From concept modelling to scene interpretation disposals: raising the issue of shape uncertainty in the field of the architectural heritage

Dr. I.Dudek , J.Y. Blaise, P.Bénistant
UMR CNRS/MCC 694 MAP-gamsau
EAML 184, av. de Luminy 13288 Marseille Cedex 09 France
Tel 33 4 91 82 71 70 Fax 33 4 91 82 71 71
email. idu(jyb)(pbe)@gamsau.archi.fr

ABSTRACT

In the field of the architectural heritage, 3D models are mot often used with regards to communication purposes. But under which conditions could they help in coping with the meanders of the architectural or archaeological investigation process? In parallel, the architectural documentation researchers base on is rarely organised with regards to what it is about – meaning edifices and their morphology. But is it out of reach to consider the edifice's shapes, and their 3D representation, as an interface to this documentation? Our proposition brings to the fore a methodological approach that aims at answering to those two questions. The research presented here is part of an interdisciplinary research programme[1] that focuses on the problems of documentation, representation and analysis of the architectural heritage. One of the main issues raised is how can 3D models help in visualising not only architectural shapes and forms but also what is known and what is ignored about them. We will present our ideas and experience on how 3D models can be used, when they are not thought as final results for communication goals but as visual interpretative interfaces. We will introduce three major aspects of our research, an analysis of the architectural morphology, an analysis of the documentation conservators base their investigations, and the solutions that we have implemented in a VRML environment. In this paper, We will particularly centre our presentation on the making of VRML scenes and on the capabilities of this standard in relation with an issue that goes beyond our field of experimentation but is vital in this field, this of dealing with shape uncertainty . Our experimental set is the medieval heart of the city of Krakow.

Categories and Subject Descriptors

H.5.2 [**Information interfaces and presentation**] user Interfaces (**D.2.2, H.1.2, I.3.6**) *Graphical user interfaces*, *Interaction styles*.
D.2.2 [software engineering] Design Tools and Techniques *Object-oriented design methods*.
I.3.6 [Computer graphics] Methodology and Techniques *Interaction techniques, standards*.

General Terms

Management, Documentation, Experimentation.

Keywords

Interpretative modelling, Architectural heritage, Object orientation, VRML standard, Interfaces.

1. INTRODUCTION

Our field of experimentation is the preservation of the architectural and urban heritage. This includes a concern for the edifice itself when it is still standing, but it also includes a concern for the edifice's documentation helping to try and state for instance how the edifice evolved through time or how the edifice was when nothing is left of it today. In this research area, the meaning of the word *visualisation* is often narrowed to this of *virtual reconstruction*. But an undocumented virtual reconstruction can hardly be considered as something more than as a dead-end realistic 3D representation (see [19]). Communication through realistic renderings is all cases an abusive simplification since the morphology is not the only element that should be visualised, (see for instance [18] or [20]).

Although such realistic 3D models prove relevant with respect to communication goals [7], we favour an opposite approach in which what is "beyond" the image is more important that the image itself, in line with contribution like [29] or [1]. What we try to visualise are not the ocular effect of elements *in the real world*, but a momentary state of knowledge on the edifice and its evolution. In our experiments, we give to the word *visualisation* another meaning: this of an interpretative graphical interface to the documentation. We propose to investigate the capabilities of the VRML standard [3] in supporting a graphical coding of 3D scenes that would allow the researcher to visualise such aspects as the state of certainty of a reconstructional hypothesis, or a

[1] ARKIW PICS 1150 CNRS/KBN, APN 2001 CNRS - SHS

comparative qualitative analysis of the documentation 's content on an edifice.

In this contribution we will first present the key issues that need to be addressed in our application domain. We will then introduce the methodological aspects of our proposition, and will finally stress our implementation in the VRML environment.

2. STATEMENT OF NEEDS

In order to interface pieces of information on the edifice, we propose to use its morphology as a support for data retrieval and documentation visualisation. Consequently, we need to isolate relevant architectural concepts (or shapes) and build out of them 3D models, as developed in [14] or [11]. Once this is said several constraints appear.

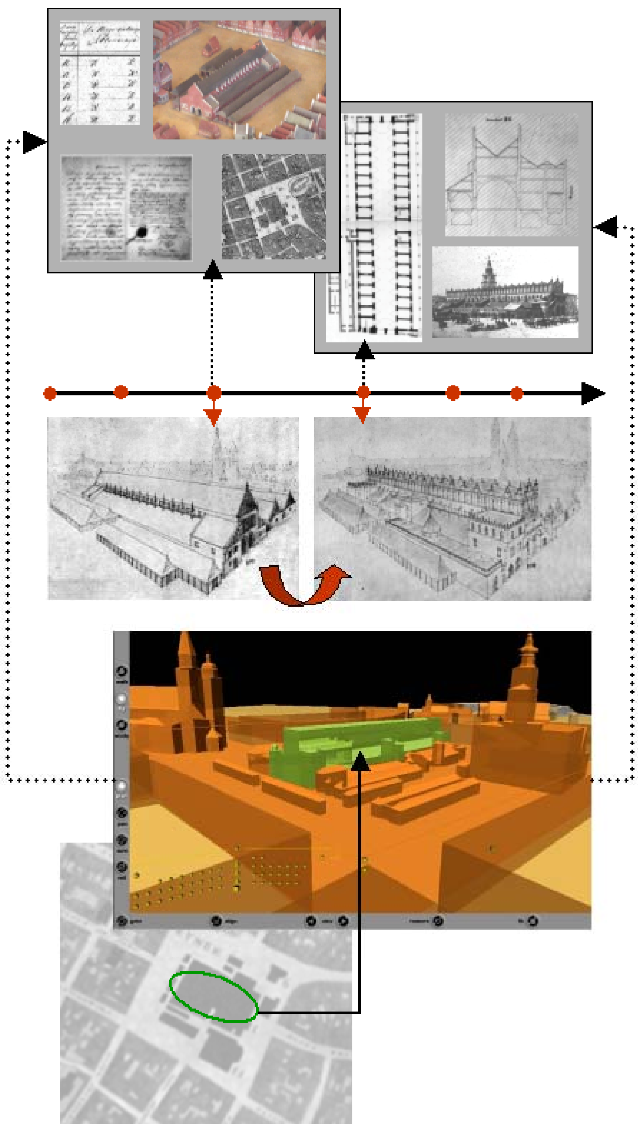


Figure 1 : Architectural and urban objects, morphological evolution in time, unique identity.

Most often, historic buildings that we study have been widely transformed throughout the centuries when they have not been totally destroyed. This means that we face the challenge to visualise the shapes that in all cases are hypothetical. We have to be aware that we will always miss some information, we will face contradictory data, etc.

3D representation of the historical objects should to take into account a large number of parameters**:**

- Morphological evolutions of the object in time (form, construction, function, ...).
- Variety and precision of the documentation the investigation is based on.
- In consequence, the notion of scale (documenting the edifice as a whole differs from documenting each element of its morphology).
- In consequence, level of certainty on the hypothesis that is represented (sure, likely, feasible, theoretically possible, ...).
- Reuses or displacements of architectural elements.
- Evolution of our knowledge and it's consequences on the model's definition .

We will briefly discus those problems in the following paragraphs in order to stress what specific 3D modelling constraints appear in our application domain.

2.1 The evolution of the object in time

During the period of life of an architectural or urban object, due to transformations throughout the centuries one element can have several forms. The transformations may be a consequence of human activity (ex. adaptations, additions, reconstructions,) or of a natural phenomenon (ex. fire, flood, earthquake). They can result in a modification not only of the object's form (ex. shape, stylistic affiliations) or structure (ex. material, constructive systems) but also in its position in space, function, property, etc.

The process of analysis of a chosen element has to take into consideration all the different phases that an element has passed through. What is therefore required is the *support for variations through time of each architectural object with preserving its identity* (variations of shape, position, etc.). Typically, an edifice retains its name although many changes can have occurred on its morphology. We will need to document and represent each phase of the edifice's evolution, and will therefore we need to formalise a theoretical model of architectural elements in which each meaningful individual concept can be given identity persistence, but state evolutions.

2.2 Variety of the documentation

Investigating an edifice's evolution bases on a documentation and it's analysis. This documentation varies in type, precision and relevance. It varies in type since it ranges from historical documents stating for instance "how many bricks" were used to build this or that part of an edifice, to recent investigations (surveys,

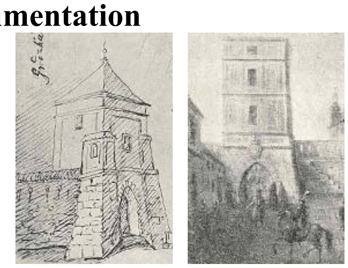


Figure 2 : Contradictory sources, two illustrations of the same city gate at the same period

archaeological findings, etc..). It varies in precision since it ranges for instance from renaissance paintings to architectural plans fixing dimensions. Finally, it varies in relevance since it ranges from actual observations on the edifice to the consultation of comparable edifices when no single piece of information on the edifice we are studying is available.

Moreover, elements of information on an edifice can be totally contradictory, calling for further investigation. This documentation obviously is far from giving all answers, far from enabling a definitive representation of the edifice's morphology. It provides clues for pursuing researches, it provides partial evidences that need interpretation.

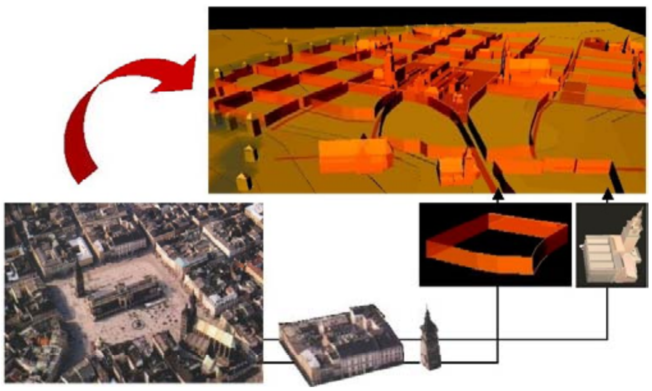


Figure 3 : From the city's morphology to its interpretation, illustration on urban blocks and edifices

Finally, it should be stressed that elements of information gathered on the edifice and its evolution are the only scientific basis for virtual reconstruction : it appears then necessary to try and figure inside the 3D scene what are a morphology's justifications.

We propose to use 3D scenes both as a mean to retrieve information from the documentation about each piece of architecture singled out in the scene, or as a mean to visualise the result of queries on the documentation. In both cases the morphology represented is a tool for investigating the edifice's documentation.

2.3 Scale issue

The documentation that is studied may deal with various subjects such as: compositional schema of the urban net, functional description of a building, architectural details, semantic analysis of the decoration, etc. In other words, the documentation that is related to one element does not relate its sub-parts or to its super-parts : each meaningful individual concept should be documented independently from the others.

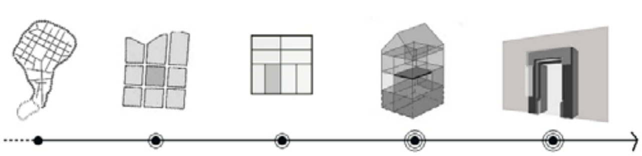


Figure 4 : The scale issue, a variety that matches the documentation

Former experiences in the use of 3D models as interfaces to a documentation [13] showed us clearly that in the case of architecture a mono-scale representation is not sufficient because it does not match the variety of the data related to the object.

We propose therefore to re-introduce of the notion of multiple architectural scale[2] in the making of 3D representation, in order to better support the documentation's variety and to deliver appropriate types of representation. This notion is oddly absent from the field of 3D modelling although its usability in the studying of the edifice has been established by [1] [28], and although its usage is widely spread in the way architects analyse edifices. Two aspects should here be disconnected:

- Multi-scale representation, matching the documentation's variety.
- Multiple levels of detail, enabling alternative representations of the same objects.

2.4 Semantics in the representations

Architectural heritage is a domain in which both documentation and visualisation play essential roles. Moreover, ensuring their interdependence has clearly been acknowledged by numerous authors as a key issue if VR models are to be included in a research process (see for instance [9], [29] or [24]): As we marked before documentation is the core of the object's description. Yet the documentation is rarely precise enough to thoroughly document all aspects of a physical object. 3D shapes to which we will want to attach pieces of information may then be *incompletely defined*, and need to be *visually marked* with an indication on what information the proposed shape is based on.

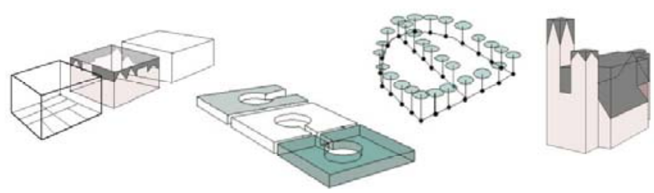


Figure 5 : Interpretative visualisation at urban structure scale

This implies the *introduction of graphical codes for 3D representation,* that would be used in order to visualise an evaluation of the nature or accuracy of the documentation attached to each architectural object represented in a scene. Such codes exist in traditional 2D representation but they are not used natively in computer-based 3D representations. We believe that, in 3D models representing reconstructional hypothesis, they could show for example the differences between the original parts and elements that were added later; or help in distinguishing what is certain in a hypothesis and what is only hypothetical.

2.5 Reuses or displacements of architectural elements

Inside an edifice that can be widely transformed, individual elements of architecture can what is more be reused or even moved somewhere else in the city. This occurs in our

[2] On issues regarding scale in architecture, also see [15]

experimental field particularly often with wooden ceilings and parts of wooden ceilings, and with openings. This introduces a level of complexity for which we lack adequate formalisms since such issues as dynamic data visualisation [27] or time handling in GIS sytems [4], although already addressed, do not bring operational breakthroughs in our application domain.

In consequence, as mentioned in section 2.4, it seems natural to try and distinguish inside the model itself elements that are reused or moved. This is not only necessary for moral comfort, it raises new questions on the edifice (ex: if this ceiling is a reused one, what was there before?) and therefore serves the investigation process.

2.6 Evolution of our knowledge

The documentation that we study is rarely precise and complete but our knowledge is changing with time. New documents or relations between different pieces of documentation are discovered, new techniques helps us to examine the objects etc. All these new data may modify the set up of a hypothesis.

More precisely, this can result in two modifications;

- Modification of an architectural element's documentation. What is then needed is an updating of the data attached to a shape but the preservation of the same interface
- Modification of the architectural element's documentation and morphology. What is then needed is a mean to re-intervene in the 3D scenes in which the element appears

Our position is that if we want our 3D models to be able to follow the evolution of our knowledge we should construct them using basing on two principles, the shape is an independent interface to the documentation, the shape is birthed out of a theoretical model defining *architectural primitives* rather than geometrical ones.

2.7 Definition of requirements.

As an answer to the preceding analysis of requirements, we have identified two research axis :

- Concept modelling, instance documentation:
  - Use architecture itself, meaning architectural shapes, as a mean to interface pieces of information,
  - enable mutual dependency of the scenes that act as interfaces and of the documentation they interface,
  - handle the evolutions in time and space of the model's instances appearing in the 3D models.
  - Monitor scene making basing on a theoretical corpus of architectural elements and allow re-interventions on the set of instances created.
- Visualisation of the documentation's analysis and elements of semantics in the representations
  - Display each object with an indication of what the documentation's analysis lets us to state about the object (original/reused, precisely dated / unprecisely dated, etc..).
  - As a consequence, introduce methods of visual marking stating the hypothetical nature of scenes and marking the documentation's analysis.
  - Show what documents justify the shapes figured.
  - Introduce multi-scale representations and alternative levels of details.
  - Add interactive graphical disposals enabling alternative readings of the scenes by letting the user to display interactively the various properties attached to the documentation's analysis, or to query various databases.

[21] have introduced geometrical objects used as visual interfaces for data retrieval on urban facilities, [30] have introduced a cad-tool-dependant representation used as an interface inside spatially determined data; where information is attached to topological concepts.

We have been trying to introduce an architectural scale in which information is attached to *architectural* concepts.

3. FROM THE DOCUMENTATION'S ANALYSIS TO THE MODEL'S REPRESENTATION.

3.1 Methodology

3.1.1 Concept modelling, instance documentation.

We consider that the best way to visualise, access and analyse the data related to the architectural and urban heritage is to use architecture itself as a mean to interface pieces of information, in line with [11]. We will therefore use the architectural and urban concepts as a means to visualise information.

In this stage, that can be defined as this of *concept modelling,* we need to identify concepts that will be used as filters on the architectural documentation. These concepts are contained in the documentation, they need to be isolated as generic pieces of knowledge that encompass various experiments, various edifices. This process takes three steps: morphological and structural analysis, identification of concepts and classification. The concepts are identified through an analysis of the morphological, structural and functional differences and similarities between the objects. Once this is done, we classify the concepts using the principle of heritage of properties (see [10] and [12]).

In our approach architecture at its various scales (from the urban analysis to the atomic elements of corpus) is described as a collection of elementary objects organised by topological relations. Each concept features a group of information (graphical and not graphical) that includes a precise definition of it's morphology as well as bibliographical references. Each concept thereby identified can then be given specific methods for visualisation, in the present experiment within the VRML environment.

The model's categorisation exploits basics of object orientation. The Aristotelian way of thinking it allows permits to gather elements that feature the same proprieties in the frame of one category. Each category can give birth to a more specialised subcategory.

A commonly used example of such organisation is the classification derived from Darwin's theory of evolution. This downwards tree-like structure includes a mechanism of heritage of proprieties and is characterised by a systematic growth of complexity. Comparable features can be observed in the field of architecture. Evolution of the architectural corpus can be expressed by successive morphological and stylistic changes enriching the collection of prototypical shapes. One could say that elements of corpus, architectural beings, birth, develop and die like natural species. This evolution is the subject of analysis of many authors dealing with history of architecture. In our work the definition and organisation of the concepts are based on the theoretical works of J.M Pérouse De Montclos[3], J. Tajchman[4] , M. Łukacz[5].

More detailed description of the methodology used can be found in the publications related to the ARKIW program[6].

Once concepts are identified, organised and formalised, the making of 3D scenes results in the instanciation of the model's theoretical shapes and a call to the relevant representation method. Such a scene features reusable elements that bear native connections to a database referencing the documentation of the instance as well as of the model.
Scenes can then be used in order to retrieve information about each object displayed, enabling it to become an *interface*.
As mentioned above, we have to handle the evolutions in time and space of the model's instances appearing in the 3D models. Instances have one unique identity but several states (of their attributes' values) are persistent, in relation with periods of time.

3.1.2 Visualisation of the documentation's analysis and Semantics in representation

The experience shows us that 3D models can be used to access and visualise the actual state of our knowledge only if our theoretical model is well adapted to the diversity of the documentation. To solve this problem we need to operate with multi-scale representation. We have defined seven different scales. For each of them the adequate concepts are identified. The concepts that have been identified and described are in relation with the chosen fields of experimentation.

In order to visualise different problems 3D representations should use different levels of abstraction, different types of coding, and a successive narrowing of the scene's span.

Each scale corresponds to different groups of problems and gives access to different types of data (bibliographic, iconographic and cartographic). Scales are classified in three groups, related to urban problems, architectural problems and problems of atomic elements. It is indispensable to mention that in this case the notion of scale is not directly related to the idea of dimension but to the elements of information described in the documentation. We follow the definition of architectural scales established by P.Boudon's works [6].

We have distinguished :

- Three *urban scales* (compositional, structural and morphologic) to cope with the questions of the urban fabrics and the issues related documentation.
- Two *architectural scales* (ensembles, architectonic) to handle the questions of dimensionally limited sets of objects and in the same time they introduce more precision, both in documentation and in representation.
- Two scales that deal with the *atomic elements* (entities and decoration) and enter into a higher level of detail, showing only particular elements of the architectural corpus and the relations between them.

From Borgès' tale [5] one can read that attempts to represent all the existing information about a territory on one single map is absurd. In the same way we believe that attempts to represent all the information related to the evolution of an urban fabric in one single 3D model appears irrelevant. Even with growing technical possibilities as mentioned in [8] or [2], we consider that from the point of view of methodology this solution can not be accepted.

Consequently we divide the *real* architectural world into different layers of abstraction. What is more we believe that in our domain one should avoid also the implementation of concepts containerisation. This means that for example the urban concepts (ex. an urban block) will not "contain" the concepts representing architectural edifices (ex. a building) although from the spatial point of view *a building* is a part of *an urban block*. In fact from the point of view of documentation content analysis those concepts may have nothing to do together. Our goal is to assure a good access to the data, therefore we have to focus on distinguishing the elements (concepts) which are used in documentation. Some of them are purely abstract concepts, some are physical beings. What joins and relates them is a position in time and space.

Documentation relates to objects at a given scale. But it also should be considered as a raw material. Only its analysis will let the researcher to know what can be derived from the documentation in terms of information on each architectural object. This analysis is in all case needed when one wants to propose a reconstructional hypothesis. The question is then : why loose in the appearance of the hypothesis' representation its most instructive aspect, the analysis? Why figure a 3D scene with graphical appearances that forbid any kind of scientific questioning? Why hide behind photo realism the essence of the investigation, and finally its very result? We believe it is vital to carry within the scene traces of the documentation's investigation, traces that will act *reading rules*. In our proposition, we strive to display each object with an indication of what the documentation's analysis lets us to state about it.

In the case of studies aiming at the creation of reconstructional hypothesis, the use of 3D computer model creates one another group of problems. As we said our knowledge is neither consistent nor precise, therefore its visualisation should take into account incoherence and impossibilities, it should make possible to mark the level of certainty, incompleteness of an hypothesis, etc.

We believe that a scene's appearance should underline the

---

[3] Jean Marie Pérouse De Montclos, "*Architecture vocabulaire - Principe d'analyse scientifique*", Imprimerie Nationale 1972-88

[4] Jan Tajchman, "*Stropy drewniane w Polsce. Propozycja systematyki*", Ośrodek Dokumentacji Zabytków, Warszawa, 1989.

[5] Marek Łukacz, "*Metrologia i pierwsza faza zabudowy staromiejskich bloków lokacyjnego Krakowa*", International Conference on Conservation, Kraków, Poland, Novembre 1998, Tom 3

[6] http://alberti.gamsau.archi.fr

problems it raises, and not hide them behind the curtain of the morphological exhaustiveness renderings tend to impose. If the knowledge about the elements that we represent is incomplete we should be able to make it visible in the scene. This question is not very widely addressed in the field of architecture, although contributions of [29][1][19] can be quoted. Our contribution introduces some proposition for uncertainty handling.

Among the wide range of properties nested inside each architectural concept we include what we called justifiers. Each object contains a group of such attributes that are responsible for displaying the object with relevant graphical codes. They are supposed to make visible inside a 3D scene the semantics associated to the source's analysis. For example the user of the scene can demand a visualisation of the level of certitude on an object's dating (represented in this example by a level of translucency).

Altogether, what we try to visualise is not the building itself but what we know and ignore about it. The experiments we have developed and report here are carried out on Kraków's medieval urban fabric. The city, former capital of Poland, has greatly changed through the ages but retains its initial layout from the mid 13th century. What is more, most edifices in the city have been transformed by internal layering, as established by [23], each building potentially retaining traces of various stylistic periods. Since a vast documentation has been gathered and preserved, and since the morphological overlapping of layers is particularly complex, the city's centre is for us a good experimentation field.

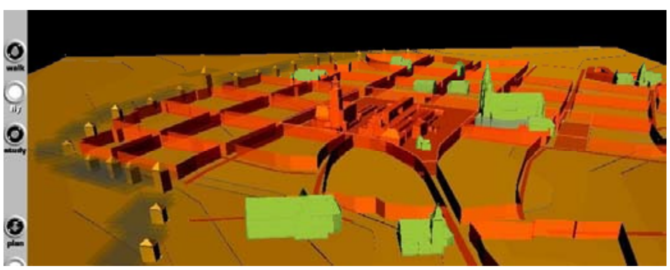


Figure 7 : Highlighting of edifices by function type

3.2 Model Implementation

Our contribution does not stress one technology but investigates a possible combination of formalisms : OO modelling, XML technologies, Interactive VRML modelling, VR scenes / e-databases interfacing. The choices we introduce hereafter can probably be better understood if we mention some of the guidelines we follow:

- Autonomy of 3D models and textual results with regards to the application that gave birth to them.
- Interactive visualisation of 3D models on the Web.
- Support for the programming of user interactions.
- 3D models stored in a format that can be manipulated with a standard programming language.
- 3D models used as graphical interfaces connecting the user either to an RDBMS or to other 3Dmodels and any other textual data.
- Use of existing RDBMS structure for the documentation itself.

3.2.1 Constitution of the model

Architectural concepts are formalised by a hierarchy of classes with the root class factorising the attributes responsible for representing the documentation's analysis.

The identification of architectural concepts and their positioning in the tree of classes of our model is a three-steps process. The edifice's morphological and structural logic is analysed in order to isolate individual concepts, with regards to these four identification rules :

- The concept corresponds to a unique object, recognised by a non ambiguous word in the architectural vocabulary (ex. the arch, the city gate),
- The concept plays a permanent and unique structural role in the ensemble it fits in (ex. arch/arcade, arcade/edifice, edifice/urbanBlock, etc..).
- The concept has an autonomous existence in the ensemble's system of topo-morphological relations (ex. the arch's function does not depend on the type and structure of the walls it fits in).

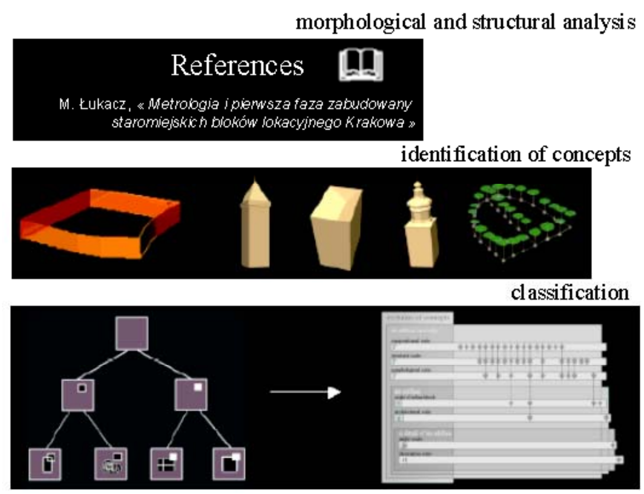


Figure 6 : Three steps in the constitution of the architectural model

- To the concept correspond relevant pieces of information in the architectural documentation.

This identification step is based on the analysis of respected scientific works[7] in which a careful attention to a non-ambiguous definition of the architectural vocabulary can be exploited for implementation in an object oriented programming language. Each concept isolated detains several blocks of attributes, five mainly qualitative – and nested inside the root class –, one related to the class' morphology - class specific. Qualitative information blocks store :

- The identification of the object, fixing notably an id for the object itself, and id's for each of the object's states.
- The localisation of the object in the city.

- A set of attributes called Evolution block fixing the dating of the object by an interval and a qualitative justification attached to the interval.
- A set of attributes called Typology block that provides a qualitative justification of the object with regards to three themes (shape, structure, function).
- Finally, A set of attributes called Documentation block that states what are the type of documents related to the object. Five such indicators are attached to conservatory documentation, four others to bibliographical, iconographical and cartographic documentation.

Evolution and Typology information blocks detain **justification attributes** : they are used to represent objects with a graphical code that indicates how credible the information we detain is with regards to specific themes (dating, shape, structure, function).

Documentation information block detains **existence attributes** : they are used to represent objects with a graphical code that indicates whether or not we have documents about the object with regards to specific media types.

These various qualitative information blocks are supported by classes that intervene in a part-of relation inside the hierarchy of architectural classes.

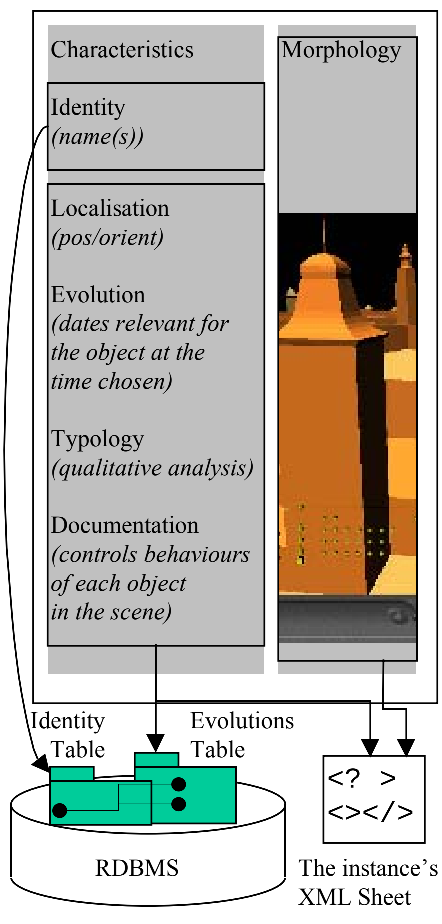


Figure 8 : Structure of classes, a combination of generic behaviours and of specific morphologies.

[7] J.M Pérouse De Montclos, J. Tajchman , M. Łukacz, op.cit.

Concepts are represented by a class that has to be positioned inside the existing tree of classes. The sixth piece of data encapsulated in each architectural class, the Morphology block, serves as the main division line in the model's organisation. More precisely, our classification is based on a morpho-structural analysis. The first level of derivation defines families of objects that share a structural role (ex: covering, opening, circulation, etc..). The corresponding classes are mainly abstract ones, they exploit the inheritance mechanism but do not fix morphological features. The second level of derivation defines individual objects or families of objects that share a morphological specificity.

In our field of experimentation, it is not credible to expect that a theoretical model will be reusable enough to exactly match each particular edifice or ensemble, its quotidian variety in the words of [22]. Our model defines a tree of classes to which we may need to add new individual concepts when the particular edifice or ensemble requires it. Each experiment will therefore potentially imply the creation of a limited number of classes that will enrich the model. Still, the model's existing structure provides the methodological tools to its extension, and the inheritance mechanism notably accelerates the process of integration of new concepts.

3.2.2 Scene making and persistence issues

Each instance of an architectural concept ought to be unique. But as mentioned before, in our application domain objects are often reused or partly destroyed. This problem has been raised in works like [23]. We have as a consequence provided each object with a persistence mechanism that stores independently the object identity (identity + concept documentation + position in the model's structure) and its various states of evolution.

Autonomy and perenniality of the VR models and of the data sheets being of crucial importance in our application domain, we have chosen to store both the visual results (VRML files) and the textual results (XML sheets) of the model's instanciation inside standard ASCII files that can be used independently form the system as a whole. In our approach, in line with [18], solutions for VRML models monitoring or Object persistence as those described in [21] or [10] are therefore not implemented here since they implicate a dependence of the results on the application that gave birth to it.

Instances are stored in an RDBMS context (mySQL) as well as in XML sheets. The top class attributes are flattened inside an identity table and inside an evolution table for its various states of evolution. Class-specific data (mainly morphology) is stored inside XML sheets.

The Parsing of XML sheets in order to re-instance and visualise objects selected by a query on the Database is done thanks to the Perl XML::SimpleObject Module [16].

The database schema includes not only the two main tables mentioned above but also the structure of the model, meaning the hierarchical relations between classes, in order to allow RDBMS-native searches not only by object type (here, object class), but also on a whole sub-hierarchy of the tree of classes. For instance, searches encompassing all coverings will be allowed by selecting the only *covering* class, from which are derived the *arch* class and all its sub categories, the *lintel* class and all its sub categories, etc..

The documentation itself is referred to by the instances' database, but is described in an independent database. This is due to the fact that we do not want to intervene on the way bibliographic entries are structured. Two tables are constructed that link bibliographic entries to the instances' database and to a localisation in the city.

Each concept detains methods relevant for persistence handling in XML files and RDBMS context, but also for scene appending in VRML files. Scenes feature instances of the model and the current state of their properties, among which the justification attributes and existence attributes mentioned in the previous section. An indicate of the documentation 's analysis (levels of certainty, type of documentation, typologies, etc…) can thereby be displayed natively or interactively inside the VRML scene, each object being represented with an appearance that indicates the Justifier's value.

3D scenes are used as a query mode (predefined time-related scenes) by selecting an object inside the 3D model or as a visualisation of the query's result, by instancing the objects corresponding to the search and calling their VRML representation method. Model and RDBMS platforms are chosen independent, the interfacing is carried out using Perl CGI Interfacing modules [10] and PHP modules that monitor the RDBMS links. The system's client/server architecture uses standard CGI programming interfaces, the various tasks are described in the following figure.

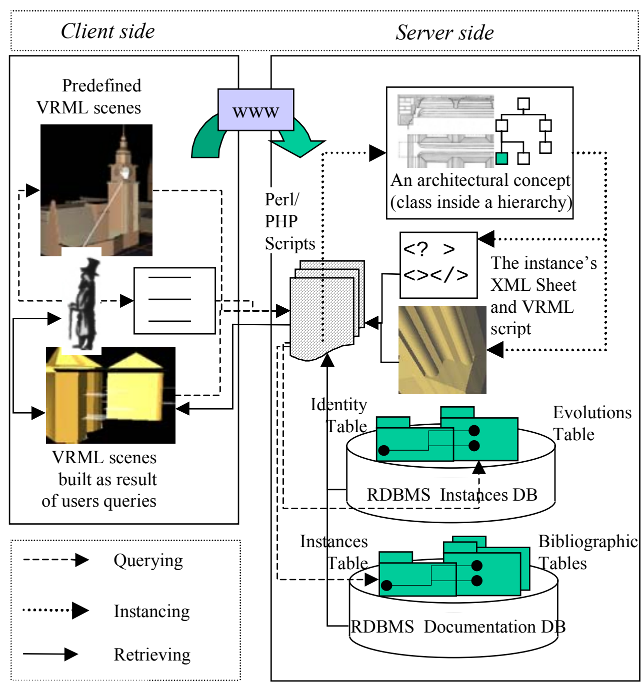


Figure 9 : System architecture

The concept's morphological characterisation provides information for the calculation of a geometry for the object. The geometrical representation in VRML can match the concept's complexity or provide a symbolical shape. But what is important to notice is that the method responsible for the representation of the object (i.e, appending a VRML file) is aware of the object's state : it can then use the qualitative information on the object to monitor alternative representations of the same object basing on what we know about it.

4. Experimentation using VRML capabilities

The pluses, minuses and possible applications of the VRML standard for architectural modelling have often been discussed , see for instance [9] or [25], we will focus its relevance in relation with our research issue. Our scenes are written in VRML 2.0 [3] both for Cosmo and Cortona plug-ins. Although often considered heavy, the language provides features that are relevant in our context, notably its events routing mechanism that we use in order to provide the user with client-side interaction disposals that are nested inside the scene and therefore not dependant on an application or an applet (see [17]). Several key aspects of this language are exploited in our development, and some of its capabilities remain leading-edge ones with regards to *interpretative modelling* issues (LOD nodes, nested/reusable interaction procedures, etc..). We have stressed the need to create scenes that would remain autonomous form the application that created them. By saying this, we rejected the possibility of investigating JAVA/VRML solutions (see [26]) that various experiences such as [21] or [8] have proven efficient; but that seem too exposed to versioning problems for use in our application domain (See for a discussion on this point [13]).

4.1 Monitoring the making of scenes

In our case, the making of scenes has to be monitored from outside the representation platform, by a standard programming language. This is due to the fact that scenes correspond to user queries and not to predefined arrangements of shapes.

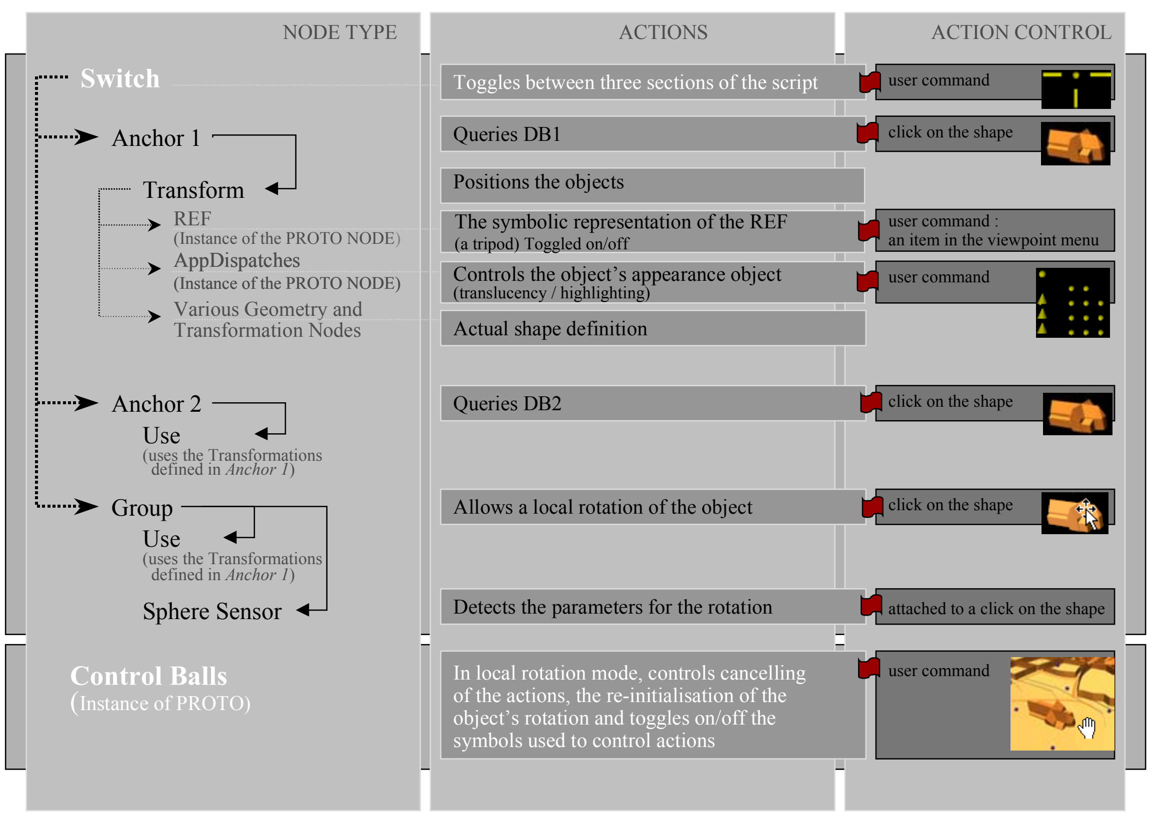
The creation of scenes is monitored by a group of classes that are in charge of several tasks, with regards to choices made by the user:

- Writing proper head and bottom for the VRML file
- Writing camera definitions
- Writing material PROTOS
- Writing interaction PROTOS, scripts and calling them
- Calling each instance's VRML file appending method
- Writing ROUTES between the set of interaction PROTOS and the set of instances displayed in the scenes

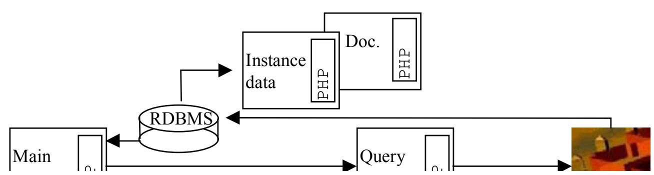
What we call interaction p
to be stressed that each ins
of the VRML file. Howe
VRML file, with notably
definition and the routing
a set of independent tool c

4.2 Scenes as a We

Our objective is to displ
environment, and use the
database. VRML brings
ANCHOR node. In a f



geometrical definition of each instance inside an ANCHOR node that contained a query on the database (in the form of a URL address). But in this first solution only one query was attached to each instance. In fact we may need to attach various queries to one instance. In the actual state of our development we need to attach three queries to one instance: a query on the states of its attributes, a query on its documentation, a query on its terminological definition. In our current experiment, we provide the scene with a user control that lets for the choice of the database to query when selecting an object. The control spans all objects in the scene. Each instance is described inside the VRML scene by a SWITCH node that distinguishes three ANCHORS. The first ANCHOR section defines (DEF node) the geometry of the instance. The two remaining only call it (USE), reducing the weight of the resulting file. It is clear that VRML is a standard that calls for a maximum attention to weight problems : it is not difficult to be verbose in VRML. But since we produce the actual VRML file from within a programming language, we are more keen to pay attention to it. The scene making process is described in the following figure.



The following table sums up the interaction possibilities we have implemented (nested in the VRML file), on the client side :

Object control

Control on :	Modification of :	Commanded by:
Justifiers (justification attributes, i.e. semantics associated to the source's analysis)	Color (transparency)	Click on the control
Documentation (existence attributes)	Color (highlighting)	Click on the control
Shape investigation	Rotation	Click on the control + on the object
Database querying	Anchor	Click on the control + on the object
Referential visualisation	On/off toggle	Click in the viewpoint menu

Scene control

Figure 13 : Color codes using transparencies

Control on :	Modification of :	Commanded by:
Controls	On/off toggle	Click in the viewpoint menu
Lighting	Scaling of light intensity	Open and position the slider
Ground anamorphosis	updating of Transform's scale	Open and position the slider

As can be observed in these tables, the *menu* nested in the viewpoints list controls toggles, whereas other controls modify elements in the scene. This means that the scene can be displayed and read without visualising interaction disposals.

The main families of interaction disposals nested in the VRML scenes are:

- Highlighting buttons : they are used in order to visualise presence or not of each type of documentation on each edifice represented. Each button corresponds to a particular item in the list of documentation types (ex: inventory, architectural drawings, photographs, etc..)
- Transparency cones : they are used to show on each edifice inside the scene how precise the documentation is : it in fact is a graphical interpretation of the justifiers. Justifiers values, initially a qualitative information, are given a numerical value that is used in order to control the object's appearance node's transparency value. Just as the information is qualitative, its visualisation is a qualitative indication.

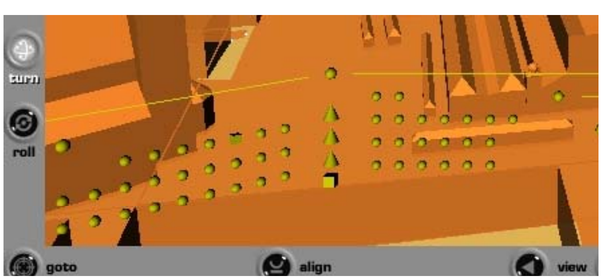
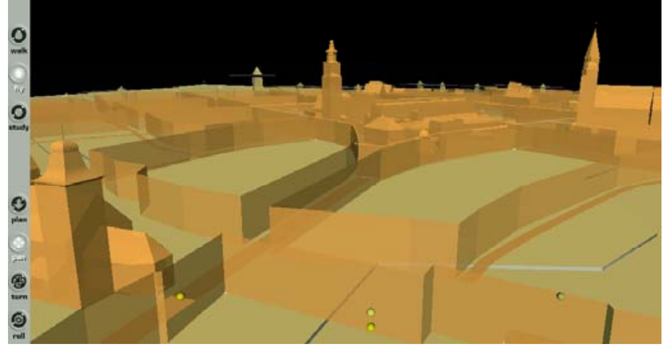


Figure 12 : Highlighting buttons and cones



- Viewpoint controlled actions : actions are nested in the viewpoint list that in this case acts as a menu, these actions toggle on and off the representation of the other controls or of each object's referential symbolised by a RGB tripod. Naturally selection of such as action is monitored not to influence the position of the observer.
- Global scene control sliders : they provide a client-side control on ground elevation and lighting conditions inside the scene. They are called sliders since they are connected to a position tracker (one of the Sensor nodes provided by the language). The tracker is nested in a PROTO node that we add to each scene and that allows the user to choose a value in a scale from (0 to 1)*factor, where factor is chosen with regards to the semantics of the slider. The PROTO offers a generic slider mechanism that can then be used for several global scene control sliders, reducing the weight of the VRML file.
- Anchor selection : we provide each scene with a control that sets which URL will be required when a click on an object is done.

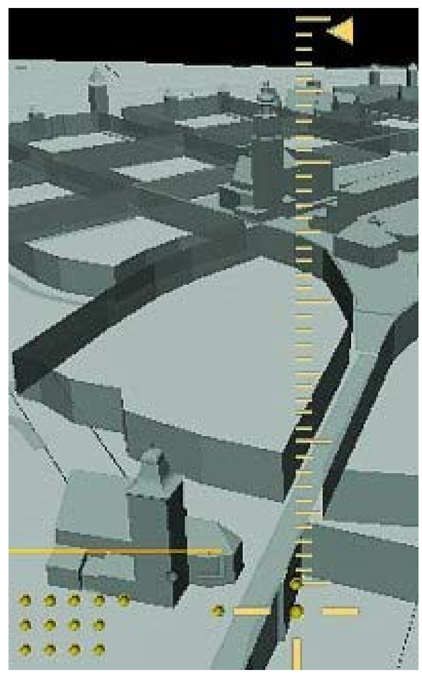


Figure 14 : The slider for lighting control, once opened.

4.4 Use of prototyping

VRML's PROTO nodes are widely used in the writing of the files. Of course an important aspect is that they help reducing the weight of the file. Although most objects represented are given a geometrical definition inside their section of the VRML file, some complex shapes are defined as PROTOS and only instanced inside their section of the VRML file. It is for example the case of the Edifice Class for which we developed a PROTO that produces an Extrusion shapes basing on two MFVec3f fields (spines) and four MFVec2f fields (two sections, two scales). The Edifice PROTO also contains fields in charge of supporting the various interactions (appearance changes, position changes, anchoring).

But the role of PROTO nodes in our development is also, and above, to control the interaction disposals. All PROTOS are written inside each VRML file produced. They are instanced at proper positions in the file under the responsibility of the VrmlScene Class. PROTOS for interaction disposals contain information on how to represent themselves with a given geometry, scripts that for instance display or hide parts of their geometry, and in general send events that will be routed to the proper sections of the VRML file.

Typical scenes contain ten PROTOS among which seven are scene controls instanced once, and three are object controls instanced inside each object. Additional PROTOS may be called if an object's geometrical definition requires it.

In terms of weight, VRML file produced are the addition of object sections, and therefore the weight could become a real difficulty. But we have mentioned the fact that we define *architectural scales* : the amount of detail of objects to display is therefore in relation to the scale addressed, the more detailed the scene is the less territory it covers. At urban scale, the scenes feature all the historical centre. But at more detailed scales only some or even one edifice will be investigated. We can reasonably expect the number of objects in the scene to remain more or less the same at each scale. At the urban structural scale, the Edifice concept is the most costly in terms of weight : in the figure below the highlighted edifice on the left "weighs" 4kb and this on the right 7 kb. In both case their morphology is relatively complex in comparisons with other edifices shown. An object of average weight like a city fortification gate weighs 2kb. This seems relatively acceptable since at that scale the edifices represented are limited to major ones. Individual buildings are not taken into consideration at this scale but at a lower one (as can be seen on the image only a vertical envelope represents urban blocks).

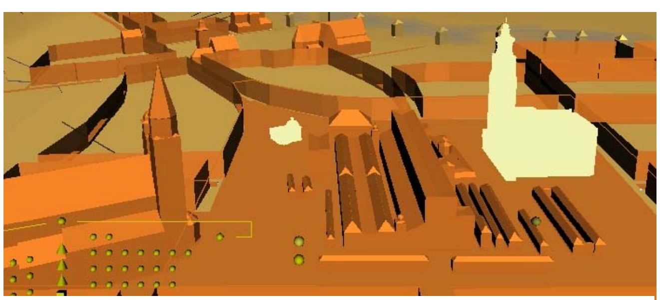


Figure 15 : Highlighting of two instances of the Edifice class, by selection of a documentation type

It is clear that PROTOS do help in trying to reduce the weight of scenes. But in our case our should bear in mind that our knowledge about objects evolves with time, forbidding de facto to count on a predefined set of parametric PROTOS to cope with the complexity of the architectural shapes. We think that the usability of PROTOS in terms of interaction monitoring is clear, but we believe its use as a shape prototyping tool should be carefully investigated with regards to the requirements of the application domain.

5. CONCLUSION

our position is that 3D models of the architectural shapes our documentation is about, are a natural and efficient filter for data visualisation and retrieval. We have proposed a methodology for featuring inside 3D models instances of a theoretical model detaining information on their existence's justification. We have implemented a technological proposal based on a combination of formalisms and stressed the possibilities that the VRML standard offers with regards to our research issue. We have not discussed the pluses and minuses of the language or of its usability, but have tried to see which of its features are suitable in the context of interpretative modelling. It appears from our experience that although the language's portability, weight and support for complexity may be obstacles, it does provide acceptable solutions, and offers an adequate framework of formalisms for the representation of architectural interfaces.

Our work clearly positions visualisation in our application domain as an *interpretation*, with an ambition not for realism but for the better documentation readability and access, in line with contributions such as [1], [19] or [29]. We however regard our contribution as nothing more than a first step in trying to use 3D modelling in the visualisation of archival information. We believe that it s possible to greatly enrich the usefulness of 3D representations provided that some attention is put to the semantics behind the rendering, and that this question opens a research area that needs more involvement.

6. REFERENCES

[1] P. Alkhoven, The changing image of the city. A study of the transformation of the townscape using Computer assisted Design and visualisation techniques (Utrecht: PHD of the University of Utrecht, 1993).

[2] F.Althoff, T.Volk, G.McGlaun, M.K.Lang, A generic user interface framework for virtual reality applications, in Proceedings 9th HCI (HCI International 2001) Conference, New Orleans, Louisiana, USA, 5-10 08 2001, Published by Lawrence Elbaum. New Jersey.

[3] A.Ames, L.Nadeau, R.David, J.L.Moreland, VRML 2.0 Sourcebook, John Wiley and sons, NY 1997.

[4] A. Bilgin, An adaptable model for a systematic approach to conservation walks : an introductory study on GIS for the urban archaeological site of Bergama, in Proceedings of. SFIIC 97 Conf., Chalon sur saone, France, 1997, 211-219

[5] J.L Borgès, Ficciones / fictions, French edition Gallimard; Paris, France, 1992, ISBN : 2070389049

[6] P. Boudon, Sur l'espace architectural, essai d'épistémologie de l'architecture, (Dunod Publications, Paris, France, 1971).

[7] R. Burton, M.E Hitchen, P.G Bryan, Virtual Stonehenge : a fall from disgrace ?, Proc. CAA 97 (Computer Applications in Archaeology), Birmingham, 1997.

[8] D.Brutzman, The Virtual Reality Modelling Language and Java, Communications of the ACM, vol 41 n° 6, June 1998, PP 57-64.

[9] D.A Campbell, VRML in architectural construction documents : a case study in Proceedings of the 3rd VRML symposium, Monterey, CA, USA, 1998.

[10] D. Conway, Object Oriented Perl, (Manning Publications, Greenwich, Co, USA, 2000).

[11] D. Donath, F. Petzold, Towards a building information system based on computer-supported surveying system, in Proceedings of. CAAD towards new design conventions, Białystok, Poland, 1997, 141-164.

[12] R. Ducournau, J. Euzenat, G.Masini, A. Napoli "Langages et modèles à objets. Etat des recherches et perspectives" INRIA Publications (Institut National de Recherche en Informatique et en Automatique), 1998, ISBN 2-7261-1131-9.

[13] I.Dudek, J.Y Blaise, 3D models as visual interfaces for Internet: an application to a multimedia documentation on architectural evolutions, in Proceedings of ICCVG 02 (Zakopane, Poland, September 2002), 250-256.

[14] I.Dudek, J.Y Blaise, I. Dudek, J.Y Blaise, Interpretative modelling as a tool in the investigation of the architectural heritage : information and visualisation issues in Proceedings of VIIP 2001, (Marbella, Spain, september 2001), 48-54.

[15] S. Francis, The importance of being abstract: an indian approach to models, in Proceedings. 17th eCAADe Conf. Turning to 2000, Liverpool, UK, 1999, 101-109.

[16] K. Hampton, The Perl XML Interfaces, published by O'Reilly on XML.com , 04 2001.

[17] H.Holten-Lund, M.Hvidtfeldt, J.Madsen, S.Pedersen, VRML visualization in a Surgery Planning and Diagnostics Application, in Proceedings VRML 2000, Monterey, CA, USA, 2000.

[18] B.Huber, Adding dimensions to GIS with VRML, issues in Proceedings of Pennsylvania GIS Conf., Hershey, US, 2000.

[19] J.Kantner, Realism vs Reality: creating virtual reconstructions of prehistoric architecture, in J.A Barcelo, M.Forte, D.H Sanders (Ed.) Virtual reality in archaeology, (Oxford: Archeopress 2000).

[20] O.Kodym, Usage of the VRML environment for the presentation of the underground mine, in Proceedings ICEE 1999, 10-14 08 1999, Ostrava, Czech Republic.

[21] S.Landes, Design and implementation of a web-based 3D campus Information System for the University of Karlsruhe, in Proceedings. COBRA 1998 Conference., Florianopolis, Brasil, 1998.

[22] D.Lowenthal, Visible cities, Harvard Design Magazine, Winter/Spring 2001, Number 13.

[23] M.Łukacz, Metrologia i pierwsza faza zabudowany staromiejskich bloków lokacyjnego Krakowa, in Proceedings. International Conference on Conservation, Kraków, Poland, 1998, 92-100.

[24] H. Nakamura, R. Homma, M. Morozumi, On the development of excavation support system, , in Proceedings 17th eCAADe Conf. Turning to 2000, Liverpool, UK, 1999, 341-348.

[25] R. Oxman, Visual Emergence in creative collaboration, in Proceedings 17th eCAADe Conf. Turning to 2000, Liverpool, UK, 1999, 357-363.

[26] B.Roehl, J.Couch, C.Reed-Ballreich, T.Rohaly, G.Brown, Late night VRML 2.0 with JAVA, Ziff-Davis Press, MacMillan Publishing, 1997.

[27] C. Russo Dos Santos, P.Gros, P.Abel, Dynamic Information visualization Using Metaphoric Worlds, in Proceedings. VIIP 2001, (Marbella, Spain, september 2001), 60-65.

[28] M. Schilcher, Z. Guo, M. Klaus, R. Roschlaub, Semi Automatic Derivation of 3D-City-Models from existing 2D-Geoinformationssystems, SIST, Wuhan, China, 1998.

[29] R. Stenvert, Constructing the past: computer-assisted Architectural-Historical Research (Utrecht : PHD of the University of Utrecht,1991).

[30] D. Whiting, S. Nickerson, Computer aided recording tools automate the creation of a site information system, in Proceedings. CIPA 1997, Göeteborg, Sweden, 1997, 121-130.