An Interaction Model and a Framework Dedicated to Web-based Geographic Applications

The Nhan Luong, Patrick Etcheverry and Christophe Marquesuzaà
T2i – LIUPPA – Université de Pau et des Pays de l'Adour
2 Allée du Parc Montaury, 64600 Anglet, France
{thenhan.luong, patrick.etcheverry, christophe.marquesuzaa}@iutbayonne.univ-pau.fr

ABSTRACT

Web-based geographic applications are continuously evolving and are becoming increasingly widespread. However, designing these applications is still a cumbersome task because it requires multiple and high-level technical skills related not only to recent Web technologies but also to technologies dedicated to geographic information systems (GIS). Due to this complexity, recent works highlight the difficulties to integrate sophisticated interactions in such applications. Our objective seeks to take the designers out of this complexity to allow them to focus their design activity on the interactive dimension of the application to elaborate. We propose adapted design models and a framework allowing designers to create applications according to an interaction point of view. All these contributions have been integrated in a Web 2.0 prototype allowing designers to visually specify/implement Web-based geographic applications.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces - *Web-based interaction*; D.2.2 [**Software Engineering**]: Design Tools and Techniques

General Terms

Design, Languages.

Keywords

Interaction model, code generation, geographic application design, visual authoring tools.

1. INTRODUCTION

Cartography on the Internet has caused a revolution not only in the uses of maps but also in the way to design applications presenting geolocalized data. First research work on geographic information system (GIS) concerned geolocalized data gathering and visualization. A lot of geolocalized data is now available in free geographic databases[1] and new geolocalized data can be easily gathered with a simple smartphone integrating a GPS chip. Moreover, simple tools like Google Maps allow any geolocalized data to be displayed in various formats and on several kind of maps.

Nowadays, design difficulties do not deal with gathering or displaying geolocalized data. Real difficulties concern

---

[1] http://www.geonames.org/ for example

the way to design rich interactions allowing end-users to interact with this kind of data. Recent research papers [2, 27] highlight this problem and the need to improve design methodologies and tools in order to integrate more interactivity in such a type of application. However, implementing applications requires multiple and high-level technical skills on Web technologies and GIS. Sometimes, constraints with such technical complexity may impact the design process in a negative way, leading designers to elaborate applications according to their technical skills rather than their creative skills.

In this paper, we aim at overcoming these difficulties through a design process focusing on the interactive dimension of the application to elaborate. To this end, we propose an interaction model and an associated framework facilitating the design of application according to an interactive point of view. Section 2 justifies the idea to carry out a design activity focusing on contents displayed by the application and on the way to interact with them. Section 3 compares our work according to related work. Section 4 presents the three subjacent design models focusing on the interaction model. Section 5 presents a visual design environment supporting the whole design approach, since the specification steps until code generation. Section 6 concludes this paper with a synthesis and presents some future work perspectives.

2. INTERACTION AS CENTRE OF DESIGN

Application design is often based on two essential complementary aspects: data and process modelling [13]. This approach deals with identifying data involved in the application and the processes applied on these data. However, during the first steps of the application life cycle, the designer is not able to easily define the properties of these two complementary points of view.

With a user-centred design approach [4, 12], the first focus is on what users will see and on what they will be able to do with it. Resulting descriptions focus on what the application will display, how things will be displayed and how these things will react when users will interact with them. Typically, first descriptions look like this example: "The application will display a text describing the travel and a map centred on the visited area. On a click on any town, any river or any mountain mentioned in the text, a zoom on the corresponding area will be carried out on the map and the text part describing the selected place will be highlighted." In the context of geographic applications, the user interacts with geographic data that can be presented in several ways. Generally the result of such kind of interaction deals with

the calculation, the transformation and/or the presentation of new geographic data.

From this hypothesis, we propose to focus the design approach on two points: what the user sees (what are the geographic displayed contents and how are they displayed?) and what the user can do (how must the application react when the user interacts with the displayed contents?).
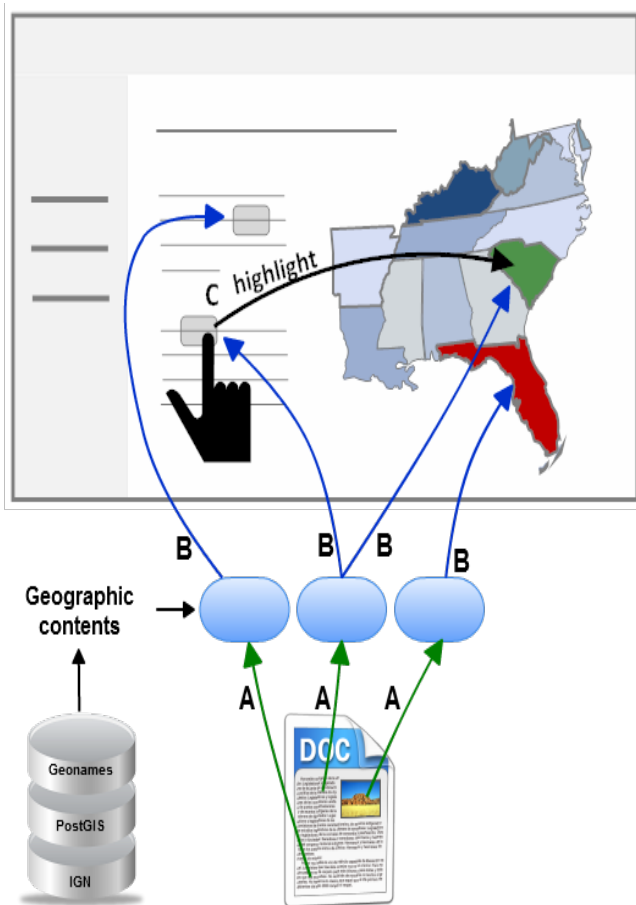

Figure 1: Design approach based on displayed contents and interaction possibilities

We propose to guide the design process considering the contents presented to the user and the interactive possibilities proposed on these contents. With such an approach (Figure 1), designer's work consists of defining which contents must be emphasized (A), the way these contents must be displayed (B) and the interactions that will be available to interact with these contents(C).

3. RELATED WORK

The work mainly deals with human-computer interaction (HCI) and interaction design.

Since there is no agreed upon interaction definition [1, 24], we can find as many interaction models as definitions. Papers describing a survey around interaction design [21, 25] propose several models to describe the various aspects of interaction:

- Tasks models, like GOMS [5], MAD [10] or CTT [22]

allow the designer to describe the activities that the user will carry out using the system. These tasks are often described in a hierarchical way and allow a specific goal to be achieved.

- Dialogue models describe the conversation properties between the user and the system. These models allow the designer to specify when the user can run commands, select or specify inputs and when the system can alert the user and display information. There are mainly three kinds of formalisms to describe the dialogue layer of an application. These formalisms are based on grammars, events or state charts.

- Presentation models aim at describing the properties and the elements composing the user interface [11]. We can distinguish abstract presentation models that allow the designer to carry out a conceptual description of the user interface (interface is described in terms of abstract objects) and concrete presentation models that allow the interface to be described in details in terms of widgets.

- Architectural models aim at identifying the significant elements and mechanisms involved in the majority of interactive systems. These models structure an application in several layers playing specific roles. Some of these layers may partially or fully correspond to the models previously described. MVC (Model-View-Controller) [14] are frequently quoted architectures that separate, each one with their specificity, the data layer, the presentation layer and the control layer which aims at managing communication flows between the two other layers.

We propose design tools allowing designers to focus on the application interactive dimension. To achieve this goal, we base on operational models to be elaborated and integrated in design environments. These environments should assist the designer from the specification step to the deployment step.

This paper is based on previous work described in [17, 16, 15, 18]. These papers focused on the geographic content model that allows the designer to define what must be emphasized with interactions [17], on the design approach [16], on the design framework [18] and on its underlying API [15]. The work presented in this paper focuses on the interaction model allowing the designer to describe the application from an interactive point of view. This model is accurately described in section 4.3.

The three proposed geographic content, interface and interaction models (Section 4) respect the admitted principle consisting in separating data from its presentation. Our contribution mainly focuses on dialogue modelling by describing what users can do and how the system reacts. Our interface model may be considered as a presentation model and our environment architecture is close to MVC. In our framework, designers may successively specify the interactive capabilities of a system. This fragmented view of interaction is favoured in order to facilitate the specification work of the designer.

Section 4 presents an overview of three models that we propose to support a design activity focusing on interaction.

4. DESIGN MODELS

This section briefly describes three models allowing to define geographic contents, the way to display them on a GUI and interactions between these contents.

4.1 The geographic content model

Geographic contents (geocontents) represent the central concept of our models because our design approach mainly focuses on emphasizing geographic data [15].

Geocontents are defined by a set of structured information, each one having a type (absolute or relative spatial entity), a value (e.g. "Paris", "Mauléon-Licharre", "10 miles in the south of Paris") and one or more possible representations of this value. Each considered representation allows a geocontent specific dimension (textual or spatial) to be emphasized (Figure 2).
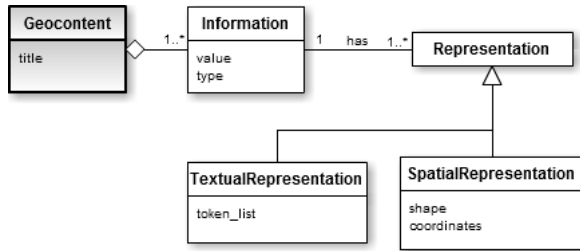
Figure 2: The content model

The textual representation of geocontents can be either: the value of the geocontent presented as a string (e.g. "Paris", "La Seine", "Mauléon-Licharre") or the position of this geocontent in the initial text used to extract it (e.g. 4th paragraph, 6th word). The spatial representation of geocontents always results in geographical coordinates allowing to locate it on a map.

The spatial representation is described according to a shape that can be either a point (for a city for example), a line (for a river or a road) or a polygon (for a specific area or a city outline for example) and a set of geographic coordinates corresponding to this shape.

4.2 The interface model

The application interface is a visualization layer allowing geocontents to be displayed in various forms [6]. An interface is built by assembling several interface components. Each component is specialized to present contents in a specific form (Figure 3). Within our approach, there are currently two kinds of interface components: `TextDisplayer` allowing geocontents to be displayed in its textual form and `MapDisplayer` allowing geocontents to be displayed on a map.

As mentioned in Section 4.1, geocontents can have several textual and cartographic representations. When geocontents are projected on an interface component, the designer will explicitly specify the representation to use.

Displaying contents on the user interface allows the designer to define the visual aspect of the final application but, as proposed by [3] "*the only way to significantly improve user interfaces is to shift the research focus from designing interfaces to designing interaction*". The next section proposes a way to model these interactions.
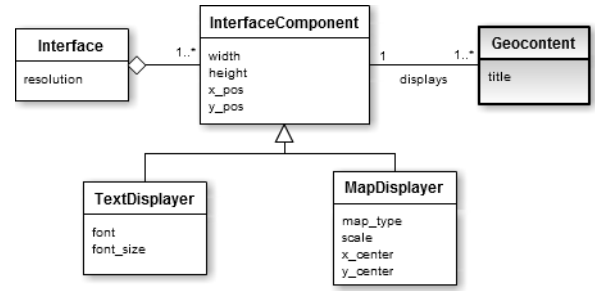
Figure 3: The interface model

4.3 The interaction model

In our approach, design is guided by interactions aiming at emphasizing contents specified by the designer. As proposed by [7, 26], the vocabulary used for designing interactions is based on user actions and system reactions. This section defines the main key concepts that we propose to describe the interactions offered by an application. Figure 4 presents an overview of these concepts.
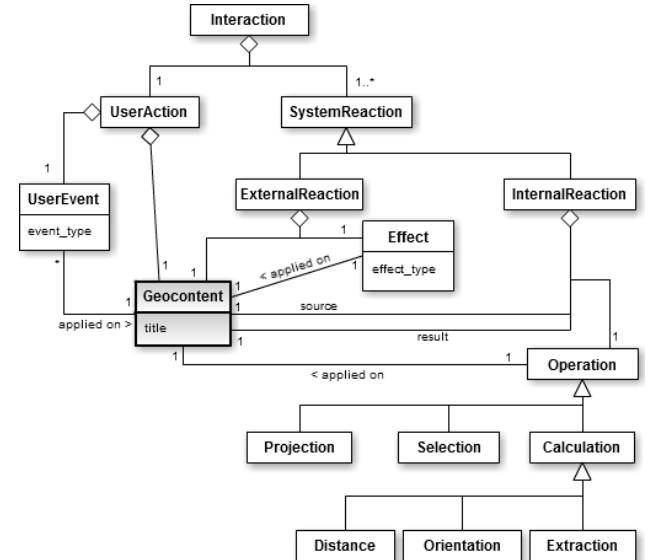
Figure 4: The interaction model

Interaction: an interaction is defined as a communication between the user and the system. This communication is always initiated by the user and ends when the application has visually reacted to the user request. An interaction is implemented by a user action triggering a system reaction.

User action: A user action always corresponds to a user event (e.g. *click*, *mouse-over*, etc.) on geocontents displayed on the screen via an interface component (see Section 4.2).

System reaction: A reaction is the system feedback resulting from a user action. As suggested by [20], we define two kinds of reactions: *internal* and *external* reactions.

External reaction: An external reaction is characterized by visual modifications of contents displayed on the interface. These modifications are carried out by the system and consist in applying visual effects on these contents.
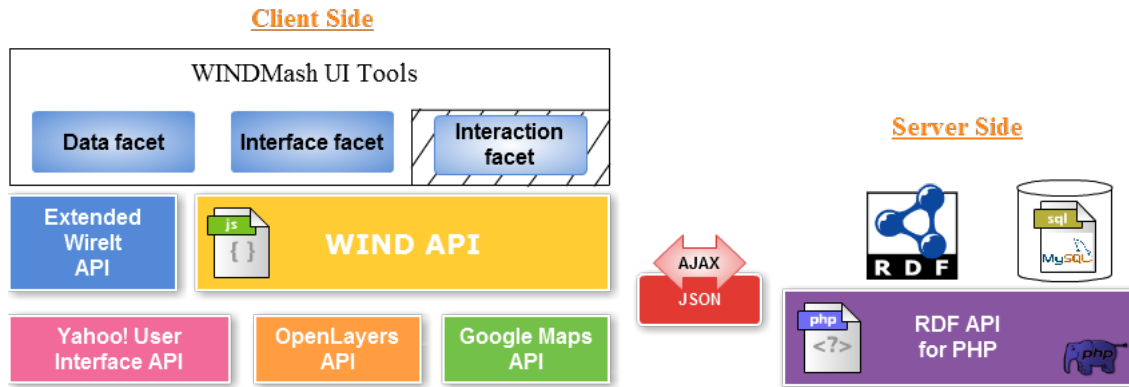
Figure 5: Web technologies used to implement the WINDMash environment.

Visual effect: An effect is a visual modification (e.g. *show*, *hide*, *highlight*, *zoom*) of contents displayed on the screen.

Internal reaction: An internal reaction is an operation allowing new contents to be computed. Three kinds of operations are possible: *projection* of contents, *selection* of contents or *calculation* of new contents.

Projection: Geographic contents may potentially be displayed on several interface components. For example, the *Paris* city can be highlighted as well in a text (occurrences of the word "Paris") as on a geographic map. The projection operation transfers existing contents towards a given interface component. This operation allows contents to be emphasized by displaying them in various forms, on different interface components.

Selection: This operation allows the user to point out specific contents by selecting a subset among contents displayed on the screen (e.g., selection of a specific town among a set of towns displayed on the map).

Calculation: This operation creates new contents by applying a calculation to contents that can be either predefined by the designer or selected by the user or previously computed (in this interaction or another interaction yet carried out). For example, a calculation operation may define all the towns located less than 100 miles from Paris.

We will now present a complete example based on these models and supported by our prototype named WINDMash[2].

5. THE WINDMASH PROTOTYPE

Our work aims at helping designers to think applications according to an interactive point of view. To relieve the designer from technical constraints, we propose an adapted software environment allowing the whole design approach to be supported, from the specification steps to the code generation of the final application.

To describe their application, designers handle WINDMash UI tools to instantiate the three proposed geographic models. To automatically generate a Web-based application according to these models, we have implemented a design environment with dedicated tools (Figure 5). Instantiated models are encoded in RDF format [19] and specific pro-

grams based on WIND JavaScript API[3]) allow the automatic generation of the application and its immediate assessment by the designer [15].

As illustrated in Figure 5, the WINDMash architecture is divided into two sides: the client one and the server one. On the client side (i.e., the web browser), we have taken full advantage of JavaScript using three open-source APIs: Yahoo! User Interface[4] API - a set of utilities and controls for building richly interactive web applications using techniques such as DOM scripting and AJAX; OpenLayers[5] API - a JavaScript library for displaying map data on Web browsers; Google Maps[6] API - a web mapping service used to get directions and locations on the map. Our WIND API is based on these three APIs and supports all concepts of the models presented in previous Section 4 for automatically generating geographic applications. We have also extended the open-source JavaScript WireIt[7] API for creating the graphical wirable modules which are used into dedicated WINDMash UI tools.

In order to use our WINDMash environment, the designers do not have to make any installation, configuration and/or maintenance. We have chosen the AJAX technology for the asynchronous communication with the server side and the JSON format for data encoding (because of its lightweight usability).

On the server side, the PHP programming language is useful to process Web services and data transformations. We have used RDF to store data on the server, because it enables to describe, to aggregate, to share and to reuse information descriptions contained into each model (i.e., contents, interface, interaction). Our code on the server side is based on RAP[8] (*RDF API for PHP*). We need also a MySQL database to save some features of our environment.

WINDMash is a prototype allowing a designer to elaborate in a visual way a Web-based geographic application according to our approach (Figure 1). The development of

---

[2]http://erozate.iutbayonne.univ-pau.fr/Nhan/
windmash3/

[3]http://erozate.iutbayonne.univ-pau.fr/Nhan/
windapi/wind.html

[4]http://developer.yahoo.com/yui/

[5]http://openlayers.org/

[6]http://code.google.com/intl/en/apis/maps/index.
html

[7]http://neyric.github.com/wireit/

[8]http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/

the *Interaction* phase is not yet fully completed but the prototype can run. We will illustrate its functioning from the following example: in order to prepare a discovery journey in the south-west of France, a teacher asks for an application displaying a text extracted from a tourist guidebook and a map showing the visited areas. To help pupils locate the various places quoted in the tourist guidebook, the teacher wishes to synchronize the text and the map as follows: When a pupil clicks on a place quoted in the text, the map automatically zooms on the corresponding place and the selected place is highlighted in the text.

In accordance with our approach, the application designer will have to define the geocontents that have to be emphasized, how these contents must be displayed and how pupils will be able to interact with these contents.The WINDMash design environment has three specific tabs supporting each one of these activities: a *Data* (geocontents) workspace, an *Interface* workspace and an *Interaction* workspace.

5.1 Specifying the geocontents to emphasize

The *Data* workspace (Figure 6), allows the designer to define the geocontents that must be emphasized in the final application. This workspace proposes, on the left side, a set of services (integrated in [17]) allowing the designer to build the geocontents to integrate in the final application.
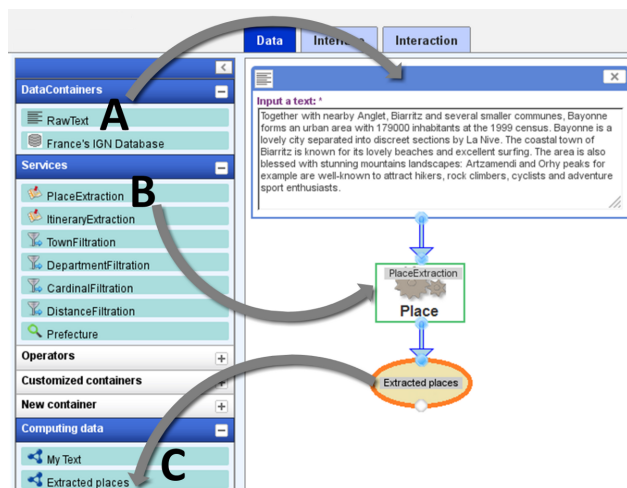

Figure 6: *Geocontents* design with WINDMash

In our example, geocontents are defined from a text extracted from a tourist guidebook: "*Together with nearby **Anglet**, **Biarritz** and several smaller communes, **Bayonne** forms an urban area with 179000 inhabitants at the 1999 census. **Bayonne** is a lovely town separated into discreet sections by **La Nive**. The coastal town of **Biarritz** is known for its lovely beaches and excellent surfing. The area is also blessed with stunning mountainous landscapes: **Artzamendi** and **Orhy** peaks for example are well-known to attract rock climbers, cyclists and adventure sport enthusiasts.*"

Starting from this raw text, the designer can use a service named "`PlaceExtraction`" [23] allowing all places (towns, rivers, etc.) quoted in the text to be automatically extracted. These extracted places become new geographic contents that will be used in the following design steps.

To create these new geographic contents (Figure 6), the designer uses a `RawText` component (A) and pastes the initial tourist text. Then, by a drag and drop operation, he/she connects the `RawText` with a `PlaceExtraction` service (B) which automatically computes a new geographic contents named `Extracted places`. These new geocontents are added to the geographic content library of WINDMash (C) and can be checked (Figure 7) by the designer: **Anglet**, **Biarritz**, **Bayonne** identified as *towns*, **Artzamendi** and **Orhy** identified as *peaks* and **La Nive** identified as a *hydronym* (waterway). When a geographic content is ambiguous (Orhy in figure 7 which is identified as a peak and a cave), the designer will be able to choose the geographic element that must be emphasized when designing interactions.

Figure 7: Geocontents overview

In the same way, the designer could define additional geographic contents using other dedicated services proposed in the WINDMash toolbox.

5.2 Specifying the display of geocontents

The *Interface* workspace of WINDMash allows the designer to define how previous created contents must be displayed on the graphic interface of the final application. WINDMash proposes two ways to present geographic contents: either in a textual way (e.g., the word *Bayonne*) or in a cartographic way (e.g., *Bayonne* town on a map). Considering the way that geocontents must be displayed, the designer create the graphic interface of its application by combining `TextDisplayer` and `MapDisplayer` components.

In our application example, the designer must create a graphic interface that displays the tourist text and the map showing the various places quoted in this text. For more reader-friendliness, the designer decides also to highlight all places quoted in the text. To create this interface (Figure 8), the designer drag and drop a `TextDisplayer` (A) and a `MapDisplayer` (B) on the workspace. The designer can easily place and resize these components in order to model the final interface of the application.

In a second step, the designer defines the geocontents that must be displayed in each one of these interface components. In the geocontent library of WINDMash, he/she selects the initial tourist text and drops it into the `TextDisplayer` pre-
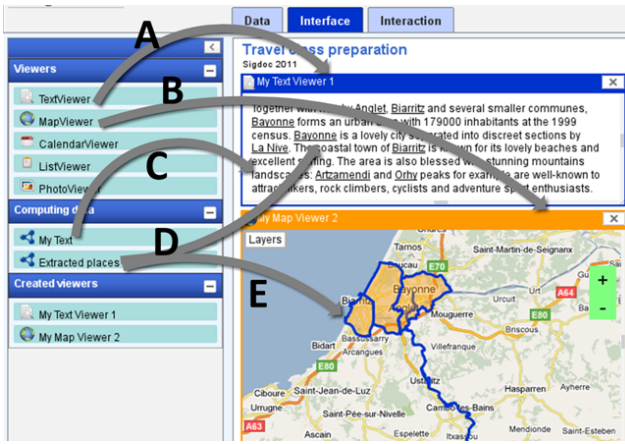
Figure 8: *Interface* design with WINDMash

viously handled (C). To highlight the places quoted in the text, he/she drags the geocontents named "`Extracted places`" and drops it into the same `TextDisplayer` as before (D). This operation highlights all the previous extracted places within the tourist text which is already displayed in this `TextDisplayer`. In order to display on the map the places quoted in the text he/she selects the geocontents named "`Extracted places`" and simply drops it on the `MapDisplayer`(E).

Thus, the interface design consists in assembling interface components and then dragging and dropping geocontents into these components. Once displayed on the screen, each geocontent can become interactive as shown in the next section.

5.3 Specifying how to interact with the displayed geocontents

In this phase, in order to design interaction, the designer should specify the behaviour of the gecontents identified in the *Data* phase and presented in the *Interface* phase. In our illustrative example, when the user selects a place in the text, this place is highlighted in the text and the map component must zoom in on this place. As illustrated in Figure 7, there are six places ("`Annotation`") automatically identified by our extraction services. These annotations are both presented on the textual displayer "*My Text Displayer 1*" and on the mapping displayer "*My Map Displayer 2*" in the *Interface* phase (Figure 8 - right part).

The JavaScript code presented in Figure 9 illustrates the implementation of the bahaviour described in the example. The variable `tpObj` is an array containing all annotations (created in the *Data* phase) on the displayer "*My Text Displayer 1*" (variable `mydoc.displayers[0]` created in the *Interface* phase). The interaction `i` is created for every element of `tpObj` with the user action "`click`". It adds successively two reactions `r1`, `r2` and is finally activated to be at end-users' disposal. `r1` is the "`highlight`" reaction applied on the same annotation on the displayer "*My Text Displayer 1*" triggered by the user action "`click`". `r2` is the "`zoom`" reaction applied on the corresponding annotation on the displayer "*My Map Displayer 2*" (variable `mydoc.displayers[1]` created in the *Interface* phase) triggered by the user action "`click`".

```
// tpObj is created in the Data phase
...

// mydoc.displayers are created in the
    Interface phase
...

// Creation of system reactions
  // Highlight on textual Annotation
  var sp1 = new WIND.LiveSensiblePart(mydoc.
    displayers[0]);
  var r1 = new WIND.Reaction(sp1, "highlight");

  // Zoom in on mapping Annotation
  var sp2 = new WIND.LiveSensiblePart(mydoc.
    displayers[1]);
  var r2 = new WIND.Reaction(sp2, "zoom");

// Creation of an interaction
  // Definition of sensible parts (instances of
      Geocontent) and UserEvent
  var i = new WIND.Interaction(tpObj, "click",
    null);

  // Association of system reactions to the
      interaction
  i.addReaction(r1);
  i.addReaction(r2);

  // Activation of the interaction
  i.activate();
```

Figure 9: A JavaScript code sample of interaction.

In the current version of WINDMash, interactions are hand coded thanks to WIND API abilities. However, in preceding work [8], we have proposed an approach dedicated to the interaction design. This approach was based on the nature of the contents to emphasize using traditional UML sequence diagrams. We thus currently plan a language derived from the UML sequence diagram which is both simplified and specialized for the description of interactions on geographic contents.

The following diagram (Figure 10) shows a possible way for the designer to specify this interaction of Figure 9.
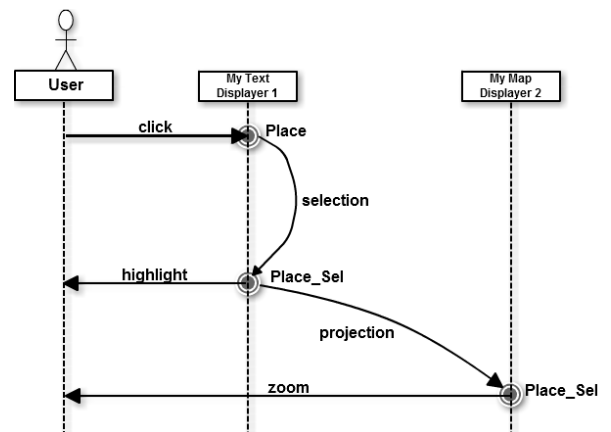

Figure 10: Example of a sequence diagram.

WINDMash not only allows a visual specification of the designed application, but also generates the corresponding source code. Each elaborated diagram is translated into an RDF file which lists the characteristics of geographic contents / interface / interactions. Merging these RDF descriptions constitutes a structured specification of the final application. WINDMash integrates a source code generator parsing these RDF files and transforming them into adequate HTML, CSS and JavaScript codes. Thus, the application may be directly assessed by the designer who can refine his/her various diagrams if the final result is not satisfactory. A demonstration video of the complete design process is available at the following URL (`http://erozate.iutbayonne.univ-pau.fr/Nhan/windmash3/demo/tourism/video.html`).

6. CONCLUSION AND FUTURE WORK

In this paper we wanted to focus the design process on the interactive dimension of the application to elaborate. We illustrated the theoretical elements of our approach within the framework of geographic application design by proposing design models and adapted software tools. The proposed design approach can be applied to other application domains because the proposed design principles remain stable: Defining the contents to highlight, defining the way to display these contents and, finally, defining how the user will be able to interact with these contents.

In a similar way as [9], we propose to specify interaction as a communication between the user and the system. However, we have chosen to couple the interface and the interaction model in order to easily describe an interaction basing on the involved interface elements.

In our proposal, an interaction is seen as a user action on geographic contents and triggering a system reaction which consists of computing and displaying new contents. From a theoretical point of view, interaction is considered as a simple process in which the user carries out an action and waits for a system reaction. We wanted to preserve this simple point of view for two essential reasons:

- to facilitate the design activity by dividing the global interaction possibilities of a system into several interactions easier to specify;

- to be able to elaborate operational models that may be automatically transformed into corresponding executable code according to model driven engineering.

The operational dimension of these models has been integrated into a design environment able to assist designers in their creation activity but also in the code production activity which is usually time-consuming. The possibility to specify an application in a visual way and then to automatically generate the corresponding code allows designers to overcome technical constraints. Hence they focus their creative skills on the interactive dimension of the application that they have to design.

Defining geographic contents that must be emphasized and then the way to make them interactive allows interactions to be described relying on these previously specified contents. We still need to propose an higher level visual language to specify the interaction abilities. This visual language would be based on UML sequence diagram which is both simple, expressive and semantically rich.

Future work also consists in allowing the designer to define several scenarios and thus several adapted sequence diagrams into the *Interaction* phase. This will enrich the usability of our environment. However, if interactions are described separately by a set of correct diagrams, the whole interaction design should also be checked to ensure the global coherence of the specified interactions.

In the WINDMash prototype, each specified interaction is encoded using RDF. We plan to exploit the underlying merging and querying mechanisms in order to try detecting all inconsistencies described by the designer. These checks should provide a first assistance level allowing the designer to keep control over the global behaviour of the application he/she is describing.

7. ACKNOWLEDGMENT

8. REFERENCES

[1] ACM SIGCHI curricula for human-computer interaction. Tech. rep., 1992.

[2] BABAR, S. *Accessibility of Web Based GIS Applications: Enhancing Accessibility of Web Based GIS Applications through User Centered Design*. LAP LAMBERT Academic Publishing, 2010.

[3] BEAUDOUIN-LAFON, M. Designing interaction, not interfaces. In *Proceedings of Conference on Advanced Visual Interfaces* (2004), pp. 15 – 22.

[4] BLACKWELL, A. F. First steps in programming: A rationale for attention investment models. In *Proceedings of the IEEE 2002 Symposia on Human Centric Computing Languages and Environments (HCC'02)* (2002), IEEE Computer Society, pp. 2–10.

[5] CARD, S. K., MORAN, T. P., AND NEWELL, A. *The Psychology of Human-Computer Interaction*, 1st ed. CRC Press, 1983.

[6] COOPER, A. *About Face: The Essentials of User Interface Design*, 1st ed. John Wiley & Sons, Inc., 1995.

[7] ENGELS, G., HAUSMANN, J. H., HECKEL, R., AND SAUER, S. Dynamic meta modeling: a graphical approach to the operational semantics of behavioral diagrams in UML. In *Proceedings of the 3rd International Conference on the Unified Modeling Language: advancing the standard* (2000), UML'00, Springer-Verlag, pp. 323–337.

[8] ETCHEVERRY, P., MARQUESUZAÀ, C., AND CORBINEAU, S. Designing suited interactions for a document management system handling localized documents. In *Proceedings of the 24th annual ACM international conference on Design of communication* (2006), SIGDOC '06, ACM, pp. 188–195.

[9] FALB, J., KAINDL, H., HORACEK, H., BOGDAN, C., POPP, R., AND ARNAUTOVIC, E. A discourse model for interaction design based on theories of human communication. In *CHI '06 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2006), CHI EA '06, ACM, pp. 754–759.

[10] GAMBOA, F., AND SCAPIN, D. Editing MAD* task descriptions for specifying user interfaces, at both semantic and presentation levels. In *Proceedings of Fourth Int. Workshop on Design, Specification, and Verification of Interactive Systems* (1997), M. D. Harrison and J. C. Torres, Eds., pp. 193–208.

[11] GUERRERO-GARCIA, J., GONZALEZ-CALLEROS, J. M., VANDERDONCKT, J., AND MUNOZ-ARTEAGA, J. A theoretical survey of user interface description languages: Preliminary results. In *Proceedings of the 2009 Latin American Web Congress* (2009), LA-WEB '09, IEEE Computer Society, pp. 36–43.

[12] GULLIKSEN, J., GOUMLRANSSON, B., BOIVIE, I., BLOMKVIST, S., PERSSON, J., AND CAJANDER, A. Key principles for user-centred systems design. *Behaviour & Information Technology Journal 22* (2003), 397–409.

[13] HOLT, J. *UML for Systems Engineering: watching the wheels*, 2nd ed. IEE Professional Applications of Computing, Septembre 2004.

[14] KRASNER, G. E., AND POPE, S. T. A cookbook for using the model-view controller user interface paradigm in smalltalk-80. *J. Object Oriented Program. 1* (August 1988), 26–49.

[15] LUONG, T. N., ETCHEVERRY, P., NODENOT, T., AND MARQUESUZAÀ, C. WIND: an interaction lightweight programming model for geographical web applications. In *Proceedings of the International Opensource Geospatial Research Symposium* (2009), OGRS'09. Available online.

[16] LUONG, T. N., ETCHEVERRY, P., NODENOT, T., MARQUESUZAÀ, C., AND LOPISTÉGUY, P. End-user visual design of web-based interactive applications making use of geographical information: The windmash approach. In *Proceedings of the Fifth European Conference on Technology Enhanced Learning* (2010), pp. 536–541.

[17] LUONG, T. N., LABORIE, S., AND NODENOT, T. A framework with tools for designing web-based geographic applications. In *Proceedings of the 11th ACM symposium on Document engineering* (2011), pp. 33–42.

[18] LUONG, T. N., NODENOT, T., LOPISTÉGUY, P., AND MARQUESUZAÀ, C. A framework to author educational interactions for geographical web applications. In *Proceedings of the Fourth European Conference on Technology Enhanced Learning* (2009), pp. 769–775.

[19] MANOLA, F., AND MILLER, E. RDF Primer. Recommendation, W3C, February 2004. `http://www.w3.org/TR/rdf-syntax/`.

[20] MARION, C. What is interaction design and what does it mean to information designers?, 1999. `http://mysite.verizon.net/resnx4g7/PCD/WhatIsInteractionDesign.html`.

[21] MORI, G., PATERNO, F., AND SANTORO, C. Design and development of multidevice user interfaces through multiple logical descriptions. *IEEE Trans. Softw. Eng. 30* (August 2004), 507–520.

[22] PATERNÒ, F., MANCINI, C., AND MENICONI, S. ConcurTaskTrees: A diagrammatic notation for specifying task models. In *Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction* (1997), vol. 96, pp. 362–369.

[23] SALLABERRY, C., ROYER, A., LOUSTAU, P., GAIO, M., AND JOLIVEAU, T. GeoStream: Spatial information indexing within textual documents supported by a dynamically parameterized web service. In *Proceedings of the International Opensource Geospatial Research Symposium* (2009), OGRS'09. Available online.

[24] SEARS, A., AND JACKO, J. A., Eds. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, 2nd ed. September 2007.

[25] SILVA, P. P. D. User interface declarative models and development environments: A survey. In *Proceedings of DSV-IS2000* (2000), LNCS, Ed., vol. 1946, Springer-Verlag, pp. 207–226.

[26] STÜHMER, R., ANICIC, D., SEN, S., MA, J., SCHMIDT, K.-U., AND STOJANOVIC, N. Lifting events in RDF from interactions with annotated web pages. In *Proceedings of the 8th International Semantic Web Conference* (2009), Springer-Verlag, pp. 893–908.

[27] WILSON, D. C., LIPFORD, H. R., CARROLL, E., KARR, P., AND NAJJAR, N. Charting new ground: modeling user behavior in interactive geovisualization. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in Geographic Information Systems* (2008), pp. 61:1–61:4.