Chapter 11

DIALOGUE-BASED DESIGN OF WEB USABILITY QUESTIONNAIRES USING ONTOLOGIES

Elena García Barriocanal, Miguel A. Sicilia Urbán, León González, and José R. Hilera
Computer Science Department, Polytechnic School, University of Alcalá,
Ctra. Barcelona, km. 33.6; 28871, Alcalá de Henares, Madrid (Spain)
{elena.garciab, msicilia, leon.gonzalez, jose.hilera}@uah.es
Tel:+34 91 885 66 63 - Fax: +91 885 66 46

Abstract Questionnaires are nowadays widely used usability evaluation instruments, and several generic usability questionnaires are available. But these generic artefacts are not always appropriate to evaluate a given setting, and constructing a questionnaire from scratch is a complex task requiring both expertise and resources, so that discount-usability approaches to questionnaire-based evaluation can make a good option in many cases. In this work, a novel knowledge-based approach to design Web usability questionnaires is described. The questionnaire model comprises different ontologies including concepts regarding questions and questionnaires, the different measures that can be obtained and the tasks that have to be carried out by users in order to evaluate a specific kind of Web application. As a proof of concept for the model, a prototype questionnaire design application is also described. The application demonstrates how facts can be gathered through a guided dialogue with the user, and how the system can use this information to tailor the resulting questionnaire to the concrete situation.

Keywords: Computer-aided questionnaire design, Ontologies, Usability evaluation, Usability questionnaire.

1. INTRODUCTION

Usability can be defined as the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions [15]. Developing usable Web applications entails significant costs, since usability must be considered in all the phases of the development

R. Jacob et al. (eds.), Computer-Aided Design of User Interfaces IV, 133–146.

life cycle [18], including evaluations at different process stages. Evaluations can be carried out using different methods, like testing, inspection or inquiry, which in turn comprise different techniques, like user testing [8], heuristic methods [21] and questionnaires [23], respectively. In this work, we focus on the use of questionnaires as a usability evaluation technique. Questionnaires can be used not only to collect factual information about users, but to obtain their likes, dislikes, needs, and understandings of the system by asking them about some concrete interface aspects. Questionnaires are widely used instruments in usability evaluation for many reasons, e.g. they are reusable, they can be used remotely, and they are a convenient vehicle for massive administration and so on. But the correct construction and configuration of a questionnaire may increase evaluation costs in terms of time and resources, because previous experience is needed in order to develop an appropriate questionnaire with a minimum figure of validity and reliability. If the questionnaire is not well-designed, biased results will be obtained, because it would not collect data about what testers really want to measure. Nonetheless, as pointed out by Brooke [4], the use of "quick and dirty" questionnaires – i.e. with no demonstrated validity and reliability –, is justified to allow low-cost assessments of usability in the evaluation of industrial systems. Several existing predefined questionnaires with good scores in validity and reliability measures can be used for that purpose, e.g., QUIS [14] or WAMMI [16], but they are not always directly applicable. Depending on the application domain, these questionnaires may not cover all the desirable aspects that must be evaluated, as occurs in educational Web applications, where a very specific set of parameters must be taken into account to obtain useful measures [6]. This fact points out to the necessity of constructing some kind of questionnaire-tailoring tools that could be used as "discount-usability" artefacts [22]. As a matter of fact, some tools that allow the construction of generic questionnaires are available, but very few ones are concerned with the specifics of usability evaluation. An exception is Perlman's user interface questionnaire page (http://www.acm.org/perlman/question.html), a Web-based tool that reads questionnaires and options from files and form data, administers a questionnaire, and e-mails data to the administrator. However, this system has limited applicability, since it's based on a generic, predefined questionnaire, and it does not provide guidance for the evaluators in the definition of the tasks that participants would have to perform to carry out the evaluation.

In this paper, we approach a computer-aided design process of usability questionnaires using a logic-based knowledge representation, in an attempt to overcome the just described limitations. Concretely, we use ontologies to represent both the concepts and the concrete information surrounding the design of a usability questionnaire. The integration and use of ontologies provides design flexibility, enables the sharing of conceptual and factual structures, and constitutes a sound basis for reasoning [19]. The design process is intended for novice users or projects lacking resources, so that it can be considered a "discount usability" approach [22], as previously mentioned. Ontologies have already been applied in Web application development, as in [1], where learning systems are designed taking into account a multi-layer authoring task conceptualization, or IIPS [17], an intelligent system which is aimed at building and maintaining data-intensive Web sites using both interface and domain ontologies. In the usability area some ontological modelling representation techniques exist, like OSM [2] which provides a structured but informal representation of the ontology of a system, forming a basis for usability assessment. But the issue of questionnaire design have not been addressed yet in any of these efforts.

The rest of the paper is structured as follows. In the second section the core components of the model and the relationships between them are described, motivated in the context of usability evaluation. In the third section, a case study illustrating some of the benefits of this ontological approach is provided. Finally, conclusions and future research directions are provided in the fourth section.

2. A QUESTIONNAIRE MODEL FOR USABILITY EVALUATION

As the complete questionnaire ontology comprises a large amount of concepts, – ranging from usability evaluation generic knowledge to specific evaluable elements and tasks –, here we limit ourselves to describe the essential elements that are directly connected to the objective of the paper. Concretely, we will first sketch the overall structure of the model and then a more detailed account of some key concepts and relationships will be provided.

2.1 Overall Structure of the Ontology

As it has been described in the previous section, the design of a Web usability questionnaire can be made easier if a model that support the whole process is available. This model should represent all the essential concepts (also called terms or entities) that play a significant role in the evaluation, and it should also be rich and precise enough to enable certain subsequent automated 'intelligent' techniques aimed at aiding in the design of a questionnaire suitable for the application at hand. The elements that must be covered include the following: (a) questionnaire structure, including sections, (b) usability attributes considered, (c) functionalities provided by the Web application, and (d) the tasks that would be carried out by participants. In Fig. 1, a UML [24] diagram showing the main model entities is provided. The model described in this paper is just a view of a more comprehensive one which comprises other terms in the domain of questionnaires in usability evaluation. Some of these concepts are described in [12] (e.g., usability techniques and methods, participants' profile, etc.), and they enable the representation of all the surrounding knowledge needed to develop applications that facilitates an "enhanced" usability evaluation using attitude questionnaires [13].

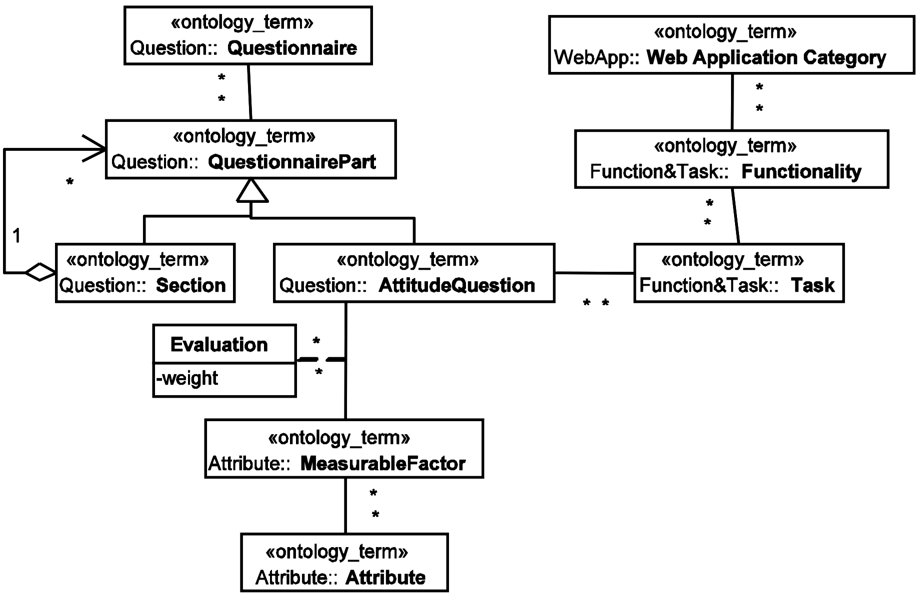


Figure 1. Core classes of the usability questionnaire model.

As we are aimed to design close-ended attitude questionnaires, we represent here exclusively the knowledge about the questions that enable the collection of user opinions according to his/her personal experience. Since it is possible that participants had never used the application before, a collection of typical tasks is provided so that they can create for themselves an opinion about the system. Each task is intended to evaluate a specific functionality of the application, and in addition, we have considered that usually each kind of Web application contains a minimum well-defined set of typical functionalities. Another important part of model describes the attributes that can be evaluated using the questionnaire. According to [20], a usability attribute can be defined as a system feature that contributes to make the system more easy-to-use. Attitude questionnaires measure user satisfaction about the application, and they can also indirectly measure the perception of the users about other usability attributes. In consequence, we have called “measurable factors” to the concrete system features that are used to estimate the intended usability attributes. These factors may have a different impact on different usability attributes, but exclusively satisfaction [15] can be directly obtained from the overall questionnaire result. A question may contribute to more than one measurable factor, and a given factor may be measured through more than one question, possibly having different weights.

2.2 Key Ontology Concepts

The elements of the model are structured in four interrelated ontologies: a “Web applications” ontology, a “functionalities and tasks” ontology, a “usability attributes” ontology and a “questionnaires” ontology, showed as UML namespaces in Fig. 1. The principles of the METHONTOLOGY approach [9] have been applied for ontology engineering, but following a literature-based process as described in [25]. In the rest of this section, a number of concepts and relations embodied in the ontology are described using description logics syntax [3]. For the sake of brevity, only elements relevant to understand the subsequent case study are provided.

2.2.1 Web Application Ontology

The Web application ontology describes the most common kinds of applications available through the Web, along with their structure. Web applications (*WA*) can be classified according to their business or information handling model. Concretely, we have adopted the taxonomy described in [5]. According to this, it can be stated the following: $WA \subseteq \exists hasType.WAType$, so that $e - Commerce \subseteq WAType$, among others. Assertions $WA(app1);\ e - Commerce(e - shop);hasType(app1,e - shop)$ can be used to denote that the Web application app1 is an e-shop. Depending on its type, a Web application usually comprises different characteristic parts ($WA \subseteq \forall includes.WAPart$), and these parts are also typed, e.g., an e-shop usually contains a registration page, a search page, a shopping cart, etc, i.e: $WAPart \subseteq \exists URL.(String) \cap \exists hasPartType.WAPartType$.

2.2.2 Functionality and Task Ontology

This ontology models both the typical functionalities (*TypFunct*) of the Web application (and/or its application parts) and the tasks that will be provided to the user as part of the evaluation. Some functionalities may be modelled as prerequisites for others transitively. Tasks may require input/output parameters, (*TInParam*) and (*TOutParam*), respectively:

$WAPartType \subseteq \exists\ usuallyHas.TypFunct$
$TypFunct \subseteq \forall\ hasPrerreq.TypFunct \cap \exists\ isTypEvaluatedBy.Task$
$Task \subseteq \forall\ requires.TInParam \cup \forall requires.TOutParam$

2.2.3 Attribute Ontology

This ontology describes usability attributes and the different factors that can be measured using a questionnaire. There is no agreed upon definition of usability [27]. Our model allows some degree of flexibility through the use of analogy and influence relations among attributes in the same or different "attribute list". Two attributes of different lists are analogous if they define the same concept using different terminology. For example, learnability as defined in Nielsen's list [20] is essentially the same that "time to learn" as defined in Shneiderman's one [26]. In addition, some attributes may influence positively others. For example, Dix defines a categorization of usability attributes at different abstraction levels [7], where flexibility is positively influenced by customisability, among others:

$Att \equiv \exists\ definedIn.AttList \cap (\forall\ isAnalogous.Att \cup \forall\ inflPos.Att)$
$AttList \equiv \forall\ contains.Att \cap \exists\ contains.Att$
$definedIn \equiv contains^{-} (symmetric\ relation)$

Several attributes can be measured (directly o indirectly) using a questionnaire. For example, WAMMI measures five factors –measurable factors in our model–, including learnability. This factor constitutes in turn an element that must be taken into account to evaluate other usability attributes, like efficiency. Some of the model terminology needed to reflect this knowledge is the following:

$Att \subseteq \forall\ isMeasuredBy.MeasurableFactor$
$MeasurableFactor \subseteq \forall\ measuresOpinionAbout.\ Attribute$
$measureOpinionAbout \equiv isMeasuredBy^{-}$

2.2.4 Questionnaire Ontology

Here we deal with attitude questionnaires with close-ended questions which may contain different sections. The model represents this fact using a composite structural design pattern [10]. A questionnaire is made up of several questionnaire parts. Each part is a question or a section, and sections may contain other questionnaire parts:

$Questionnaire \equiv \forall\ isMadeUpOf.QnnPart \cap \neg Section$
$Question \subseteq QnnPart;\ Section \subseteq QnnPart$
$Section \subseteq \exists\ contains.QnnPart$

Finally, each question is intended to contribute to one or more measurable factors possibly with different weights:

$Weight \subseteq \exists\ weights.MeasurableFactor \cap \exists\ value.(real)$
$Question \subseteq \forall\ hasWeight.Weight$

The rest of the terms of these four ontologies are integrated as sketched Fig. 1 above.

3. AN ONTOLOGY-BASED APPROACH FOR QUESTIONNAIRE DESIGN: A CASE STUDY

The model described above can be used to implement usability evaluation computer aided tools. Here we describe a prototype tool that guides the questionnaire design process through a dialogue with the user. The information needed in the different steps of the design process does not require any depth knowledge about usability evaluation, so that this approach can be considered a useful tool for novice information architects and Web designers. The tool has been developed as a Web wizard that leads the designer through the questionnaire design. During wizard execution the specific features of the concrete application that must be evaluated are asserted as instances and relations in the ontology. The application is modelled according to the characteristic defined in the predefined Web application types described above.

The first step in the dialogue collects basic application data like name, a brief description and URI, creating an instance of *WA* concept: *WA* (*app1); URL(app1,"http://…")*. In the second steps the designer specifies the application type by navigating the *Web application ontology* (Fig. 2). Concretely, the system enables navigation from the more general categories of Web applications to more specific ones –pressing *Refine* button– until no more subclasses or instances of selected terms are found (a process similar to that described in [25]). For example, in left part of Fig. 2, subclasses of *WAType* are shown, and in the right part of the same figure, the commerce application category is expanded, in this case retrieving the following instances of the ontology: *CommerceSite(e – shop); CommerceSite(e – mall); CommerceSite(virtual – market Place); CommerceSite(e – auction)*. When the designer finishes the selection of the application type, the corresponding type is asserted, for example: *hasType(app1, e – shop)*.

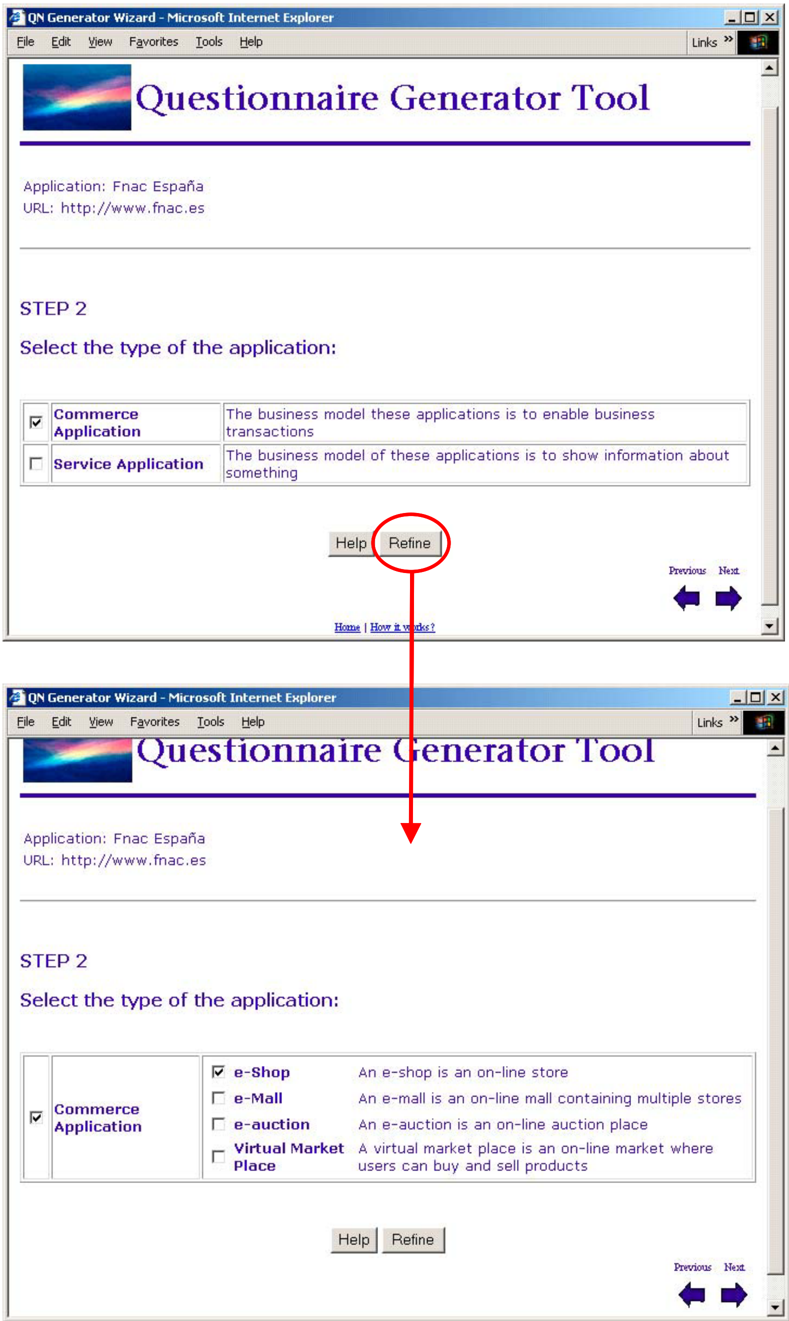
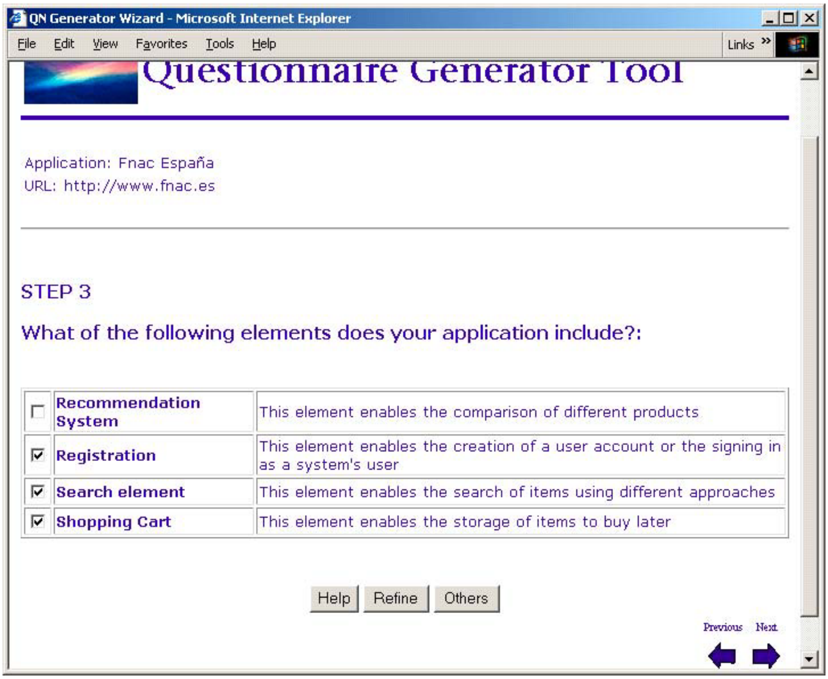


Figure 2. Second step: Selection of the application type and refinement.



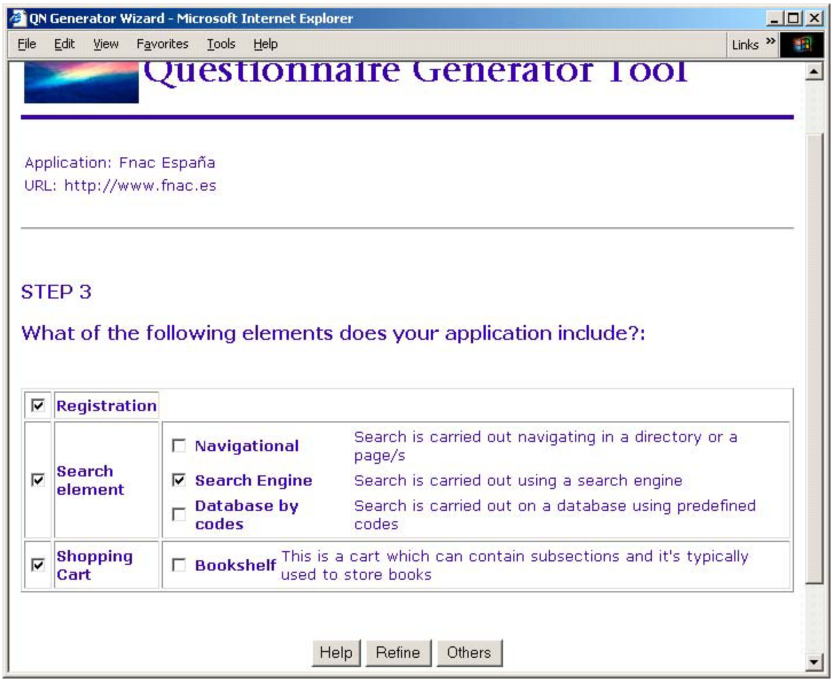


Figure 3. Third step: Selection of the elements that must be evaluated.

This navigational search through the ontology provides two main advantages: On the one hand, designers are able to use different abstraction levels to classify their application – the more specific type, the more concise become in the following steps. An on the other hand, designers are able to catalog the system using several terms at the same time, so that the approach provides a large flexibility to be used within a wide scope of applications.

Once the application type is specified, the wizard shows the parts that the selected kind of application usually includes to support its typical functionalities. Following the example, Fig. 3 shows the parts that an e-shop normally includes: recommendation system, shopping cart and searching and registration facilities.

The tool retrieves these elements using semantic relationships, e.g.:

$$WAType \subseteq \forall usuallyIncludes.\ WAPartType$$
$$WAPartType(RecommendationSystem);\ WAPartType(RegPage)$$
$$WAPartType(SearchPage);\ WAPartType(CartPage)$$
$$usuallyIncludes(e-shop,\ RecommendationSystem)$$
$$usuallyIncludes(e-shop,\ CartPage)$$
$$usuallyIncludes(e-shop,\ RegPage)$$
$$usuallyIncludes(e-shop,\ SearchPage)$$

According to the terms selected by the designer in the interface, the corresponding assertions are created. Using the *hasPartType* relation, the specific parts of the application can be linked to typical application parts (depending on the application type). For example, if the designer specifies that app1 contains a registration page and a shopping cart page, we have:

$$WAPart(app1RegPage);\ WAPart(app1CartPage)$$
$$includes(app1,\ app1RegPage);\ includes(app1,\ app1CartP\ age)$$
$$hasPartT\ ype(app1RegPage, RegPage)$$
$$hasPartType(app1CartPage, CartPage)$$
$$URL(app1CartPage\ \text{"http : //..."});\ URL(app1RegPage,\ \text{"http : //..."})$$

On the basis of the previously selected elements, tasks are retrieved using the relationships among the concepts of the *functionality and task ontology*. In the next step of the construction process, designer is asked for specific parameters required by the tasks, in order to contextualize them. To do so functionalities are obtained by traversing *usuallyHasfunctionality* from the selected instances of *WAPartType*. As the wizard shows the *typical* functionalities of the selected parts, the concrete functionalities that the application implements have to be asserted:

$$WAPart \subseteq \exists hasFunct.Funct$$

$$Funct \subseteq \exists isLike.TypFunct \cap \exists isEvaluatedBy.ConcTask$$
$$ConcTask \subseteq \forall needs.ConcTaskInParam \cap \forall needs.ConcTaskOutParam$$
$$ConcTaskInParam \subseteq \exists type.TaskInParam$$
$$ConcTaskOutP\ aram \subseteq \exists type.TaskOutParam$$

According to the selected functionalities the designer is asked for the parameters required to complete each task (e.g., element to add in the shopping cart, element and search criteria, etc.). Using this information task instances are created. The use of the *functionality and task ontology* also enables some other features like the establishment of pre-required between tasks. For example, the task use to evaluate a shopping cart part requires the sign in and/or the registration task. Subsequently the designer is asked to select the usability attributes. To do so, he is able to select a complete list or some of its attributes using a refinement process similar to the one illustrated in Fig. 2. A default list of attributes can be selected if desired. Finally, questions are retrieved in accordance with the selected attributes and functionalities, coming up with a complete questionnaire as illustrated in Fig. 4.

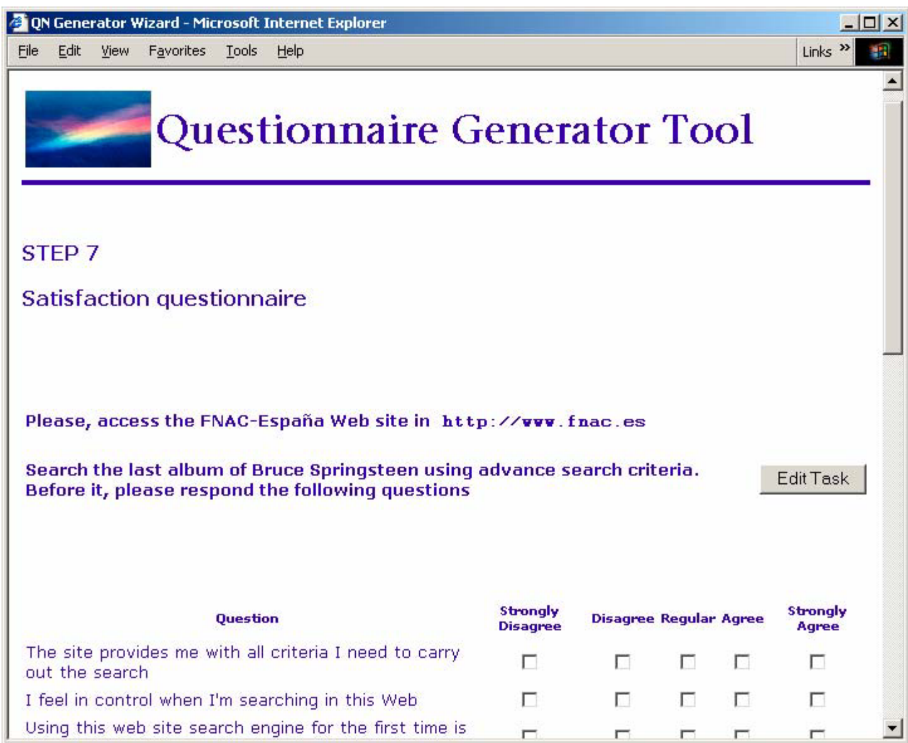


Figure 4. Example of generated questionnaire.

Once the design is completed, an editable Web form is automatically created that allows the administration of the questionnaire, and stores collected data in a relational database form consistent with the ontological model [12].

4. CONCLUSION

A new approach to design usability attitude questionnaires has been described, intended to be used as a "discount usability" tool. The approach is based on a knowledge representation comprising four ontologies: *Questionnaire ontology, attribute ontology, functionality and task ontology* and Web *application ontology*. The use of a well-defined ontological model allows for different applications like the one presented in this paper: a dialogue-based construction of questionnaires.

It can be especially useful to novice information architects and designers since the tool is able to suggest both the functionalities and the task that should be evaluated, depending on the type of the Web application. Ontology-based approaches to questionnaires also enable a common, shared information model for questionnaire results, that could be later exploited by machine learning techniques as described in [11].

REFERENCES

[1] Aroyo, L. and Dicheva, D., *Authoring Framework for Concept-Based Web Information Systems,* in Proceedings of the Workshop on Concepts and Ontologies in Web-based Educational System, during International Conference on Computers in Education ICCE'2002 (Auckland, 3-6 December 2002), CS-Report 02-15, Technical University of Eindhoven, Eindhoven, 2002, pp. 41–48.

[2] Blandford, A. and Green, T.R.G., *OSM: An Ontology-Based Approach to Usability Evaluation*, in Proc. of Workshop on Representation in Interactive Software Development, Queen Mary & Westfield College, July 1997.

[3] Borgida, A., *On the Relative Expressiveness of Description Logics and Predicate Logics*, Artificial Intelligence, Vol. 82, No. 1-2, 1996, pp. 353-367.

[4] Brooke, J., *SUS: A 'Quick and Dirty' Usability Scale*, in P.W. Jordan, I.L. McClelland, B. Thomas, and B.A. Weerdmeester (eds.), Usability Evaluation in Industry, Taylor and Francis, London, 1996.

[5] Ceri, S., Fraternali, P., and Matera, M., *Conceptual Modelling of Data-Intensive Web Applications,* IEEE Internet Computing, Vol. 6, No. 4, 2002, pp. 20-30.

[6] Díaz, P., Sicilia M.A. and Aedo, I., *Evaluation of Hypermedia Educational Systems: Criteria ad Imperfect Measures*, Proc. of International Conference on Computers in Education ICCE'2002 (Auckland, 3-6 December 2002), IEEE Computer Society, Los Alamitos, 2002, pp. 621-626.

[7] Dix, A., Abowd, G., Beale, R., and Finlay, J., *Human-Computer Interaction*, Prentice Hall, New York, 1998.

[8] Dumas, J.S. and Redish, J.C., *A Practical Guide to Usability Testing*, Intellec, Bristol, 1999.

[9] Fernández, M., Gómez-Pérez, A., Pazos, J., and Pazos, A., *Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment*, IEEE Intelligent Systems, Vol. 14, No. 1, 1999, pp. 37-46.

[10] Gamma, E., Helm, R., Johnson, R., and Vlissides, J., *Structural patterns*, in Design Patterns, Addison Wesley, Boston, 1995, pp. 163-175.

[11] García, E., Sicilia, M.A., Hilera, J.R., and Gutiérrez, J.A., *Extracting Knowledge from Usability Evaluation Databases*, in Proceedings of the 8th IFIP TC. 13 International Conf. on Human Computer Interaction INTERACT'01 (Tokyo, 9-13 July 2001), IOS Press, Amsterdam, 2001, pp. 713-715.

[12] García, E., Sicilia, M.A., Hilera, J.R., and Gutiérrez, J.A., *Computer-Aided Usability Evaluation*, A questionnaire case study, Advances in Human Computer Interaction, Typorama, 2001, pp. 85–91.

[13] García, E., Sicilia, M.A., González, L., and Hilera, J.R., *Machine Learning Techniques in Usability-Evaluation Questionnaire Systems*, in Proceedings of 2nd International Conference Learning ICDL'02 (Cambridge, 12-15 June 2002), IEEE Computer Society Press, Los Alamitos, 2002.

[14] Harper, B., Slaughter, L., and Norman, K., *Questionnaire Administration via the WWW: A Validation and Reliability Study for a User Satisfaction Questionnaire*, in Proc. of the WebNet World Conference (Toronto, 31-5 October 1997), accessible at http://www.lap.umd.edu/webnet/paper.html

[15] ISO/IEC/JTC 1/SC 7. ISO/IEC FDIS 9126-1, Software Engineering - Product quality - Part 1: Quality model, Technical Committee of International Organization for Standardization, 2000.

[16] Kirakowski, J., Claridge, N., and Whitehead, R., *Human Centered Measures of Success in Web Site Design*, in Proc. of the 4th Conference on Human Factors and the Web HFWeb'1998 (New Jersey, 5 June 1998), 1998.

[17] Lei, Y., Motta, E., and Domingue, J., *An Ontology-Driven Approach to Web Site Generation and Maintenance*, in Proc. of 13th International Conference on Knowledge Engineering and Knowledge Management EKAW'02 (Sigüenza, 1-4 October 2002), 2002, pp. 219–234.

[18] Mayhew, D., *The Usability Engineering Lifecycle*, Morgan Kaufmann, San Francisco, 1999.

[19] Menzies, T., *Cost Benefits of Ontologies*, ACM SIGART Intelligence, 1999, accessible at http://menzies.us/pdf/99sigart.pdf

[20] Nielsen, J., *Usability Engineering*, Morgan Kaufmann, San Francisco, 1994.

[21] Nielsen, J. and Mack, R., *Usability Inspection Methods*, John Wiley and Sons, New York, 1994.

[22] Nielsen, J., *Guerrila HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier*, in R.G. Bias and D.J. Mayhew (eds.), Cost-Justifying Usability, Academic Press Professional, Boston, 1994, pp 245–272.

[23] Oppenheim, A.N., *Questionnaire Design, Interviewing and Attitude Measurement*, Pinter Pub Ltd, 1992.

[24] Rumbaugh, J., Jacobson, I., and Booch, G., *The Unified Modelling Language Reference Manual*, Addison Wesley, Boston, 1998.

[25] Sicilia, M., García, E., Díaz, P., and Aedo, I., *A Literature-Based Approach to Annotation and Browsing of Web Resources*, Information Research, Vol. 8, No. 2, 2003, accessible at http://informationr.net/ir/8-2/paper149.html

[26] Shneiderman, B., *Designing the User Interface*, Addison-Wesley, New York, 1998.
[27] van Welie, M., van der Veer, G., and Eliëns, A., *Breaking down usability*, in A.M. Sasse, Ch. Johnson (eds.). Proceedings of 7th IFIP TC.13 International Conference on Human-Computer Interaction INTERACT'99 (Edinburgh, 30 August-3 September 1999), IOS Press, Amsterdam, 1999, pp. 613–620.