EXPLOITING THE ARCHITECTURAL HERITAGE'S DOCUMENTATION: METHODOLOGY AND TOOLS FOR DATA ANALYSIS AND VISUALISATION

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
Documentation analysis and organisation are vital to the researcher when trying to understand the evolution of patrimonial edifices and sites. Documentary sources provide partial evidences from which the researcher will infer possible scenarios on how an edifice may have been changed throughout the centuries. They are the only scientific basis from which virtual renderings can be proposed and justified. Still, in the field of the architectural heritage, there is a gap to fill between well established data management technologies that provide solutions for documentation handling, and geometric modelling techniques that underlie reconstruction efforts.
Documentation is organised with regards to what the documents are, books, illustrations, etc… Virtual renderings feature a geometry that bears no link to the documentation's analysis. Our contribution introduces a solution for attaching the documentation to architectural concepts that represent physical beings used in the edifice's structure, and this without modifications on existing documentation descriptions. Three dimensional scenes can then be used as one of the means to retrieve or visualise the information we hold on the edifice's or site's evolution. Our position is that the 3D representation of architectural objects can be an efficient filter on the set of data architects, conservators or archaeologists handle.

KEY WORDS
Architectural heritage, Object Persistence, XML parsing, Interfaces, Information visualisation, Web databases.

1. INTRODUCTION
Preservation of the architectural and urban heritage includes of course a concern for the edifice itself, , but it also includes a concern for the edifice's documentation. This documentation helps the researcher to try and state for instance how the edifice evolved through time or how the edifice was when nothing is left of it today. It provides *clues* that will need an *interpretation* from the researcher or conservator in order to *understand* the edifice or site's evolutions. Finally, a better understanding of the edifice or site's evolutions will help in decision-making when facing conservation challenges, may they concern physical interventions on remains or virtual reconstruction. In this process, the step of data interpretation is a critical one. What are the solutions one can base on in order to organise a documentation with regards to an architectural analysis of its content?

Numerous experiments, mainly in the field of archaeology, have been carried out using the paradigms of GIS such as [1], [2] or [3]. But in the case of architecture at its various scales, geometry cannot be considered as a relevant intermediate between the documentation and the edifice, as established by [4]. In parallel, realistic 3D models prove relevant with respect to communication goals, see for instance [5]. But communication through such realistic renderings does not serve the understanding of the edifice since it results in an abusive simplification, stressing the only morphology whereas it is not the only element that should be visualised, (see for instance [6] or [7]). We develop an approach in which what is "beyond" the image is more important that the image itself, in line with contribution like [8] or [9].

We observe that although the edifice is not the information, the information is relative to the edifice. This means that at least a quotation link exists between them. Our contribution investigates a possible solution that stands in-between formalisms dedicated to either database management or geometric modelling; in order to use physical elements of architecture as filters on the set of data architects, conservators or archaeologists handle. In contributions to the VIIP conference series[6][10], we have introduced some aspects connected with the specificity of patrimonial studies, and have investigated the possible uses of 3D visualisations when they are considered, not as representing the edifice, but as representing our interpretation of what it can have been like at each phase. In this paper we will focus on the methodology we develop for supporting the step of *data interpretation* that conservators, architects, archaeologist face when they critically examine the data collections they handle[11]. We will introduce data analysis issues in our field of experimentation, examine the role of architectural concepts in supporting this analysis, and detail various implementation aspects:
-Object Oriented formalism for the representation of the architectural concepts.

-XML/RDBMS persistence for the architectural concepts.
-Structure of the documentation database and link to the architectural concepts.
-VRML interfaces.

Our position is that 3D models of edifices or sites, considered as interpretations of our knowledge, can be efficient in retrieving information (i.e., documentation) about architectural evolutions.
[12] have introduced geometrical objects used as visual interfaces for data retrieval on urban facilities, [13] 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. Our proposal stands in between Geometric modelling tools and GIS platforms. We strive to use architectural beings in order to retrieve or visualise information where the above mentioned solutions use geometrical beings.

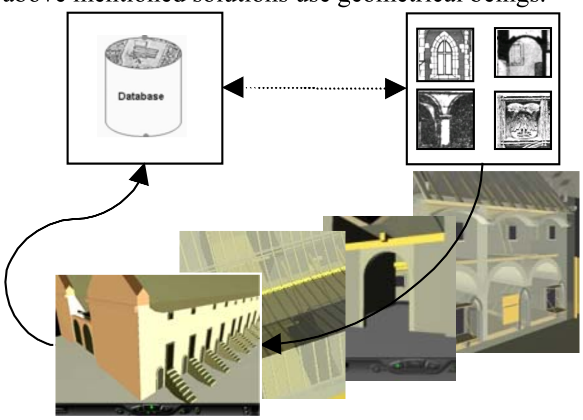

Figure 1 : Attaching information to architectural beings, the case of Kramy Bogate

2. DATA COLLECTION

The methodology used by historians of architecture and conservators in order to analyse evolutions of an architectural object is based on the interpretation and comparison of various types of documentation, as stated in the [14] charter. Therefore the idea that different *pieces of documentation are in relation to architectural elements* (a building, a portal, etc.), is for them a natural (although often unspoken) part of their work methodology.
One key goal of our research is to capture, capitalise and visualise the actual basis of this methodology –relations between an architectural artefact and the wide range of documents that refer to it. For example, in the studying of the historical evolutions of Krakow's Kramy Bogate[1] (definition of its hypothetical shape at the period of the construction, its successive modifications and date them) one should analyse and compare various historical documents, investigate comparative studies and analysis conducted by different authors. One should take into consideration technical and stylistic specificity of the local architectural production, examine archaeological and architectural investigations, study documents mentioning the object's function, structure, position, orientation and dimensions, its role in a wider architectural or urban context, etc.
Such a documentation is heterogeneous, it may vary from stereoscopic photographs, through records of the city belongings to pieces of poems. For us, each piece of documentation that may potentially have an influence on the course of the analysis and therefore an effect on the final hypothesis, is interconnected with the physical artifacts. Our experiences are focused on historical edifices located in Kraków (Poland). We gather in a database called SOL (Sources On Line) references of various historical documents connected to the urban fabric of Kraków' s Old Town and problems related to the disciplines of history of architecture, conservation and protection.
Following ideas of [8] the documents are described in two ways:

- *standard data identification* describing what the document is (author, edition, type of media, technique, etc..) in a way that is commonly used by the libraries, museums, etc.
- *interpretation of data-content* (morphology typology, etc.), used especially in art and architectural historical studies.

All types of relevant documentation are categorised in three groups: bibliographical documentation, iconographical and visual documentation and cartographic documentation. On one hand, from point of view of standard data identification, references in the SOL database are classified using name of author, document's title and type (e.g. manuscript, sketch, photograph, …), date of creation, etc.
On the other hand, our main interest being focused on urban and architectural evolution, what is required is a description that takes into account an interpretation of data or image content and therefore the object's morphology, the period that it describes, a typology, a position in a town etc. This part of description is a result of document's analysis. A formalism of thematic bias has been developed in order to add to each data sheet such feature. We deal therefore with two independent types of document's description. We separate them also in a database structure, as will be shown in section 7.

The documentation that we describe is stored in various national and private collections (libraries, museums, archives, ...). All of those units have their own classification and access polices. The decision of giving access to the digitised copies of particular pieces of documentation belongs to those institutions. Having it in mind we consciously avoid giving a direct access to digitised sources of various collections. Our proposal introduces a distributed computer architecture in which we only refer to pieces of information that are detained by various institutions. It has to be stressed that our goal is

[1] object destroyed in 1868

not to deliver the digitised copies of documents that can be used in the architectural analysis, but to *localise* them in terms of :

– In which library(ies) can they be found?
– To which architectural objects do they refer?

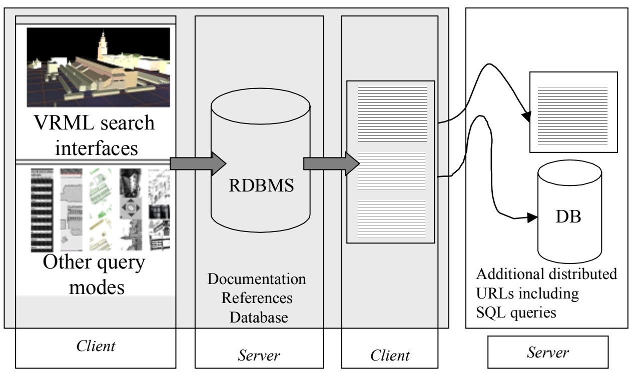

Figure 2 : Managing references and data distribution in the SOL database

3. DATA ANALYSIS ISSUES

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 [6] or [15].

But the documentation that serves as source of evidences is far from being exhaustive and non-ambiguous. What is more it is not structured with regards to the edifices or sites that it documents. We will therefore face several difficulties when wanting to implement a link between a document and an architectural object inside 3D scenes:

1. 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 shapes that in all cases are hypothetical. Consequently we will need to provide the scenes with graphical codes marking the evaluation of the hypothesis.
2. Investigating an edifice's evolution bases on a documentation and it's analysis. But this documentation varies in type, precision and relevance. We may face partial evidence, contradictory evidence, lack of evidences. It may also vary 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. We need to propose visual markings of the objects represented in a 3D scene that correspond to the type and content of their documentation .
3. 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. Scale can act as this complementary filter in the information available on the edifice We will need to distinguish the concepts to which correspond pieces of information with regards [4]'s notion of scale. This notion is oddly absent from the field of 3D modelling although its usability in the studying of the edifice has been established by [9], and although its usage is widely spread in the way architects analyse edifices.
4. 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, underlining another problem, this of localisation of architectural elements. It appears then necessary to produce representations that features architectural elements located in the space of the city at a given period of time.
5. In the field of architecture, 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 [16], [8] or [17]). Visualising with the third dimension is clearly established as a necessity. Consequently, we will need to isolate elements of morphology and represent them in order to use them as anchors on the documentation. But this can only be achieved if we build a theoretical model of architectural elements that will serve as intermediate between the user and the set of documents to interface.

In brief, when considering 3D scenes for use as interfaces, several issues are addressed:

– Marking the hypothetical nature of the scene.
– Showing what documents justify it.
– Handling multi-scale representations.
– Handling evolutions in time and space.
– Considering reusability problems in their making.

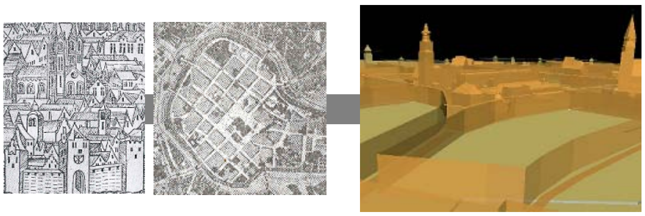

Figure 3 : Providing 3D scenes with graphical markings for evaluation purposes, the case of transparencies.

The solution we currently investigate introduces graphical disposals nested inside a scene, written in the VRML2.0 standard, and readable inside Internet navigators, in order to show our analysis of the documentation. Scenes are here given four roles:

- Represent an interpretation of the architectural morphology.
- Visualise each object's documentation's analysis through a graphical marking.
- Retrieve information on any object in the scene.
- Visualise, as a 3D scene, the result of a query on the documentation.

4. A HIERARCHY OF ARCHITECTURAL CONCEPTS

The concepts we use concepts as a mean to visualise information 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 [18]). More detailed description of our concept identification methodology can be found in the publications related to the ARKIW program[6][10][19].

Each concept is formalised in a class that contains six blocks of information. The Morphology block detains the information relative to the concept's geometrical features, it serves as the main division line inside the model. Documentation, Typology and Evolution information blocks are used in order to attach to each instance a set of qualifying attributes that we call justifiers, and that are used in order to monitor the actual appearance of the object. The Documentation block is responsible for the handling of SQL queries on the SOL database that references the documentation. The Typology and Evolution blocks store an evaluation of the documentation that will be carried inside the 3D scene, letting us for instance to visualise with different colour codes edifices for which we know architectural drawings exist those for which such documents do not exist.

5. PERSISTENCE OF INSTANCES

In our application domain objects are often reused or partly destroyed. This problem has been raised in works like [11]. We have as a consequence provided each object with persistence mechanism that store independently the object identity (identity + concept documentation + position in the model's structure) and its various states of evolution.

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.

Each concept detains methods relevant for persistence handling in XML files and RDBMS context. 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 [20].

It has to be stressed that autonomy and perenniality of the of the data sheets are of crucial importance in our application domain. We have chosen to store the textual results (XML sheets) of the model's instanciation inside standard ASCII files that can be used independently from the system as a whole. In our approach, solutions for Object persistence as those described in [18] are therefore not implemented since they implicate a dependence of the results on the application that gave birth to it. Good elements for a discussion on the XML one input / several outputs paradigm can be found in [21]. We propose in line with this author a solution based on the idea that a unique input- the instance's XML sheet; will have several outputs. In our case, at this stage of the development, the XML sheets store each instance's states in order to re-instance the objects but is also used as one of the pieces of information the user can retrieve from 3D scenes by the selection of an object (besides SQL queries on the instances database and on the documentation database). XML sheets being written by a call of method on each instance, its structure matches the concept's structure with regards to a position in the tree of classes. On the other hand, the RDBMS only supports the top class attributes : consequently there is no such relational / object mismatch issue as raised in [22].

6. CREATION AND USE OF 3D INTERFACES

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. Scenes can then be used in order to retrieve information about each object displayed, enabling it to become an interface.

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. An indication of

the documentation 's analysis (levels of certainty, type of documentation, typologies, etc.) can thereby be displayed natively or interactively inside the VRML scene.
3D scenes are used as a query mode (predefined time-related scenes) by selecting an object inside the 3D model. They can also be used 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 [18] 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.
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.
The pluses, minuses and possible applications of the VRML standard for architectural modelling have often been discussed, see for instance [16] or [23]. Our scenes are written in VRML 2.0 both for Cosmo and Cortona plug-ins. We have stressed the need to create scenes that would remain autonomous from the application that created them. By saying this, we rejected the possibility of investigating JAVA/VRML solutions (see [24]) that various experiences such as [12] or [25] have proven efficient, but that seem too exposed to versioning problems for use in our application domain.

7. DATABASES & IMPLEMENTATION

In the experiment we report here, we have used two MySQL databases : one stores qualitative states of each instance of the architectural model, another stores references to the documentation and links to the instances database (see sub-sections below). Our contribution does not stress one technology but investigates a possible combination of formalisms with regards to open content issues raised in [26] : OO modelling, XML technologies, VRML 2.0 standard, VR scenes / e-databases interfacing, RDBMS platform and CGI interfacing.Our experimental platform's interface uses a three-parts software architecture:

- a web server,
- a browser with VRML 2.0 plugin,
- a database server and/or XML files.

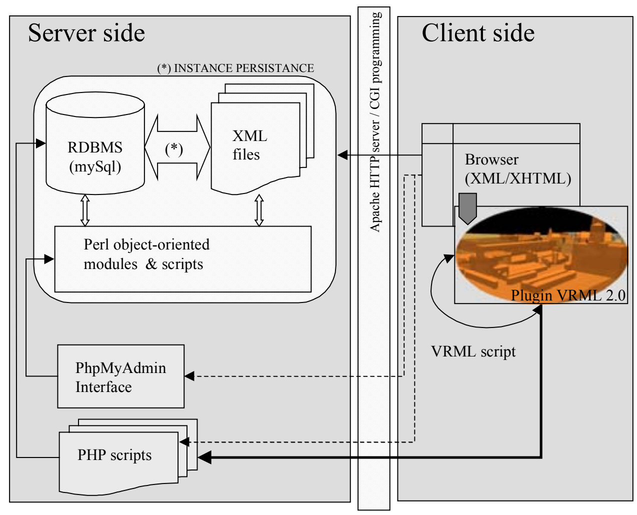

Figure 4 : Client / server System architecture and 3D interfaces

The formalisation of architectural concepts as well as the writing of XML/VRML files is monitored through PERL 5 modules. PERL is a scripting language but also an oriented object language that has proven efficiency, as mentioned in [18], and performances that remain acceptable in our research context.
In parallel, we developed the RDBMS interfacing through PHP modules. PHP is a widely-used scripting language that is especially suited for Web development and can be embedded into HTML. The Web server is the free of charge (GNU Licence) Apache, the most popular httpd server. We test our implementations on two browsers that support VRML 2.0 plugins, Opera and Internet Explorer.
The RDBMS we use is MySQL[2], described as the world's most popular Open Source Database. It is designed for speed, power and precision in mission critical, heavy load use (but at this time, this not our aim). The software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. Although they do exist, we will not here detail the limitations of MySQL since they are not obstacles in the

[2] (http://www.mysql.com)

current stage of our developments, but a good case study is given in [26]. The administration of the databases is supported by pMyAdmin[3], a tool written in PHP and intended to handle the administration of MySQL over a Web interface. Currently it can create and drop databases, create/drop/alter tables, delete/edit/add fields, execute any SQL statement, manage keys on fields, which is quite enough for us actually. The making of the user's interfaces to the databases is also PHP driven. The formatting of the results uses either the XML standard (extraction of data sets) or XHTML files from which the user can retrieve the 3D scene corresponding to the set of architectural objects matching his search criteria. The 3D scene (VRML) is computed online via a PERL script that instances each object, formats and appends a VRML file.
We in fact use the DB for its efficiency in processing queries on the set of objects, but once a query is processed the 3D scene featuring the selection of objects is computed online by Parsing the XML sheets in order to retrieve the morphological information that the DB does not detain. It has here to be stressed that the morphological definition of an object may vary once new elements of information are gathered on an object: it appears then as highly useful to disconnect the dimensional aspects and the informational aspects, the former benefiting from the flexibility of XML formalisms.

7.1 The instances' database

The database storing instances of architectural concepts (called VIA) closely matches the structure presented in section 4. The main table stores a unique identity for the instance. It defines fields for a UNIQUE-keyed Id, a name (and a link to a table of alternative names), a link to the Documentation table and a Period stored as two Integers (fixing an interval). The Periods table encompasses Localisation, Evolution and Typology information blocks. It defines an XYZ+ angle localisation, two periods (each stored as two Integers) and four Justifiers (each of them containing three textual values that are evaluations of the instance's documentation). The Periods table naturally defines links to the main table: each instance has a unique ID but it may have evolved through time. Morphological evolutions are stored in XML sheets, but *their justification* remains stored in the DB.
Whereas the Periods table fixes an evaluation of the instance's documentation, the Documentation table only states whether or not an instance is documented. In fact we have defined seven fields that received a value chosen in a predefined list of documents types.
The main table also defines a link to a table containing the structure of the model, meaning the hierarchical relations between concepts in the tree of classes; this in order to allow RDBMS-based searches not only by object type (here, object class), but also on a whole sub-hierarchy of the tree of classes (example: 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.)
This last table is updated by instances of the PERL "controller" hierarchy that we have developed for that purpose, and that maintain consistency between the OO architectural model and the DB. Methods are also implemented that ensure the consistency of data between data stored respectively in DB context and in XML sheets.

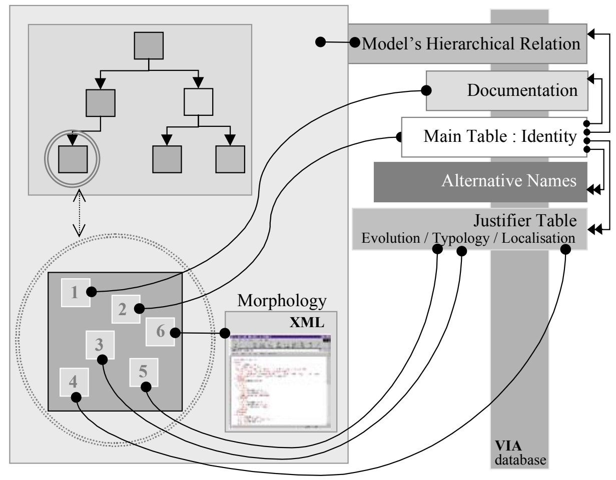

Figure 5 : The six information blocks of an architectural concepts and their relation to the VIA database

7.2 The documentation database

The SOL database describes what the data is: a book, a plan, a cloud of digitised points, etc. We attach this data to what it is about : an edifice, a part of an edifice, etc. It has to be mentioned that in our research context the architectural object exist whether or not we have metrical information on it. Its dimensions are for us only one element of its information. Consequently, we consider that data exists prior to our efforts of interpretation and of 3D representation. The 3D scenes do not then describe the data but will only figure our interpretation of the data.

As mentioned in section 2, we have distinguished *standard data identification* and *interpretation of data content*. The database contains tables supporting a standard indexing of documentary sources: document identification, author identification, graphics description. It also contains four tables which support the interpretation phase from various points of view:
-The *Availability* table states for each source whether or not it is available in the city's libraries.
-The *Localisation* table attaches an XYZ position + orientation in the city to graphics or text, potentially enabling searches in 2D graphics in our future developments.
-The *Instances* table attaches an architectural object stored in the VIA database to graphics or text. In this case as well as in the *Localisation* table what is attached is not a whole document but the part that is relevant (page number inside a book, figure number, etc.).

[3] (http://sourceforge.net/projects/phpmyadmin/)

-Finally, the *Comment* table attaches to entries in the Localisation or Instances tables additional information, notably URLs that can be used in the context of a distributed system architecture.

The following figure summarises links between the two databases we have implemented:

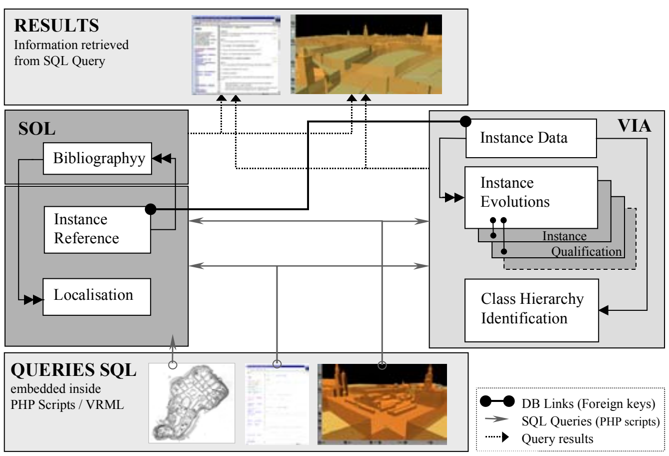

Figure 6 : Roles and links of the SOL and VIA databases

8. CONCLUSION

In the field of architectural conservation, documentation and representation play essential roles in the understanding of the edifice and of its evolutions. Documentation provides evidences for an analysis of the edifice's morphological evolution. 3D helps evaluating hypothesis on morphological evolutions. Moreover, 3D representations can help in visualising the documentation by providing a visual interpretation of the documentation's content.
Our proposal introduces a methodological approach, and investigates a possible technological solution, for an enhanced link between physical objects and what documents them: raw documentary sources. It provides an operational framework for the capitalisation (through database and XML formalisms) of this *interpretation phase* during which researchers establish meanings of raw data, and that derives from the reading of sources possible scenarios of morphological evolutions.
Although the documentation analysis step is a costly one, it is in our application a vital one, and it can greatly benefit from the memorisation we have tried to provide tools for.

9. ACKNOWLEDGEMENT

Our experiences are carried out in the framework of PICS and APN research programmes (CNRS / KBN, CNRS). The field of experimentation of our research is Kraków's historical centre, a UNESCO-listed urban ensemble of great architectural value and diversity.

REFERENCES

[1] C.Ioannidis, H.Chlepa, Spatial information system for the geometric documentation and restoration studies of monuments: an application to the wall of ancient Messene, *Proc. ISPRS WG V/5 Workshop* Thessaloniki, Greece, 7-9 juillet 1999.
[2] 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 Proc SFIIC 97 Conf.*, Chalon sur saone, France, 1997, pp 211-219
[3] A.Roberts, Virtual site planning, *in Proc Ecaade 1999*, Liverpool, UK, 1999, pp 442-448.
[4] P. Boudon, *Sur l'espace architectural, essai d'épistémologie de l'architecture*, (Dunod Publications, Paris, France, 1971).
[5] R. Burton, M.E Hitchen, P.G Bryan, Virtual Stonehenge : a fall from disgrace ?, *Proc. CAA 97 (Computer Applications in Archaeology)*, Birmingham, UK, 1997.
[6]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 Proc VIIP 2001*, Marbella, Spain, 2001, pp 48-54.
[7]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).
[8] R. Stenvert, Constructing the past: computer-assisted Architectural-Historical Research (Utrecht : PHD of the University of Utrecht,1991).
[9] 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).
[10] I.Dudek, J.Y Blaise, 3D models as visual interfaces for documenting the architectural heritage: the defensive system of Cracow, *in Proc VIIP 2002*, Malaga, Spain, 2002, pp 48-53.
[11] M. Łukacz, Metrologia i pierwsza faza zabudowany staromiejskich bloków lokacyjnego Krakowa, Proc. International Conf. on Conservation, Kraków, Poland, 1998, pp 92-100.
[12] S. Landes, Design and implementation of a web-based 3D campus Information System for the University of Karlsruhe, *Proc. COBRA 1998 Conf.*, Florianopolis, Brasil, 1988.
[13] D. Whiting, S. Nickerson, Computer aided recording tools automate the creation of a site information system, Proc. CIPA 1997, Göeteborg, Sweden, 1997, 121-130.
[14] Cracow Charter 2000, signed by representatives of conservation institutions, in Proc Int. Conf on Conservation Krakow 2000, pp191-193.
[15] D. Donath, F. Petzold, Towards a building information system based on computer-supported surveying system, in Proc. CAAD towards new design conventions, Białystok, Poland, 1997, 141-164.
[16] D.A Campbell, VRML in architectural construction documents : a case study, *in Proc 3rd VRML symposium*, Monterey, CA, USA, 1998.

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

[18] D. Conway, *Object Oriented Perl*, (Manning Publications, 2000).

[19] I.Dudek, J.Y Blaise, IT applications for architectural intervention and documentation in monuments' ensembles *Międzynarodowe Sympozjum Konserwatorskie,V Teoria i praktyka w ochronie zabytków architektury, zespołów i miejsc historycznych* , 1999, Kraków, POLAND.

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

[21] N.Walsh, XML: One Input – Many Outputs : a response to Hillesund, *in Journal of Digital Information, vol 3 issue 1, 2002*

[22] J. Chen, Q. Huang, Eliminating the impedance mismatch between relational systems and Object-Oriented programming languages, in *Database reengineering and interoperability*, *Proc 6th international Hong Kong computer society database* workshop, Honk Kong, 1995.

[23] R. Oxman, Visual Emergence in creative collaboration, *Proc. 17th eCAADe Conf. Turning to 200*0, Liverpool, UK, 1999, pp 357-363.

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

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

[26] B.Pfaffenberger, Why open content matters, *in Linux Journal*, www.linuxjournal.com, April 2001.

[27] P.Bryant, Choosing between competing Free/Open Source Databases : A case study*, contribution to the Open Source Schools*, opensourceschools.org, Feb 2002.