Cultural routes applying knowledge qualitative representation

M.J.Escalona, A. Reina, J.A. Ortega, J. Torres, R.M. Gasca, M.Mejías

Languages and Computer Systems Department
University of Seville
{escalona, reinaqu, ortega, jtorres, gasca, risoto}@lsi.us.es

Abstract. The recent interest in historical and cultural patrimony has made a great number of information systems arise in order to divulge information about patrimony. However, these information retrieval systems should be powerful enough to recover the concrete and precise information, but, at the same time, they should offer a simple interface that allows a computer novice to query the information needed. Supporting this idea, we think the use of knowledge qualitative representation could be very helpful to fulfill this goal, as it is shown in this paper. This article introduces how qualitative programming techniques and models can be used to help a user to choose a cultural route adequate for his needs in a simple way.

1. Introduction

Qualitative reasoning and its techniques have been frequently used, mainly in engineering. This techniques managed in conjunction with constraint satisfaction programming are applied in many different research areas. In this case, we are going to use it for the generation of cultural routes.

One of the main advantages that a user can find when he uses qualitative reasoning is that he can use its capacity and flexibility. Thus, the user can express complex constraints in a very simple way. These qualitative constraints will be considered to obtain the desired results.

Moreover, the technology progress and the use of Internet are leading to new needs, and new questions arise everyday. Only a few years ago these requirements weren't taken into account by companies and institutions when they showed the information via Internet. Nowadays, they have provoked that companies must offer simple, but at the same time, powerful query systems to retrieve the information the way the users want.

Information obtained using models can be quantitative, qualitative, or a mixture of both of them. This mixture is going to be referred as semi-qualitative in this paper. All these pieces of information must be considered when these models are been studied.

We have taken different levels of numeric abstraction into account, from a completely qualitative point of view [8], semi-qualitative [7], using numeric intervals [14] and quantitative.

As we have already mentioned, semi-qualitative knowledge representation can be used for many different purposes. This work applies this kind of knowledge in a real problem of a public company in Andalusia. However, it could be extended around other geographical areas. This public company is the "Historical Patrimony Andalusian Institute" (HPAI). This institute manages information about historical patrimony in Andalusia. Many different communities are interested in this kind of information. On the one hand, it is very important for patrimony researchers and cataloguers, as well as for the staff of the HPAI. And, on the other hand, it is very interesting for those people who are not expert in patrimony, like tourists. So, one of the main aims of HPAI is spreading the information out via Internet.

In order to bridge the gap between users and the information that the institute is offering, a system have been developed to allow users to make queries using a simple, but quite expressive language.

There are many different ways to offer this information, and this paper describes one of them. We want to give the user the possibility to plan his visit to Andalusia via Internet using a tool that allows him to enter some constraints about some characteristics of the monuments, such as, geographic location, style, historic period, and so on. This tool will study this constraints and it will elaborate a possible route. This route not only must satisfy the constraints that the user entered, but it has to be the most efficient according to time, prices and/or distance. The route will clarify which monuments should be visited and when they should be visited.

To introduce the problem and our results, in the second section of this work, we outline and describe the problem, the structure of the information showed to the user and the different constraints that the user can express.

In the third section, we give an introduction to programming with constraint paradigm and knowledge qualitative representation. To conclude, in the fourth section, we show how the problem is solved and the used methodology to do it. In the fifth section, we obtain some conclusions and point out some future works.

2. Outline of the problem

HPAI staff catalogue and manage a great amount of data about Andalusian historic patrimony. From its very beginnings, one of its main tasks has been the diffusion of information about patrimony to researching groups and people, in general. The procedure to access to the information was the following: the patrimony researcher requested the information via phone or via e-mail, and later, the answer was given by HPAI when it was possible. This procedure had many problems: only a few requests could be attended simultaneously, and the researchers hadn't immediate answers. These inconveniences became harder when the number of researchers rose. We are going to outline the problem offering a system that allows users to submit queries via Internet. This real-time system will be connected to the HPAI database, and will send the answers back immediately. The use of the Internet makes easier the information query by any given user.

2.1 Information structure

A great amount of data is stored to describe monuments. They use different media, like images, text, sound, etc. This information is grouped in different sets, but we are going to focus our attention in a few of them. They are described below:

- *Identification data set*. In this set, the user can find information like the name of the monument, its code, if it is a public monument or when it can be visited.
- *Location data set*. This is a very interesting set because it contains the information needed to express constraints. With this data set the user can determine where the monument is, that is, its address, its province, its town or its location in a map.
- *Description data set:* This set includes data which describes the monument. The most interesting ones are:
  - *Typology:* It indicates what kind of monument it is, a church, a cathedral, etc.
  - *Style:* This data describes the artistic style of the monuments, for example, Gothic or Baroque.
  - *Historic Period:* This data shows the historic period of each monument.
  - *Author:* This field groups the group of people who produced the monument.
  - *Description:* It is a text where the monument is described in detail.
  - *History:* It is also a text where the history of the monument is stored.
- *Image set:* This set groups all images where the monuments appear. This information is very interesting in our problem, because we want to show images of the each selected monument to the user.

Other sets of data are stored, like the bibliographic data set, that is, books or magazines that have been written about the monuments, or the conservation data set, but they are out of the scope of this paper.

Although this is the information stored in the HPAI database, we have realised that we need to enrich the geographical data in order to define the cultural routes and to offer all the functionality and semantics that the user hopes.

2.2 Requirements and query language

When requirements are going to be defined, we should think about which the users' needs are. The system should have an interface where the user can enter the basic information for his requests. Basically, the screen will display the following data:

1- Initial town: In this field the user will enter the Andalusian town where he is going to start his visit.
2- Final town: Here the user will indicate the Andalusian town where his visit will finish.
3- Kilometres: The user will establish the length of his route. Usually, this parameter can not be determined precisely, therefore, we have enriched this field with a few labels that are more expressive. These labels are shown in table 1. As a consequence of the use of these labels, the user can request that the distance should be much less than 300 km or approximately 200 km.

4- Price: It indicates the amount of money that the user wants to spend visiting the monuments. This price do not include the journey price, because this one will depend on the transport used (rented car, own car, bus, train, plane, etc). The constraints about price can be defined flexibly, too, that is, the user can express them by means of the labels shown in Table 1.

5- Visitors who are more demanding about patrimony could be interested in describing the kind of monuments where they wanted to go. Consequently, the user needs to express constraints about the following things:

   a. Typology. The user will be able to enter if he is interested in cathedrals or paintings.
   b. Historic Periods. The user will be able to choose the historic period, for example, Middle Ages.
   c. Style. The user will be allowed to express constraints about the style of the monuments.
   d. Authors. At last, the user will be able to indicate the monument's author or authors.

   All these fields are stored in the HPAI database. Thus, when the user wants to choose a typology, the system will show all the typologies allowed.

6- Those visitors, who are interested in visiting only some towns, are allowed to use another field to specify the towns they wanted to know.

Although these are the basic requests and the ones which we have thought of, if a tool to generate cultural routes wants to be done, it will be necessary to include new aspects. We will show future upgrades for the route production program. Currently, we have only taken into account the ones mentioned above.

Once we have discussed the different possibilities that the user has, we should give him a simple way of expressing the constrains that he wants. In order to do this, the user will watch a screen similar to the one depicted in Figure 1.

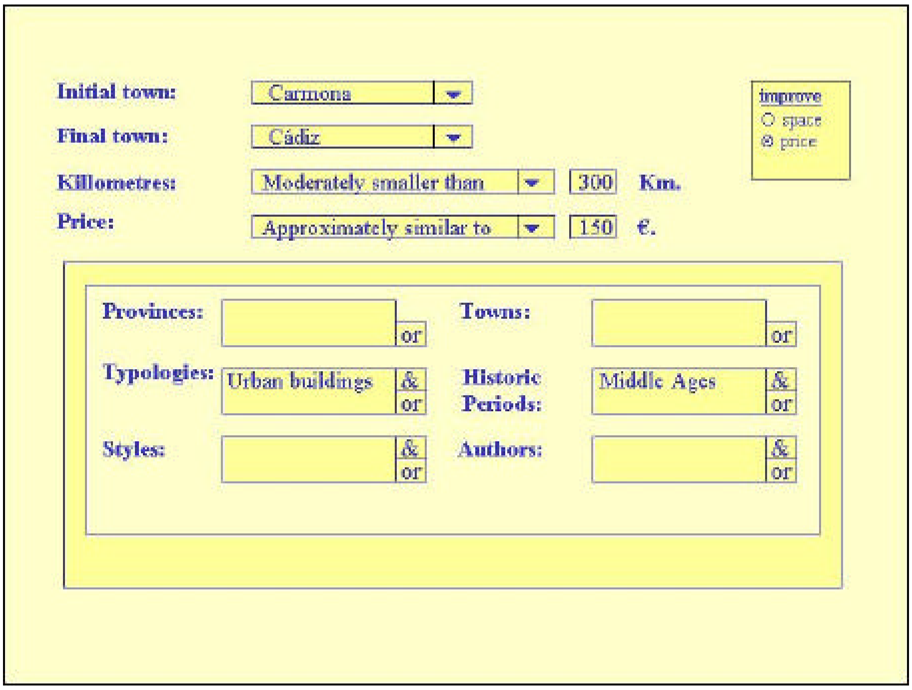

Figure 1: Screen to enter queries

In this screen, the user will enter the values of the fields he wanted to constraint, the Andalusian town where he wants to start the visit, the distance he wanted to travel and the time constraints. As it is shown, these parameters can be seen from a semi-qualitative point of view, that is, mixing qualitative and quantitative information [10] by means of the defined labels. The user is allowed to enter qualitative labels with quantitative information.

In this screen the user can prioritize its preferences according to the parameters: time, space or price (this is on the right and on the top of the screen). This selection will be very important because it will imply what the objective function will be to the constraint satisfaction algorithm.

Concerning to the bottom of the screen, the application have to let the user enter more than one value in every field, and let him express advanced queries, too. For example, the user could decide that he doesn't want to cover more than 150 km, and visit monuments from the Middle Ages which are urban buildings. Moreover, he could want to start his visit in Carmona, a town in Seville. This query is the one shown in Figure 1.

We should highlight that the constraints involving towns and provinces are disjunctive, because one monument can not be in two towns or provinces. In addiction to the selection of the province and the town, users can constraint the typology, the style, the authors and the historic period. These constraints could be declared disjunctive (using the `or` button placed on the right of the field) or conjunctive (by means of the `and` button). This decision will depend on the search criteria that the user wants to introduce.

As we can see in Figure 1, the basic functional requirements that we have mentioned below are represented in this screen.

2.3 Showing the results

Once the user has sent the query, the system will execute the pertinent operations on the specified constraints, as we are going to present in the next section, and, then, will return the results that will be displayed using a screen similar to the one showed in Figure 2.

On this screen, the user can read the information as a text (as we see on the top and on the right) and as a graph, using a map of Andalusia. Moreover, if the user wants to get more detailed information about his route, he can do it using the link of the bottom, for example, he can obtain all the details about the times of monuments, its characteristics, the distance,

The user can obtain more detailed information about every concrete monument if he navigates using the links of the cultural route. He can watch the description of the monument, its history, photos and images, its time, etc.

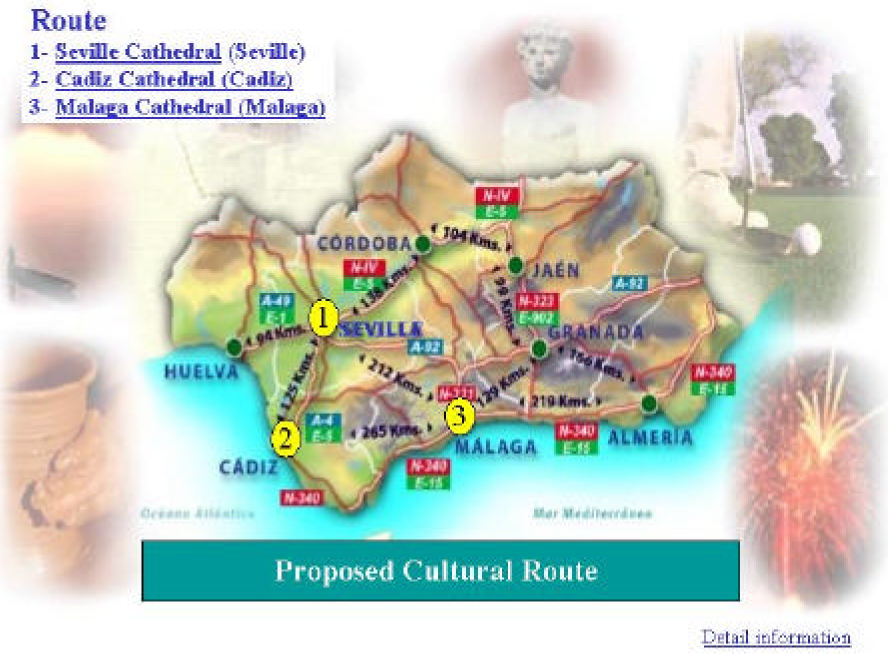

Figure 2: Query results screen

2.4 Work algorithm

To continue, we are going to describe the algorithm followed by the system to generate the results, as the ones obtained in the previous section. Once the user has entered his preferences and has requested the route, the system must follow these steps:

Step 1- The system will generate the constraints by means of the semi-qualitative labels that the user had selected. Thus, we are representing knowledge using quantitative constraints with intervals.

Step 2- The geographical area is limited by the system. If the user has selected a starting town or province and he has limited the number of kilometres, the system will obtain the utm x and y and will select from the database all those monuments whose distance is that number of kilometres or less. So that, it has to query the database and make mathematical calculations to calculate distances, and the total price when they have been constrained. You should notice that every town will be represented by an utm point (x, y) which is referred to the city centre. Moreover, the system will treat the monuments and towns as points not as irregular-surface polygons, as they really are. Although we could have worked with polygons in a more precisely way, the system would have been more inefficient and the results wouldn't have been improved so much, as we have proved in earlier experiences.

Step 3- After this, the system has to query the datatabase. When step 2 is over, the system will join all constraints and get the query which will be sent to the database system to get all the monuments that fulfil the constraints.

Step 4- Show the results. The system will expose the concrete route with the monuments resulting from the query. For this, the system will implement algorithms to put monuments into a map and highlight the possible routes to follow. (Figure 2).

3 Constraint programming and qualitative knowledge

Constraint programming idea begun in the 70's as a way to solve constraint satisfaction problems that had been arisen in the processing image research field. The key ideas of the constraint programming can be found in [11] [3].

The programming with constraints is concerned with the process of solving problems that are specified using a set of constraints. The most common method for specifying constraints is as a CSP (Constraint Satisfaction Problem). A CSP is defined as a group of variables. Each of them can have a group of values (its domain) and a group of constraints (relations) among them. The solution of a CSP is a group of variable-value pairs that fulfil the constraints. CSP's deal with variables with discrete and finite domains. CSP's are solved by means of consistency techniques in addition to backtracking.

This programming paradigm allows the formulation of knowledge as a group of constraints without specifying a method to fulfil these constraints. A lot of techniques have been developed in literature in order to find partial or complete solutions to different kinds of constraint expressions.

This technique has been applied to different tasks, such as, design, diagnosis, truth maintenance, planning, scheduling, space-time reasoning, logic programming and user interfaces. Many of them are NP-complete problems and require time of exponential order to solve them. In this paper the reasoning that the program will do to find the constrained cultural routes will be based on the resolution of a CSP using constraint programming. The obtained solutions will be complete, valid and stable.

3.1 Constraint Satisfaction Problems

CSP's, as we have already mentioned, are composed by a group of variables, each of them with a certain domain (discrete, continuous and united), and a set of constraints that have the same priority (in some problems, some constraints can be prioritize), but different nature: logic, algebraic, etc.

Constraints represent requirements or conditions that solutions must fulfil. A constraint [11] can be described as a relation in a space of possibilities. From a mathematical point of view, they are relationships that are specified precisely among variables, each one getting values in a predefined domain. Constraints limit the possible values that the variables can have and represent some partial information about some variables.

Constraint networks are used to solve CSP problems and they are defined by:

$$V = \{X_1,..,X_n\}$$
$$D = \{D_1,..,D_n\}$$
$$C = \{C_1,...C_s\}$$

where V is the set of variables in the network, and X_i is its possible values which are in a defined domain, Di. C represents the set of constraints in the network. In each constraint C_i are some variables defined in V. Thus, each constraint is relations defined in a subset of V. Tuples in this relation are those ones. Each group of possible values in this subset is a possible solution. To conclude, a constraint is a relation that must be satisfied by possible values of variables in it.

3.2 Qualitative knowledge

In our problem, we can introduce some qualitative labels which have a special meaning in the system, in such a way that using the knowledge of an expert, these labels can be transformed into constraints. These constraints will allow us to define a CSP.

In our posed problem, some qualitative labels are shown to the user. These labels are going to determine set of values, but the user will be unaware of all this process. The interval of values defined for al qualitative label will be selected by the patrimony expert. In Table 1, labels defined for our problem are shown. Each label defines an interval. This interval has parameters, as it is shown in Table 1. As we see, each label defines an interval. Some parameters are used to define these intervals. In Section 4, we present the values of these parameters, which have been defined by the staff of the HPAI.

Defined labels can be classified depending on their associated operators. Thus, these operators can be unary or binary. Binary operators are divided into operators connected with the subtraction and operators related to the division. Each operator has a label and an associated interval, which are defined using some parameters which will be defined by the expert.

Table 1 depicts these labels and its associated values related to each operator.

Unary operators		
Qualitative label	**Operator**	**Domain**
very small/ very nearby	NG	[-v? ,-?]
moderately small/ moderately nearby	NM	[-?????]
lightly small/lightly nearby	NP	[????]
medium	0	0
lightly big/lightly far	PP	[0,?? ?
moderately big/ moderately far	PM	[????]
very big /very far	PG	[? ?v?]
Binary operator related with the difference		
Qualitative label	**Operator**	**Domain**
bigger or similar	≥	[0, v?]
similar	=	0
smaller or similar	≤	[-v? ,0]
Binary operator related with the division		
Qualitative label	**Operator**	**Domain**
quite smaller than/ very nearby to	‹‹	[0, Pre]
moderately smaller than/ moderately nearby to	-<	[Pre, $(1+Pre)^{-1}$]
lightly small than/lightly nearby to	~<	[$(1+Pre)^{-1}$, 1]
Approximalely similar to	˜	[$(1+Pre)^{-1}$, 1+Pre]
lightly bigger than/lightly far to	>~	[1, 1+Pre]
moderately bigger than/ moderately far to	>-	[1+Pre, 1/Pre]
quite bigger than /very far to	››	[1/Pre, v?]

Table 1: defined labels and operators

Unary operator intervals have been defined according to [12]. ? ?y?? are used to distinguish domains where qualitative magnitudes can get values. Binary operators are also represented by an interval [9], which is defined using a special tolerance parameter `Pre`. This parameter represents a value "much lower than one". In both operators, v? represents the infinity. Obviously, when these parameters are defined, helped by experts, it is necessary to study relations between unary and binary operators to be coherent [13].

4 Generating cultural routes

4.1 Requirement specification of cultural routes

The problem of finding cultural routes is solved by means of a graph, where its nodes represent monuments that fulfil a set of constraints, and its edges are the paths between monuments. In this section, we are going to determine normalized constraints defining possible predicates used to generate cultural routes. These predicates can be concrete or qualitative.

?? *Concrete predicates*: These represent constraints in fields which only can take values in a concrete domain. For example, in magnitudes like typology, style, or historic period.

`Style(X) = 'Gothic'`, it is true when x is a Gothic monument.

?? *Qualitative predicates*: These express constraints where qualitative labels in table 1 can be used. These are associated to continuous magnitudes like time, space, etc. For example:

`Distance moderately lower than 300`, it is true when the distance is in the interval defined by the label `moderately lower`.

Also, these predicates can be combined with classic logic predicates like *and, or,* or *not*.

To conclude, when a user submits a query, the system generates constraints associated to discrete magnitudes, named r_D. But, on the other hand, it also has to generate constraints related to continuous magnitudes, named r_C. Finally, the system combines both of them using logic operators, getting the syntax of the problem.

After defining the syntax, semantics must be established. Semantics gives rules to convert predicates in a CSP, which will be solved by means of ILOG Solver [5]. Concrete predicates are included in the continuous model adding a new variable *s=CSPDiscrete* (r_D). This variable is 1 when there is a possible route which verifies this discrete constraint. That is, the constraint is *s? ?* , which means that s is not an empty set.

Continuous constraints are defined using qualitative labels, thus, their semantics is more difficult to get. Some transformation rules are necessary to get it. These rules are depicted in Figure 3.

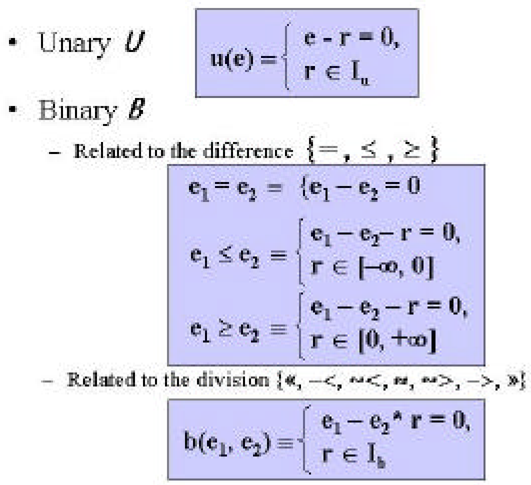

Figure 3: Transformation rules

In order to get the interval, the expert must establish the values of the parameters used in Table 1. The values defined by patrimony experts for our problem are shown in table 2.

		Campos (o magnitudes)			
Operator	Parameter	Province	Town.	Km.	Price
Unary	? ?			20	10
	? ?			300	300
Binaries	Pre	0.1	0.1	0.2	0.2

Table 2: operators, fields and parameter values

Shadowed cells indicate that the label does not have sense in that field. For example, if we have the constraint `Space moderately lower than 300` using rules in Figure 3 and Tables 1 and 2, we get the following constraint:

`Space moderately lower than 300` => space-300 * r, r?? [0, 0.2]

4.2 Methodology to get routes

When constraints are known, an algorithm to solve them has to be implemented. A generic algorithm could be used. However, numeric and discrete constraints appear in our posed problem, so a methodology has to be proposed. In figure 4, this methodology structure is presented.

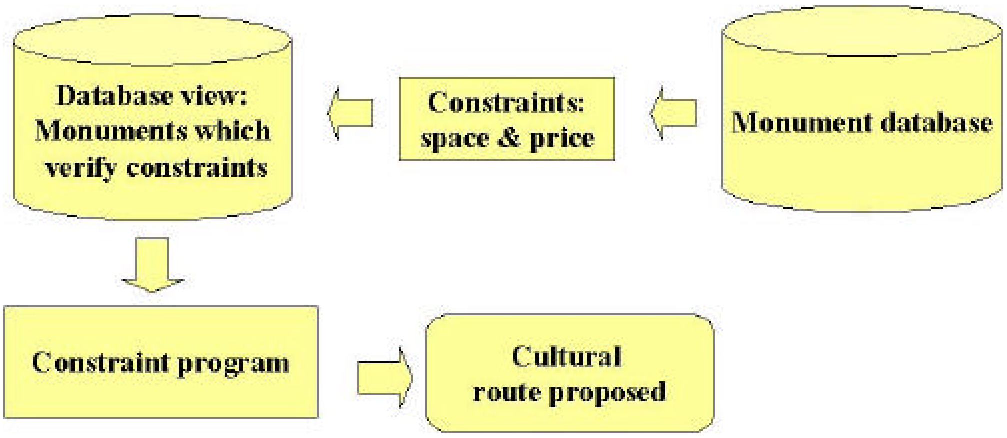

Figure 4: Methodology structure

The process starts in the monument HPAI database. The system does a previous request to get monuments which are situated in a suitable geographical area to be included in the cultural route. As the same, they verify geographical constraints. The monument database is a huge database which has a lot data, so this previous request allows its reduction. After that, the system gets a view of the initial database which is much lower than the original. After being selected these monuments, the suitable program, which is written in ILOG Solver, is developed. This program is applied to the database and allows the obtaining of the most interesting monuments for the user. Finally, an algorithm to put these monuments into the map and to define the best route is applied (similar to figure 2)

5. Conclusions and future works

When the problem is known, the methodology is defined and different groups of constraints are detected, it is not difficult to get a system to elaborate automatic cultural routes using the programming with constraint tools and programming languages is not difficult. Thus, we can claim that this paradigm is very suitable to this kind of problems.

However, this problem is more difficult than the version presented in this paper. There are a lot of aspects which have not being solved. One of them is the interface elaboration. To get a suitable and complete interface, also easy to be understood by the user and useful to everybody, is an open point in this work. In [1] and [3] this problem is analysed but their propositions have to be adapted to our problem.

Also, this problem could be enlarged to offer better results. For example, it could be interesting to extend the interface to allow doing questions about transports or about the different routes or highways which could be used in the cultural area. Also information about spent days in the cultural routes or total price (including the trip and lodgings) could be offered.

Nowadays, only the presented tool is being developed. It has a database management system which works with a view of the complete monument system in HPAI. Using Java Swing and JASP, the user interface is being designed and the system to solve constraints is made with ILOG Solver. However, as a future work, we want to enlarge it to get a more suitable and useful result.

References

1. P. Castells, F.Saiz, R. Morrión, F. García. *Modelización y Diseño Interactivo de Interfaces con estructura dinámica*. I Jornadas de Interacción Persona-Ordenador. Granada, Junio de 2000.
2. M.J.Escalona, M.Mejías, J.Torres. *Propuesta metológica para el desarrollo de sistemas para el tratamiento de bibliotecas digitales*. Informe interno del departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. Sevilla, Julio de 2000.
3. E.C. Freuder and Wallace M. Constraints IEEE Inteligent Systems Enero/Feb 2000
4. M.González, M. Mejías, M.J.Escalona, J.A.Ortega, R.M.Gasca. *Interacción con los usuarios en bibliotecas digitales*. I Jornadas Dolmen. Sevilla, Junio de 2001.
5. ILOG. *ILOG Solver 4.4. User's manual*. ILOG products 1999.
6. M. I. Jarvis. P. Shen Q. 2000 *Flexible Graphplan ECAI2000*, pp 506-510.
7. H.Kay, *Refining imprecise models and their behaviours*. Ph.D. Thesis. University of Texas (USA) 1996.
8. B.J. Kuipers. *Qualitative reasoning. Modeling and simulation with incomplete knowledge*. The Mit Press. Cambridge, Massachussetts 1994
9. M.L. Mavrovouniostis, G.Stephanopoulos. *Formal order-of-magnitude reasoning. Readings in Qualitative Reasoning about PhysicalSystems page. 323-336.*
10. J.A. Ortega. *Patrones de comportamiento temporal en modelos semicualitativos con restricciones.*Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. Abril 2000.
11. Van Hentenryck P. and Saraswat et al. 1996 *Strategic Directions in Constraint Programming* ACM Computing Surveys Vol. 28 Nº 4 Diciembre
12. L. Travé-Massuyès, Ph. Dague, F. Gerrin. *Le raisonement qualitativ pour les sciences de l´ingenieur. Herme Ed., Paris 1997.*
13. L. Travé-Massuyès, F. Prats, M.Sanchez, N.Agell. *Consistent Relative Absolute Order-of-Magnitude Models*. Por aparar ecer en QR2002. Barcelona, Junio 2002.
14. M.Vescovi, A. Fraguhar, Y. Iwasaki. *Numerical interval simulation: combined qualitative and quantitative simulation to bound behaviours of non-monotic systems.* Proc. 14th Int. Joint Conference on Artificial Intelligence pages 1806-1812. Canada 1995.