INTELLIGENT SOLUTION EVALUATION BASED ON ALTERNATIVE USER PROFILES

Georgios Bardis
Laboratoire Méthodes et Structures Informatiques – MSI, Faculté des Sciences, Université de Limoges
83, rue d'Isle, 87060 Limoges cedex, France

Technological Education Institute of Athens, Department of Computer Science,
Ag.Spyridonos St., 122 10 Egaleo, GREECE

Georgios Miaoulis
Technological Education Institute of Athens, Department of Computer Science,
Ag.Spyridonos St., 122 10 Egaleo, GREECE

Dimitri Plemenos
Laboratoire Méthodes et Structures Informatiques – MSI, Faculté des Sciences, Université de Limoges
83, rue d'Isle, 87060 Limoges cedex, France

Keywords: Machine Learning, Multicriteria Decision Making, User Modeling

Abstract: The MultiCAD platform is a system that accepts the declarative description of a scene (e.g. a building) as input and generates the geometric descriptions that comply with the specific description. Its goal is to facilitate the transition from the intuitive hierarchical decomposition of the scene to its concrete geometric representation. The aim of the present work is to provide the existing system with an intelligent module that will capture, store and apply user preferences in order to eventually automate the task of solution selection. A combination of two components based on decision support and artificial intelligence methodologies respectively are currently being implemented. A method is also proposed for the fair and efficient comparison of the results.

1 INTRODUCTION

The continuously increasing performance of modern computer hardware has made available software features that used to be prohibitive in terms of required time and complexity a few years ago. System developers are now not only willing but also able to design and implement powerful environments, rich in characteristics, capable of producing vast numbers of results in limited time. Nevertheless, the diversity of the user basis as well as the increased complexity and power of software systems call for intelligent features that will adapt the environment to each user's characteristics. Personalized system behavior with respect to an individual user's profile facilitates its use, increases both user and system efficiency and improves quality of the results.

Adoption of user preferences for intelligent system response has been presented in numerous efforts in the area of hypermedia and the WWW, e.g. (Brusilovsky 01), (Chen 02), (Pazzani 97), (Soltysiak 98). Incorporation of user preferences in geometric representations is presented in (Essert-Villard 00), where the user submits a set of constants together with a sketch from which the system extracts additional solution restrictions as well as in (Joan-Arinyo 03) where a genetic algorithm is periodically aided by the user to produce solutions closer to the latter's preferences.

On the other hand, the notion of multicriteria evaluation of building assemblies has also been

Bardis G., Miaoulis G. and Plemenos D. (2005).
INTELLIGENT SOLUTION EVALUATION BASED ON ALTERNATIVE USER PROFILES.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 74-82
DOI: 10.5220/0002523900740082

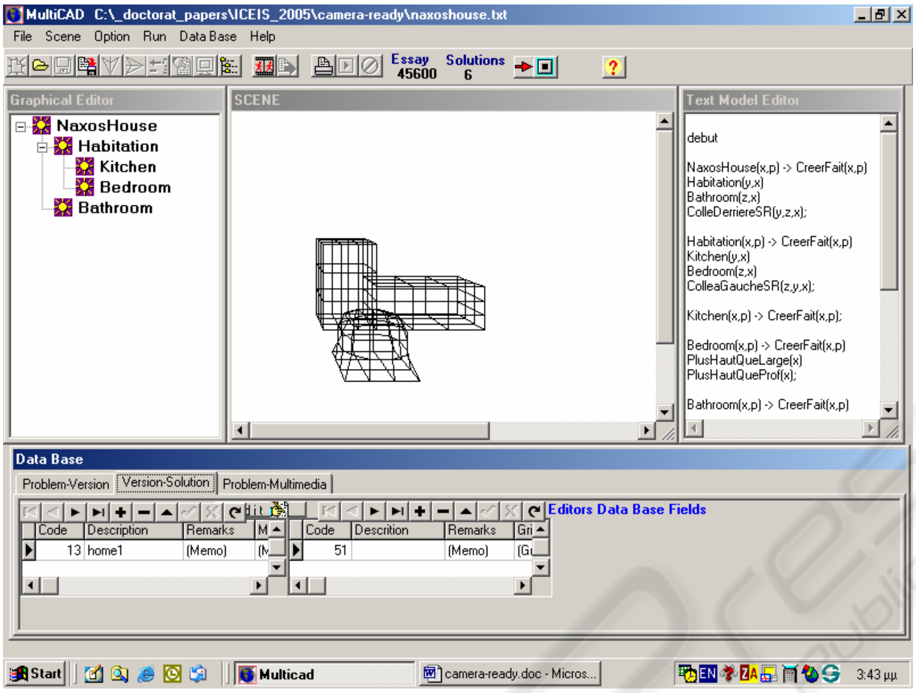

Figure 1: A Typical MultiCAD Session

presented in (Nassar 03), based on AHP calculated weights and a heuristic evaluation algorithm, but machine learning and user preferences have not been discussed.

Our work proposes a component for intelligent solution evaluation according to user's preferences in a declarative description environment. The proposed component combines multicriteria decision support and machine learning techniques for user modeling requiring only qualitative feedback on behalf of the user instead of exact geometric properties.

2 THE MULTICAD ENVIRONMENT

The MultiCAD system, presented in detail in (Miaoulis 02), was introduced as a platform supporting declarative object modeling (Plemenos 95), thus assisting the transition from the intuitive to the geometric object representation. The described system is a complete design environment including modules for validation of the object description, storage and maintenance of the solutions produced, etc. Subsequent works have successfully implemented most of the described modules as well as additional ones that have evolved from this initial design – solution generation using CSP (Bonnefoi 02) or genetic algorithms (Vassilas 02), concept modeling and ontology (Ravani 04), incorporation of architectural styles (Makris 03) and collaborative design (Golfinopoulos 04). Solution evaluation based on user preferences in this context was first introduced in (Plemenos 02) proposing a system based on the representation of each scene by a dedicated neural network. A new approach towards a user profile module was presented in (Bardis 04) describing two methods for user modeling and solution evaluation: a method based on the multicriteria nature of the problem and one relying on a neural network for the representation of each user's preferences.

The current work continues towards this direction by presenting the final design of the specific module – an approach incorporating multicriteria decision support and machine learning techniques – and the current stage of the implementation. Moreover, a set of criteria regarding alternative methods performance is introduced, that will serve as the basis for future testing and adjustment of the implemented module.

3 THE INTELLIGENT USER PROFILE MODULE

Figure 1 shows a typical session of the current implementation of MultiCAD where the declarative description of a scene appears together with the

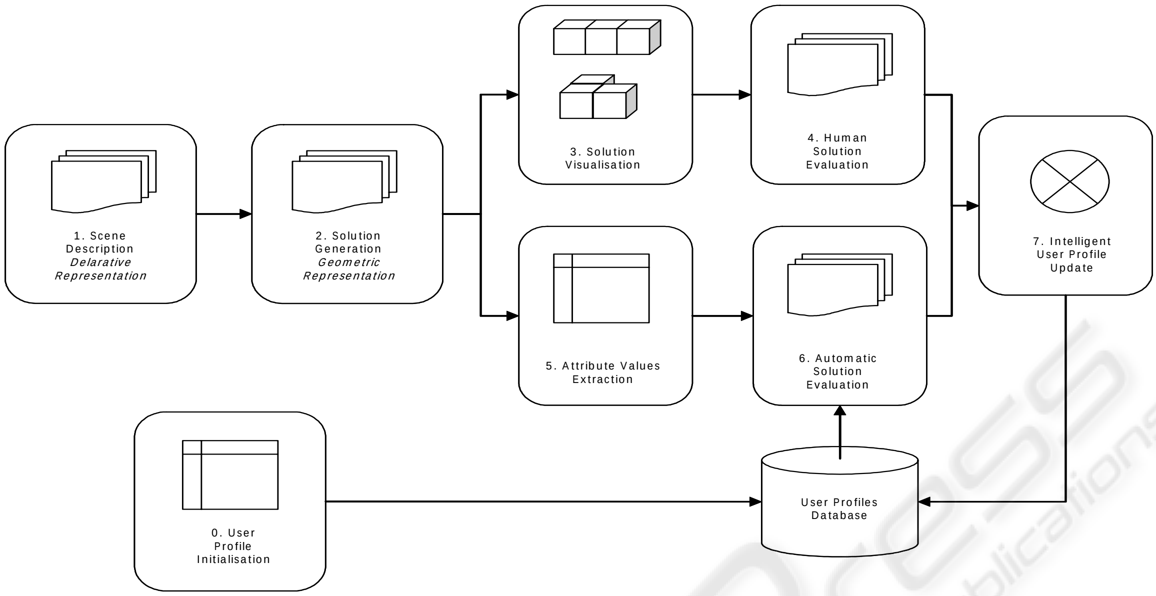

Figure 2: User Profile Module - Block Diagram

visualization of one of the corresponding solutions. The diagram in Figure 2 concentrates on the integration of the User Profile Module to the MultiCAD platform.

This version of the system applies user profile information only after the geometric representations of the described objects have been generated, i.e. not during their generation. The current stage of our work focuses on the construction of those modules that are immediately connected to the user profile component of the system.

Each declarative scene description may lead to a few thousands geometric representations complying with this description: the solutions. However, not all solutions are equally preferred by the user. Our intention is to eliminate those solutions not conforming to the user's preferences. Ideally, this has to happen with minimal or no user intervention. In particular, optimal user profile incorporation to the solution visualization process will maximize solution visualization throughput (SVT) (Bardis 04) and minimize the user intervention at later stages. In order to achieve this we have to resolve the following inter-connected problems:

- Solution representation and evaluation.
- User preferences modeling and representation.

4 SOLUTION REPRESENTATION

The geometric representation of each solution is translated to a set of attributes. This is a need that arises mainly from the fact that we have to reduce the complexity of the representation of each solution in order to be able to submit it as input to a neural network. In addition, this approach complies with the multicriteria decision methodologies (Vincke 92), (Goodwin 98), in particular, their requirement to request the user’s evaluation through a limited set of object attributes instead of numerous geometric properties. We choose to observe a minimal set of attributes (Fribault 03), (Bardis 04). This set will be extended or revised in the future, since, for the moment, we care more for the development of a prototype covering all stages of the MultiCAD cycle (Miaoulis 02) instead of capturing all possible aspects of user preferences with respect to a building assembly. The attributes we have chosen are based on geometric characteristics of each solution and, therefore, can be easily extracted by its geometric representation.

In particular, the observed attributes for any solution S_i are:

BD_i = Number of bedrooms
BT_i = Number of bathrooms
NA_i = Night-zone area
DA_i = Day-zone area
NDS_i = Night-zone / Day-zone separation
SWB_i = Existence of at least one south-western bedroom

Therefore, each solution S_i is represented by a vector of values:

S_i = (BD_i , BT_i , NA_i , DA_i , NDS_i , SWB_i),
for example
S_i = (2, 3, 52.4, 40.8, Partial, No).

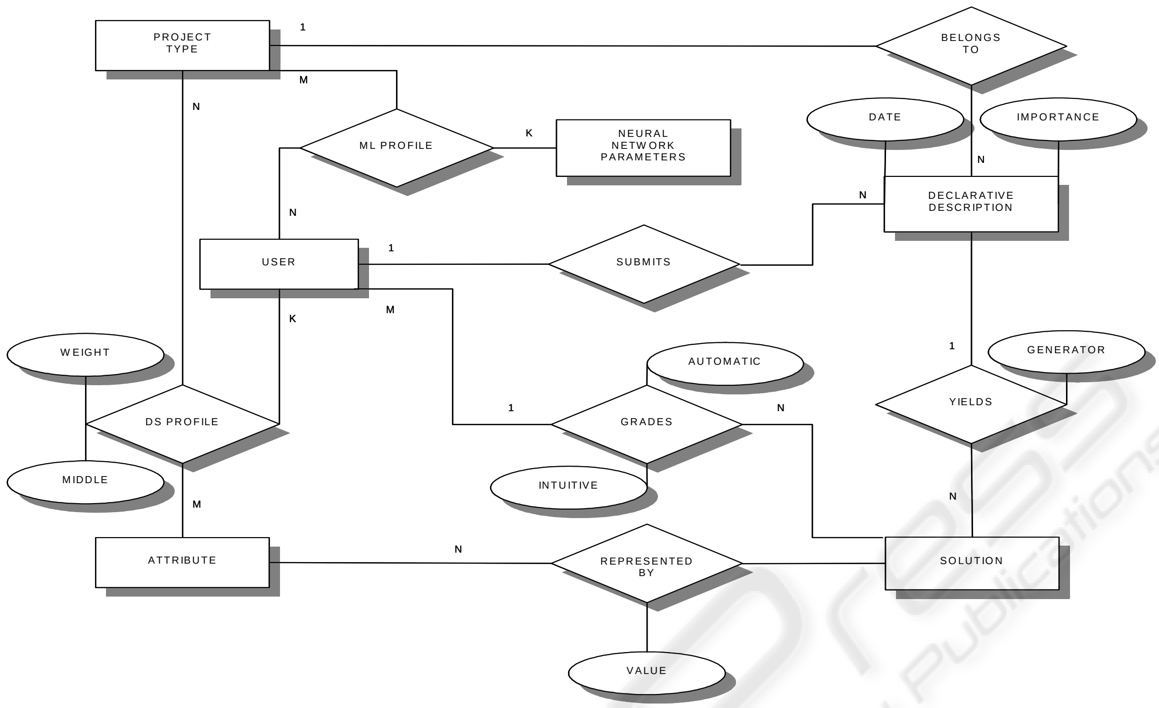

Figure 3: User Profile Module - Entity Relationship Model

The exact range of values for each observed attribute will be affected by the design of the ML component described in Section 6 as well as the solution generator used. Nevertheless, it is important to observe that solution generation of the existing system is not based on observed attributes. This implies that the number of generated solutions is not restricted by the range of values of the observed attributes. The observed attributes map each generated solution to a vector of values of restricted range. Thus, it may be the case that, at the present stage, two or more different solutions, i.e. of different geometric representations, are mapped to the same vector of observed attribute values.

5 USER MODELING

Figure 3 presents the database model that has been developed in order to store and maintain user profile information. Notice that only a few representative properties of entities and relationships appear in the ER graph. Apart from the entities directly connected with the user profile module, the database model also includes entities representing scene descriptions and geometric representations of the corresponding solutions, namely the DESCRIPTIONS and SOLUTIONS entities. Specialized databases have already been developed as part of other work, taking place in the context of the MultiCAD platform (Ravani 03). The database model proposed here is flexible enough to cooperate with these already existing structures and yet able to incorporate alternative implementations in case these become available in the future.

Figure 2 offers insight regarding the choice of the specific entities and relationships appearing in the database model. In particular,

USER. Example fields for the properties for the specific entity are the (unique) User Name, Password, First/Last Name, etc.

PROJECT TYPE. A text name plus extra information connecting the specific project type with the corresponding description prototypes contained in alternative databases of the MultiCAD environment.

DESCRIPTION. This entity represents the Prolog-like description of a scene. Importance represents the influence of the results of this session, i.e. the properties of the approved solutions during the specific this session, to the overall user profile for the specific project type.

SOLUTION. Full geometric representation of each solution.

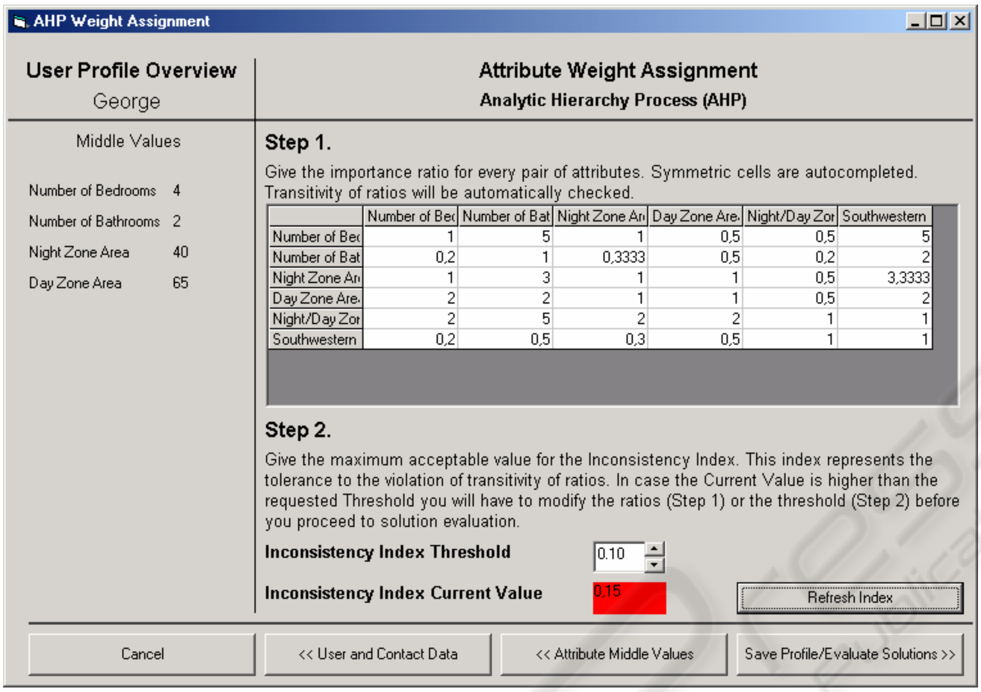

Figure 4: Decision Support Component - Weight Assignment using AHP

ATTRIBUTE. Attributes used to map solution to a smaller space that will allow further processing with respect to user profiles. The fields include name, type (int, real, scalar – big, medium, small, max, min, etc.)

NEURAL NETWORK PARAMETERS. The values of all neural network construction variables.

In addition to the aforementioned entities, a set of relationships will maintain the information about the active interconnection of the entities during user sessions. In particular,

submits. Users submit scene descriptions for processing. Each scene is connected with a certain project type.

yields. Each description, using one of the alternative solution generators that are available, results in a set of solutions complying with the description.

is mapped to. Each solution is mapped to a set of values for the attributes we have chosen to observe.

decision support profile. This relationship contains all information regarding the initial user profile as obtained by the Decision Support component. In particular, for each attribute of a specific project type, the user has already provided his/her personal interpretation of its importance for solution evaluation. This importance is represented by the corresponding weight. An additional personal parameter is that of the middle value for any attribute. The user is requested to suggest the middle value for all attributes, i.e. the actual value of the attribute that represents 50% performance of a solution with respect to the specific attribute.

machine learning profile. This is the relationship that interconnects the user, in the context of (any) specific project type, with the Machine Learning component, i.e. all values needed to fully describe the neural network used to represent the specific user's dynamic profile.

grades. Solutions are evaluated by the user through visual inspection, thus yielding the intuitive grade (*approved/rejected* or a number in case an alternative grading scheme is used). In addition, solutions are also evaluated based on the specific user's profile for the specific project type. This information may be regenerated based on the contents of the database. Nevertheless, solution evaluation is a crucial and time-consuming task; hence, once the results are available they are stored in the database.

6 SOLUTION EVALUATION

Two alternative approaches are used for solution evaluation based on user preferences forming two independent components of the user profile module. Nevertheless, their concurrent operation and results affect the overall behavior of the system, as it will become apparent in the following section.

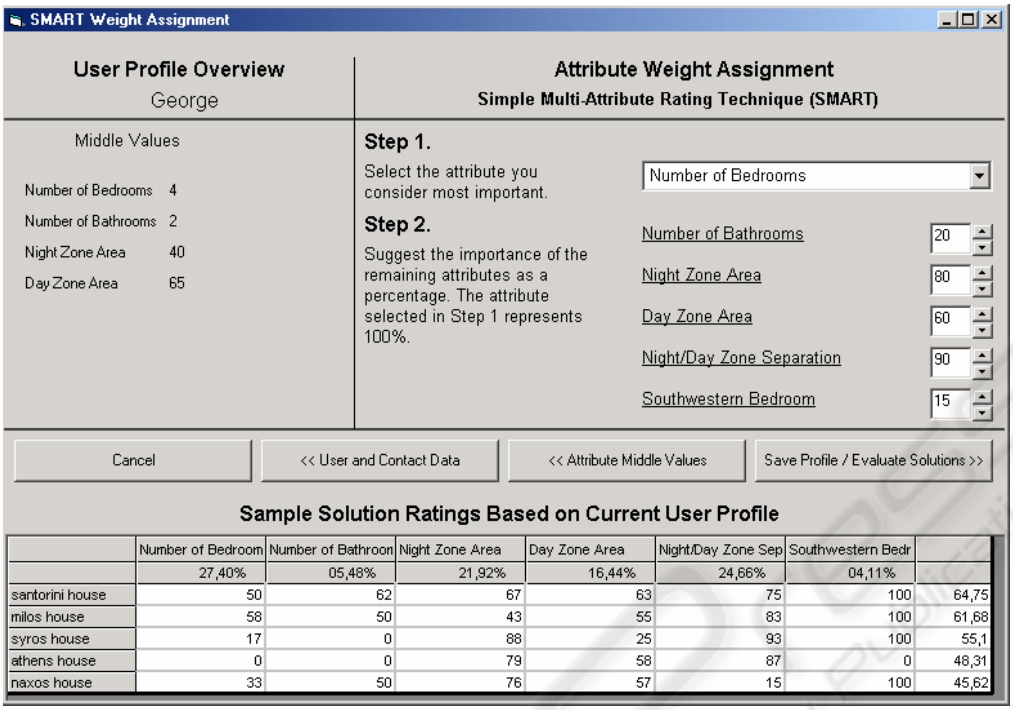

Figure 5: Decision Support Component - Weight Assignment Using SMART

6.1 Decision Support Component

Each user is requested to assign weights representing the importance of each one of the observed attributes. Two alternative methods for weight assignment are available to the user based on the corresponding stages of SMART (Goodwin 98) and AHP (Saaty 90) multicriteria decision algorithms respectively. An interesting discussion regarding dynamic weight assignment versus fixed weight values is presented in (Roberts 02).

The weights are then used to produce a score for each generated solution using a fitness function. Alternative functions may be used in the future but we currently use the inner product of the user weights with the attribute values for each solution (Bardis 04). These scores are then used in order to sort the set of solutions as a list of descending user preference order: from the most to the least preferable solution. This Decision Support (DS) component has already been implemented and currently operates on solutions in the form of attribute vectors. An example execution of the DS component for user profile initialization is shown in Figures 4-5. Notice the inconsistency index in the AHP method, signifying discrepancies in the user's answers, as well as the normalized weights appearing at the top row of the sample solution set evaluation of the SMART method. The user may rely on this component in order to obtain a set of automatically selected solutions based on the aforementioned weights. In this case solutions will be automatically visualized and presented to the user in descending preference order. However, the user may choose to manually select the preferred solutions thus contributing to the training of the neural network described in the next section.

6.2 Machine Learning Component

The ML component will be based on a neural network of six inputs – one for each observed attribute, at least one hidden layer and a single output, representing the approval/rejection of a solution by the user. Alternative structures of the network will be implemented and tested according to the criteria presented in the next section. This process may lead to the selection of more than one structures for systematic use as alternative ML components, similar to the use of two alternative methods for attribute weight assignment in the DS component.

The user, having submitted a scene description, will evaluate the solutions that are generated and visualized for the specific scene. This set of approved/rejected solutions will serve as one of the training sets for the network(s). In particular, each example in a training set represents a correct input-output mapping. In the present context, the input part, for any given solution, is comprised by the attribute values representing the specific solution. The output part simply contains the user's approval or rejection for the specific solution.

It is important to notice that the aforementioned process will be a completely transparent system task: the user will not have to submit any additional information with respect to the network(s) training and therefore, does not have to be aware of it. Automatic solution evaluation will be available from the very first session via the Decision Support component. In general, automatic solution evaluation will be at the user's discretion and the choice of the most appropriate component for this purpose will be based on the criteria presented in the next section.

7 PERFORMANCE COMPARISON

Either during the testing period or during the regular use of the system, at least two alternative methods will have to be compared with respect to their performance. In particular, solutions approved by each method will be compared to the solutions approved by the user.

7.1 Performance Indices

In order to be able to compare these methods we must provide the means to measure their performance. We concentrate on the solution selection stage that takes place after solution generation. Therefore, in the following, we will assume that solution generation has already taken place and the methods have been applied to the results. The application of each method to the solutions yields the corresponding subset of approved solutions. For simplicity, we mention only two methods in the following whereas, in practice, more alternatives – due to alternative weight assignment, alternative network adjustments, etc. – may be concurrently evaluated.

In particular, let us define the following sets:

G = The solutions generated based on the specific description of a scene.

U = The *preferred* solutions, i.e. solutions in G that comply with the user preference. These solutions represent the user preference in the current context. Formally, $U \subseteq G$. In the following we may also refer to the members of U as *approved* solutions.

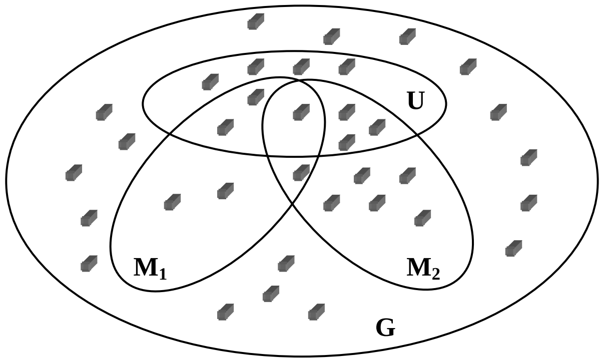

Figure 6: Example Methods Performance

G-U = The *discarded* solutions, i.e. the generated solutions that are not preferred by the user.

M_1 = solutions in G approved by Method 1. Formally, $M_1 \subseteq G$.

M_2 = solutions in G approved by Method 2. Formally, $M_2 \subseteq G$.

|S| = the number of members of any set S.

Therefore we may now define the hit rate of each method as:

$$HR_i = \frac{|M_i \cap U|}{|U|}, \quad i \in \{1,2\},$$

i.e. the percentage of approved solutions captured by the specific method.

The ratio of approved vs. total solutions selected by each method could also be used as measurement of their performance:

$$PR_i = \frac{|M_i \cap U|}{|M_i|}, \quad i \in \{1,2\}$$

There are more than one ways to define a miss rate. We may define it as the percentage of discarded solutions that are selected by the method, expressed as:

$$MR_i = \frac{|M_i - U|}{|G - U|}, \quad i \in \{1,2\}$$

However, we expect that, only a small number of the generated solutions will fulfill the user preference. This is mainly due to time limitations posed by the requirement for human visual inspection. On the other hand, |G| greatly depends on the description and can vary significantly. Therefore, we need to relate the size of the error for each method with |U| instead of a quantity including |G|. Hence, we could alternatively define:

$$MMR_i = \frac{|M_i - U|}{|U|}, \quad i \in \{1,2\}$$

and interpret a lower value as a better performance. Intuitively, this interpretation implies that a method should not select many discarded solutions when only a few preferred solutions exist. For example, when this rate is more than 1 the method gives more discarded solutions than the total number of approved solutions. Instead of 1, another value may be selected to reflect a specific performance threshold.

The above are clarified in the example Venn diagram of Figure 6, representing a general case (i.e. no intersection is empty, no two sets are equal). For the sake of simplicity of the picture, the total number of solutions is rather small, i.e. |G| is only 35 whereas this is generally not the case. Nevertheless, for the specific example, we have the following numbers:

$|G| = 35$, $|U| = 10$, $|M_1| = 6$, $|M_2| = 10$, $|M_1 \cap U| = 3$, $|M_2 \cap U| = 4$, $|M_1 - U| = 3$, $|M_2 - U| = 6$

Therefore for M_1 we have:

$HR_1 = 3/10$, $MR_1 = 3/25$, $MMR_1 = 3/10$, $PR_1 = 3/6$

and for M_2 we have:

$HR_2 = 4/10$, $MR_2 = 6/25$, $MMR_2 = 6/10$, $PR_2 = 4/10$

Table 1: Example Methods Performance Indices

Method	Hit Rate	Performance Ratio	Miss Rate	Modified Miss Rate
Method 1	3/10 = 30%	3/6 = 50%	3/25 = 12%	3/10 = 0.3
Method 2	4/10 = 40%	4/10 = 40%	6/25 = 24%	6/10 = 0.6
Extreme Case 1	1/10 = 10%	1/1 = 100%	0/25 = 0%	0/10 = 0.0
Extreme Case 2	10/10 = 100%	10/35 = 28.6%	25/25 = 100%	25/10 = 2.5

M_2 could be considered a worse (because of the higher miss rate) or a better (because of the higher hit rate) method than M_1 depending on the interpretation of these numbers. Ideally, hit rate should be equal to 1, miss rate equal to 0 and performance ratio equal to 100%. Notice, however, that a method selecting only one preferred solution every time it is invoked (Extreme Case 1) would yield a performance ratio of 100% without necessarily representing an optimal method as shown by the low hit rate. On the other hand, simply selecting all produced solutions (Extreme Case 2) maximizes the hit rate but yields a low performance ratio. Extreme Case 1 appears to represent a method with acceptable performance whereas Extreme Case 2 represents a trivial approach of no practical use. Therefore, a high Performance Ratio appears to be a necessary, although not sufficient, indication of an efficient method and it becomes apparent that we need a combination of these indices in order to accurately evaluate the performance of each method.

7.2 Method Integration

The training process of the neural network will have to continue, with alternative scene descriptions, until the ML component is considered ready to support automatic solution evaluation. We may define this threshold of ML component based on the values of the performance indices we described above for the two alternative DS components as well as for the ML component itself. In the following we will assume that the user has initialized his/her profile giving answers to the DS components that reasonably represent his/her preferences.

In particular, we will be able to rely on the results given by the network, and therefore adopt fully automated solution selection, as soon as the ML component performs consistently better than both of the DS alternatives. We can state that the ML component is *mature*, and therefore ready to take over automatic solution selection iff:

$$PR_{ML} > PR_{DS1} \quad \wedge \quad PR_{ML} > PR_{DS2} \quad \wedge$$
$$HR_{ML} > HR_{DS1} \quad \wedge \quad HR_{ML} > HR_{DS2} \quad \wedge$$
$$MMR_{ML} < MMR_{DS1} \quad \wedge \quad MMR_{ML} < MMR_{DS2}$$

for an (adjustable) number of recent descriptions submitted by the user.

The strictness of this set of conditions may be relaxed by omitting some of the inequalities. In any case, it is important to observe that, if the complete set of conditions is repeatedly true, this implies that the ML component will be capturing preferences better than the weight vectors submitted by the user himself/herself. In such a case, it will also be interesting to explore the possibility of capturing additional criteria that the user is not fully aware of, i.e. *sub-conscious* criteria. This could become apparent through the examination of experimental results and users' comments regarding system performance with respect to their preferences.

8 FUTURE WORK

This stage of our work will conclude with the detailed design and implementation the ML component. Subsequent performance comparison of the two components will lead to further refinement of their properties. Extending the solution evaluation to *grade assignment* instead of plain *approval/rejection* will also be considered. In that case performance indices will have to be modified to also reflect the *quality* of the selected set of solutions.

The next major stage of our work will focus on the enhancement of the user profile model with information originating from the connection between the declarative description and the corresponding approved solutions. Such an association will offer insight regarding the specific user's interpretation of declarative properties and relations. Successful modeling of user preferences at the declarative as well as the geometric level will allow incorporation of user profile information to the process of solution generation, thus significantly improving system performance.

ACKNOWLEDGMENTS

This study was co-funded by 75% from the European Union and 25% from the Greek Government under the framework of the Education and Initial Vocational Training Program – 'Archimedes'.

REFERENCES

Bardis G., Miaoulis G., Plemenos D., 2004. An Intelligent User Profile Module for Solution Selection Support in the Context of the MultiCAD Project. 7e Infographie Interactive et Intelligence Artificielle (3IA), Limoges, France.

Bonnefoi P.-F., Plemenos D., 2002. Constraint satisfaction techniques for declarative scene modeling by hierarchical decomposition, 3IA, Limoges, France

Brusilovsky P., 2001. Adaptive Hypermedia, User Modeling And User-Adapted Interaction 11: 87-110.

Chen C.C., Chen M.C., 2002. PVA: A Self-Adaptive Personal View Agent, Journal Of Intelligent Information Systems, 18:2/3, 173–194.

Essert-Villard C., Schreck P., Dufourd J.-F. 2000. Sketch-Based Pruning of a Solution Space within a Formal Geometric Constraint Solver, Artificial Intelligence 124, 139-159.

Fribault P., 2003. Modelisation Declarative d'Espaces Habitable, (in French), Doctoral dissertation, University of Limoges, France.

Golfinopoulos V., Dragonas J., Miaoulis G., Plemenos D., 2004. Declarative Design in Collaborative Environment, 7e 3IA, Limoges, France.

Goodwin P., Wright G., 1998. Decision Analysis for Management Judgement, Second Edition, Wiley.

Joan-Arinyo R., Luzon M.V., Soto A., 2003. Genetic Algorithms for Root Multiselection in Constructive Geometric Constraint Solving. Computers and Graphics 27, 51-60.

Makris D., Ravani I., Miaoulis G., Skourlas C., Fribault P., Plemenos D., 2003. Towards a domain-specific knowledge intelligent information system for Computer-Aided Architectural Design, 3IA conference, Limoges, France.

Miaoulis G., 2002. Contribution à l'étude des Systèmes d'Information Multimédia et Intelligent dédiés à la Conception Déclarative Assistée par l'Ordinateur – Le projet MultiCAD (in French), Professorial dissertation, University of Limoges, France.

Miaoulis G., Plemenos D., Skourlas C., 2000. MultiCAD Database: Toward a unified data and knowledge representation for database scene modeling, 3IA, Limoges, France.

Nassar K., Thalet W., Beliveau Y., 2003. A Procedure for Multicriteria Selection of Building Assemblies, Automation in Construction 12, 543-560.

Pazzani M., Billsus D., 1997. Learning And Revising User Profiles: The Identification Of Interesting Web Sites, Machine Learning 27, 313–331.

Plemenos D., 1995. Declarative modeling by hierarchical decomposition. The actual state of the MultiFormes project, Communication in International Conference GraphiCon'95, St Petersburg, Russia.

Plemenos D., Miaoulis G., Vassilas N., 2002. Machine learning for a General Purpose Declarative Scene Modeler. International Conference GraphiCon'2002, Nizhny Novgorod, Russia.

Plemenos D., Tamine K., 1997. Increasing the efficiency of declarative modeling. Constraint evaluation for the hierarchical decomposition approach. International Conference WSCG'97, Plzen, Czech Republic.

Ravani I., Makris D., Miaoulis G., Constantinides P., Petridis A., Plemenos D., 2003. Implementation of Architecture-oriented Knowledge Framework in MultiCAD Declarative Scene Modeling System, 1st Balcan Conference in Informatics, Thessaloniki, Greece.

Ravani J., Makris D., Miaoulis G., Plemenos D., 2004. Concept-Based Declarative Description Subsystem for Computer Aided Declarative Design (CADD). 7e 3IA, Limoges, France.

Roberts R., Goodwin P., 2002. Weight Approximations in Multi-attribute Decision Models, Journal of Multicriteria Decision Analysis 11: 291-303.

Saaty, T.L., 1990. The Analytic Hierarchy Process, RWS Publications, Pittsburgh, USA.

Soltysiak S.J., Crabtree I.B., 1998. Automatic learning of user profiles — towards the personalization of agent services, BT Technology Journal Vol. 16, No 3.

Vassilas N., Miaoulis G., Chronopoulos D., Konstantinidis E., Ravani I., Makris D., Plemenos D, 2003. MultiCAD-GA: A System for the Design of 3D Forms Based on Genetic Algorithms and Human Evaluation, SETN 203-214, Thessaloniki, Greece.

Vincke P., 1992. Multicriteria Decision-aid, Wiley (1992).