 2 (b) shows the high-level patterns of the language, which have been derived from the collection of design patterns proposed in [11]. This language should support metadesigners in defining the interaction experience offered by a system that allows administrative personnel to create software programs without being aware of that, by just carrying out their usual activities, such as editing content, selecting content from available choices, pasting content from other sources. To this aim, a wizard interaction style appears as adequate to employees’ profile, while the creation of service components could be supported by a limited number of selectable actions (action panel), each one allowing the user to add a pre-defined procedure, create text fields (fill-in-the-blanks), and generate lists of exclusive and multiple choices (dropdown chooser, edit-in-place). To keep under control the cognitive burden of service creation, these patterns, also emerged in [12], recall those patterns adopted in services themselves (see Section 2).

Both languages are implemented as web sites, the former to be used for communicating with domain experts, the latter to address interaction design issues with HCI experts.

5. CONCLUSION AND FUTURE WORK
We have described the ongoing work with the Brescia Municipality to empower administrative personnel to create web-based G2C services. A pattern-based approach has been adopted to represent, from the employees’ and HCI experts’ points of view, the problems encountered while creating e-government services and their possible solutions. This approach, differently from task-based analysis [8], does not describe the interaction according to a predefined model of tasks, but leads to identify and represent the recurring problems in the given domain along with their solutions. The approach can be extended with a scenario-based representation of pattern content [10]. A further pattern language will be defined to capture the software engineering perspective, which should drive the actual development of an interactive system for G2C service creation. These languages can be regarded as metadesign tools for EUD, which, on one hand, should technically facilitate service creation, and, on the other, socially promote and improve existing best practices, thus increasing end-user developers’ motivation and engagement in EUD activities.

6. REFERENCES