THE MINERVA SYSTEM: A STEP TOWARD AUTOMATICALLY CREATED VIRTUAL MUSEUMS
Francesco Amigoni *, Viola Schiaffonati *
* Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy

Online Publication Date: 01 March 2009

To cite this Article Amigoni, Francesco and Schiaffonati, Viola(2009) THE MINERVA SYSTEM: A STEP TOWARD AUTOMATICALLY CREATED VIRTUAL MUSEUMS, Applied Artificial Intelligence, 23:3, 204 — 232
To link to this Article DOI: 10.1080/08839510802715159
URL: http://dx.doi.org/10.1080/08839510802715159

PLEASE SCROLL DOWN FOR ARTICLE
THE MINERVA SYSTEM: A STEP TOWARD AUTOMATICALLY CREATED VIRTUAL MUSEUMS

Francesco Amigoni and Viola Schiaffonati
Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy

The application of artificial intelligence (AI) tools to cultural heritage yields new technological solutions for the fruition of museums and art exhibitions. In this article we present a version of the Minerva system, able to support the organization of virtual museums by cooperating with curators to set a collection of works of art in an environment. This new version organizes a part of the archeological finds belonging to the collection of the Archeological Museum of Milan. The role of Minerva is not to substitute the curators but to assist them in setting up a museum or an art exhibition. In this perspective, Minerva carries out automatically some tasks involved in museum organization and constitutes a unique system among those oriented to cultural heritage. The experimental activities have been conducted in cooperation with archeological experts who validated the results produced by Minerva.

INTRODUCTION

The study of computational tools for enhancing the realization of cultural contents is a topic of increasing importance in current research (Edmonds and Candy 2002). In particular, the application of artificial intelligence (AI) tools to cultural heritage is acquiring a central role, promoting new technological solutions for museums and art exhibitions. In general, these tools are devoted to enhance the way the works of art are experienced or to improve their management. During the last years, we have developed a software system, called Minerva, that, instead, is applied to the organization of works of art. Minerva is able to automatically organize virtual museums, starting from the collections of works of art and the environments in which they must be displayed (Amigoni and Schiaffonati 2003; Amigoni et al. 2001). By organization, we mean two distinct
processes: the preparation process, which is the arrangement of works of art in conceptually relevant ordered groups, and the allocation process, which is the placement of these groups within a given geometrical space trying to preserve their arrangement.

In this article we present a version of Minerva which is oriented to organize some archeological finds belonging to the collection of the Archeological Museum of Milan. This version significantly improves those presented in Amigoni and Schiaffonati (2003) and Amigoni et al. (2001). From a functional point of view, the main innovation is the enhanced interactivity with the user, usually a museum curator, both in the preparation and in the allocation processes. From a technical point of view, the main innovation is a completely new client-server implementation that is now based on the JADE (Telecom Italia Lab 2006) multi-agent platform. The improved interaction with the user is particularly significant and has a strong impact on the architecture of the system. The final organized museum is the expression of the inferential processes performed by the system supported by a dialogic interaction with the user. This contrasts with the previous versions of Minerva, in which the emphasis was on the attempt to completely automate the tasks related to museum organization. As a positive feedback of this new approach, the results obtained with the new version of Minerva have been better judged by the archeological experts we cooperate with. Generally speaking, Minerva users are people who can create and visit their own virtual museums; for example, people visiting a physical museum can enhance their experience by creating a virtual museum alternative to the physical one. However, in this article, we refer to museum curator users, namely, to expert users who can fully exploit the potential of Minerva. Minerva constitutes an example of a combination of traditional and recent AI techniques that are successfully applied to the organization of collections. The implementation of Minerva blends together the classical inferential techniques of AI (Russell and Norvig 2003) and the more recent paradigm and technology of multi-agent systems (Weiss 1999). To the best of our knowledge, Minerva represents a unique system in the world of computational tools for museums. We are not aware of any other system with similar purposes, features, and degrees of autonomy.

The main purpose of Minerva is to support the organization of museums, given the collections and the environments in which they must be displayed. Organizing a museum is usually thought of as a typical human-exclusive activity based on cultural knowledge and curatorial judgements: the final result is a global coherent setting that allows an effective visit of the museum. However, every organization follows—at least partially—explicit criteria and some of them can be considered for an automatic organization. These include physical criteria: some works of art, for example, could not be inserted in some rooms because they are scarcely lit or too small. In addition,
also some artistic criteria are involved in museum organization: they can play a central role in the final result, since they are concerned, for example, in how to give emphasis to particularly important works of art or how to build connections among different works of art.

The improved interaction with the user makes Minerva particularly suitable in providing museum curators (or whoever is involved in museum organization) with the possibility to quickly prototype and compare different virtual organizations based on different criteria. This is expected to reduce the time and costs of museum organizations. Metaphorically, the role of Minerva can be assimilated to that of a truth maintenance system by which it is possible to reason on different “what-if” possibilities (Russell and Norvig 2003). In particular, Minerva is able to manage some physical aspects of the allocation process that a curator always must consider, but that can be usefully delegated to an artificial system. These aspects include normative, space, and budget constraints about the placement of the works of art in the environments. The obvious advantage is that the curator can pay more attention to the artistic aspects involved in the preparation process, such as the ways in which the works of art can be enjoyed at the best, and avoid the tedious task of adjusting the positions of the works of art to respect the legislation. Roughly, we might say that, in order to produce a museum organization, a curator drives the preparation process while the system performs the allocation process. Let us suppose that a curator decides to set an exhibition of design objects with the aim of narrating the evolution of the Italian design from the end of WWII to the end of the twentieth century. (A previous version of Minerva was applied to the organization of design museums (Amigoni and Schiaffonati 2003).) The exhibition can be organized, for example, on the basis of the level of innovation of the material and the shape of the objects, or alternatively, on the basis of the innovation in their manufacturing processes. If the curator could complete a preliminary evaluation on the two virtual prototypes of the exhibition produced by Minerva, the choice between the two alternatives could be done in a more thoughtful way. Given that the final decision is made by the curator, Minerva offers the opportunity to combine different criteria to produce several potential exhibitions of the works of art.

The current tendencies in organizing museums and exhibitions are based on the idea of telling a story, a sort of tale that is narrated to the visitors by the works and objects exhibited and by the connections emerging from their arrangement (Greenberg, Ferguson, and Nairne 1996; Karp and Steven 1992). The criteria by which the story is created are not necessarily chronological, but depend on the messages, ideas, emotions, and sensations the curator would like to transmit to the visitors. By producing virtual prototypes of the possible arrangements, Minerva can support this creative organization of a museum as a story.
INTERACTION BETWEEN MINERVA AND THE USER

As previously mentioned, the purpose of Minerva is not to substitute humans in organizing museums. Rather, it aims at supporting them in their work by enhancing their creativity, while providing a sophisticated computational tool for automatizing some tasks related to museum curatorial work. The museum produced at the end of the organization process is the result of both the “intelligent” abilities of the system and the decisions of the human user. Indeed, the user interacts with the program and intervenes when the system cannot manage some parts of the process. This enhanced interaction is one of the distinctive features of the version of Minerva described in this article. Although in the first version of the system the emphasis was on the desire for its complete autonomy (Amigoni et al. 2001), this has not been judged positively by some museums experts (including the director of the Archeological Museum of Milan) that were asked to evaluate the system. They preferred more control in the museum organization process, since they believe that an artificial system cannot manage such complex and creative tasks automatically. For this reason, we have chosen to better balance the abilities of the system with the human intervention in the new version of Minerva.

In order to describe the possibilities offered by the system, let us consider a session with Minerva. The works of art the system manages are a subset of those belonging to the collection of the Archeological Museum of Milan. The user—typically, a museum curator—can select a set of works of art and an environment in which to display them. For example, in the case shown in Figure 1, the user has decided (see the left part of the window) to display some works of art in the rooms of the Caserma Napoleonica (Napoleonic Barracks), an historical building in the center of Milan. (The interface is in Italian, since at this stage, Minerva is oriented towards Italian users.) Other buildings, among those stored in a database of environments, can be chosen. Currently, these environments are indoor and structured in rooms. In the central part of the window of Figure 1 the user can choose which works of art are going to be displayed. For example, the user can select the celebrative archeological findings together with funerary and sacred findings. The user cannot choose the single works of art he or she intends to display in the virtual museum, but only the sets of works of art, according to the selected criteria. The works of art the user can select are extracted from a database. This database can contain objects both from permanent collections and from temporary exhibitions. Note that at this stage of the project, we do not provide any fancy user interface, since we have focused on the development of the kernel of Minerva.

After the selection of the environment and the set of works of art, the user can apply the criteria of the preparation process. They drive the
grouping of the selected works of art and determine the order in which the groups are displayed in the selected environment. For an archeological collection the chronological criterion is, of course, the most important one. Besides that, five other criteria can be selected by the user in this version of Minerva: they have been chosen on the basis of what has been recognized by archeological experts to be relevant in organizing archeological museums. They are: the place of origin of the works of art, their material, their culture, their function, and the kind of objects (such as statues, steles, and busts). According to the criteria selected by the user,
the works of art are classified in homogeneous ordered groups by the system. Groups can be further divided into subgroups. During the preparation process, the user has the possibility to change both the groups (e.g., by splitting a group in subgroups according to a given criterion) and their order. In Figure 2, the user has grouped the works of art in four ordered groups according to their kind: the funerary steles, the balsamariums, the drinking vessels, and the plates. The user can further refine the groups, for example, by splitting the group of funerary steles into two subgroups: the Roman ones and the Etruscan ones. The possibility for the user to act on the groups of works of art and their order is one of the main innovations of the new version of Minerva. In the previous
versions, this preparation process was performed automatically by the system starting from the criteria selected by the user. The results produced in that way were not considered satisfactory by the archeological experts we asked for testing the system. This was mainly due to the fact that it is almost impossible to find general rules to organize archeological works of art. The selection and the application of preparation criteria generally depend on the available works of art and on the sensibility of the curator. For this reason, in the new version of Minerva, we let the user decide how to apply these criteria.

When the user is satisfied with the preparation, namely, with the groups of works of art and their order, the allocation process takes place. The allocation process, which is performed automatically by Minerva, places the works of art in some showcases, and then places the showcases in the rooms of the environment selected by the user. The user can set some parameters of the allocation process, which control the placement of the works of art in the environment (Figure 3). These parameters include

![Image of Minerva allocation process](image-url)

**FIGURE 3** Some parameters of the allocation process.
the maximum and minimum occupation of the area and the perimeter of
the rooms, the maximum number of works of art that can be placed in a
showcase, the fraction of the space that must be left free in a showcase,
and the order (clockwise or anti-clockwise) in which the works of art are
placed in the rooms. During the allocation process, Minerva tries to
allocate each group of works of art in a room of the environment. The groups
are considered in the order set during the preparation, while the rooms are
considered in the order in which they are accessible. When a room is too
small to contain a group, some works of art of the group are allocated in
the next room. When a room is too large and a group of works of art does
not fill it completely, the user is asked if Minerva has to allocate the
next group in the same room or if the room has to be left underoccupied
(Figure 4). The user takes a decision according to the aesthetic effect he
or she would like to obtain. Note that this choice can be hardly made auto-
matically by the system, since it depends on several factors, like the kind and
number of the works of art in the groups. This interaction with the user dur-
ing the allocation process is another of the innovations of the new version of
Minerva presented in this article. In the previous versions, all these decisions
were taken automatically by the system, leading to museum organizations
that were considered too “schematic.”

When the preparation and allocation processes have been performed,
the user can visit the organized museum in a VRML virtual reality setting, as
shown in Figure 5. Each work of art is virtually reproduced according to its
3D model obtained from the real object or from its representations (e.g., pic-
tures). In the virtual museum, the user can navigate freely or follow a list of
viewpoints. The user can also access the detailed information (including a
more detailed VRML model and a textual description) associated to the works
of art by clicking on the works of art themselves, as shown in Figure 6.

IMPLEMENTATION OF MINERVA

Minerva is implemented as a multi-agent system composed of auto-
nomous software agents that perform different tasks (Weiss 1999). Some
agents employ inferential techniques in order to carry out the organization
of the works of art. The museum organizations produced by the system are
the result of the cooperation of the agents of Minerva.

The architecture of the new version of Minerva presented in this arti-
cle differs significantly from those presented in Amigoni and Schiaffonati
(2003) and Amigoni et al. (2001). The main conceptual innovation is
the more intense interaction with the user. The main technological inno-
vations are the client-server architecture and the use of the JAVA Agent
Development (JADE) framework JADE (Telecom Italia Lab 2006) platform
to implement the system. More precisely, we coded the agents in JAVA
FIGURE 4 A window that asks the user if the Roman busts have to be allocated in the same room with the Roman statues.
FIGURE 5 Some snapshots of the VRML virtual visit of an archeological museum organized with Minerva.

FIGURE 6 The detailed information about a stele.
within the JADE platform and exploited the Jess framework (Friedman-Hill 2006) to perform inferential activities. JADE is a software framework fully implemented in JAVA language that offers a middleware to simplify the implementation of multi-agent systems. JADE is compliant with FIPA specifications (Foundation for Intelligent Physical Agents 2006b), which are a widely used collection of standards for multi-agent systems. Agents in JADE communicate by exchanging messages in a language called agent communication language (ACL) (Foundation for Intelligent Physical Agents 2006a). The notation $m(a, b, C)$ indicates a message $m$ with content $C$ sent by agent $a$ to agent $b$. To allow an open and effective communication between the agents of Minerva, we developed an ad hoc ontology for the system. The JADE platform includes some middle agents that provide some useful services. The agent management system manages the platform, supervises access, and provides naming and white pages services. It automatically associates a unique identifier to each agent registered on the platform. Moreover, the directory facilitator provides yellow pages services, by maintaining a list of abilities of the agents registered on the platform. The yellow pages services allow the agents to find themselves on the basis of the functions they perform without knowing their implementation details. For example, for querying the databases (DB) of Minerva, an agent asks the directory facilitator for the identifier of the agent that currently provides the DB management services and then, obtained that identifier, sends a query message to the appropriate agent.

The agent-based implementation of Minerva provides several advantages. For instance, the modularity of the system is improved (as it is typical for agent-based systems (Jennings 2001)). We can change the details of the internal functioning of an agent (e.g., of the agent providing DB management services, switching from MICROSOFT ACCESS to MYSQL technology) without affecting the other agents. Moreover, the scalability of the system is empowered by having more agents with the same functions, possibly running on different computers.

Figure 7 presents the overall architecture of Minerva. The agents can run on different computers. In the following sections, the main agents of Minerva are described.

**The Client Agent**

The client agent is the user interface of Minerva. A client agent is created at the start of the client side of Minerva and is associated to a user (more precisely, to a user’s session). The interface provided by the client agent is JAVA-based and is very simple, because so far we have focused on the development of the core components of the Minerva. The client
A comprehensive discussion on the interaction with a client agent of Minerva has been presented in the previous section. The main phases of the interaction with the user are: the selection of the works of art and the environment where they will be displayed; the creation of the groups of works of art and their ordering; the setting of the allocation parameters and the answering to the system questions during the allocation process; and the visit of the VRML representation of the organized virtual museum. The groups of works of art and their order, as selected by the user during the preparation process, constitute a taxonomic tree (like those represented on the left-hand-side of Figure 2 and in Figure 8). The taxonomic tree is built by the user with the assistance of the client agent. This tree contains the criteria by which the groups of works of art have been formed and the order in which the groups should be displayed in the virtual museum. The leaves of the taxonomic tree represent the works of art that are conceptually homogeneous according to the criteria represented by their parent nodes. The order of the nodes in the tree represents the order of the corresponding groups according to the user’s selections. For example, in the case of Figure 8, the user has decided to classify the objects according to their culture (Roman, Etruscan, and unknown) and to further classify the Roman works of art according to their kind (busts, statues, and funerary steles). The single works of art are displayed by their codes. The order of the nodes is, as said, significant.

We outline that the data the client agent presents to the user and that populates the taxonomic tree are dynamically retrieved from the databases.
of Minerva on the server side. For example, in Figure 8, when the user decides to classify the works of art according to the culture criterion, the values Roman, Etruscan, and unknown for the culture criterion are retrieved by the client agent from the server side of Minerva. Similarly, in Figure 8, the works of art shown under the different branches (Roman busts, Roman statues, Roman funerary steles, Etruscan, unknown) are retrieved by the client agent by querying the databases on the server side of Minerva.

Finally, we also implemented the possibility of selecting for the virtual visit one of the allocations previously performed by the user (and stored in an allocation database). This allows the user to quickly compare different allocations, without repeating the corresponding organization processes.

**The Minerva Server Agent**

The Minerva server agent is the “gateway” of the server side of Minerva and is the only agent a client agent views. It manages the message flow between the client agents and the other agents on the server side of Minerva. More precisely, the Minerva server agent manages a number of
objects called sessions, each one associated with a client agent. The session is used to keep the state of the conversation between a client agent and Minerva. When the Minerva server agent receives a message $m(a, b, C)$ coming from a client agent $a$ and sent to an agent $b$ on the server side, it adds to the content $C$ of the message some information about the session of $a$ and forwards $m$ to the supervisor agent (described later). Conversely, when the Minerva server agent receives a message $m(a, b, C)$ sent from an agent $a$ on the server side to a client agent $b$, it eliminates from $C$ the information about the session and forwards $m$ to $b$. Note that the Minerva server agent does not care about the actual content and intent of the message—this is done by the supervisor agent.

The Supervisor Agent

The supervisor agent receives (via the Minerva server agent) the messages sent by the client agents and decides what to do on the basis of the content and the performative of the messages. Broadly speaking, the performatives define the intent of the messages, like request, inform, query, . . . (Foundation for Intelligent Physical Agents 2006a). Introducing a supervisor agent enhances the modularity of Minerva. The other agents on the server side of Minerva offer services that are composed by the supervisor agent in order to support users, interacting with client agents, building and navigating virtual museums. For example, the DB manager agent offers the services of managing the databases of Minerva, but it does not know what data should be stored or retrieved to perform a given operation, like the allocation of the works of art in the rooms. (Some agents cited in this section will be described later.) The services of the DB manager agent are invoked by the supervisor agent that maintains a global view of the process of museum organization. The supervisor agent, for example, sends to the DB manager agent the allocations performed by the allocator agents. This way, the services offered by the agents are uncoupled from the way these services are composed, enhancing the modularity of the system.

More precisely, for each client agent $c$, the supervisor agent performs the following main operations (in the given order). All of these operations are triggered by $c$ or by the user interacting with $c$:

- During the preparation of the museum, the supervisor agent sends to $c$ the following data that are retrieved from the DB manager agent:
  - the preparation criteria and their values;
  - the works of art that match with the selected preparation criteria.
- During the allocation of the selected works of art in the rooms of a museum, the supervisor agent initially creates an allocator agent $a_c$.
associated to \( c \) (and thus to the user interacting with \( c \)). Then the supervisor agent can receive from \( a_c \) the request to ask the user whether two different groups of works of art should be allocated in the same room or in different rooms. The supervisor agent sends this request back to the user's answer. The answer is then sent to \( a_c \). Finally, at the end of the allocation process, the supervisor agent receives the resulting allocation from \( a_c \) and sends it to the DB manager agent that stores the allocation in the allocations database.

- During the building of the VRML model of the museum, the supervisor agent initially creates a VRML builder agent \( v_c \) associated to \( c \). Then, the supervisor agent sends to \( v_c \) the VRML models of the works of art and the rooms of the museum. These models are requested by \( v_c \) and are retrieved from the DB manager agent.

Moreover, the supervisor agent regularly (every 10 seconds in our implementation) checks the presence of the agents that have been registered to the directory facilitator of JADE to offer a service. Firstly, it gets the list of agents currently present in the system, exploiting the agent management system of JADE. Secondly, it checks if the agents that are registered in the yellow pages to offer some service are actually running. This way, the supervisor agent can detect if an agent has gone down and, in this case, can recreate the agent. A similar procedure is applied by the Minerva server agent to recreate, when needed, the supervisor agent. Note that this fault recovery procedure is very primitive: for example, the Minerva server agent cannot be recreated and no attempt is made to recover the data the agents were working on. Despite its simplicity, our fault recovery procedure has proven adequate for the current stage of development of Minerva.

The Preparator Agent

The preparator agent receives from the client agent associated to a user the taxonomic tree produced by the user. The structure of the taxonomic tree is far from regular: for example, its depth and the number of nodes for each level strongly depend on the preparation criteria selected by the user. For this reason, we need a more efficient representation of the taxonomic tree to work on.

The preparator agent translates the taxonomic tree in a knowledge base that allows for an easier management of the tree. More precisely, the preparator agent uses the rules and facts of the jess production system to represent the groups of works of art, each one composed of homogeneous objects. The facts represent the works of art and the rules group together
the works of art. For example, the following rule pairs two works of art that are similar given a set of preparation criteria:

(defrule Pair-Objects
  (selected-criteria ?head $?tail)
  (object (name ?name1))
  (object (name ?name2))
  (and (neq ?name1 ?name2)
       (eq (extract-feature ?name1 ?head)
            (extract-feature ?name2 ?head)))
  (not (exists (pair ?name2 ?name1)))
  =>
  (assert (pair ?name1 ?name2)))

The representation of the taxonomic tree as a knowledge base in JESS is more efficient to perform the allocation process, which is implemented as an inferential process in JESS.

The Allocator Agent

An allocator agent is created by the supervisor agent when it receives, from a client agent, the request to start the allocation process. Each allocator agent \( a_c \) is univocally associated to a client agent \( c \). An allocator agent, starting from the JESS-based representation of the taxonomic tree, automatically finds a geometrical allocation of the works of art in the rooms of the museum. The allocation of the works of art is subject to some constraints, which museum curators usually take into account. For example, the rooms should be considered in the order they are accessible from the main entrance of the museum. Moreover, the rooms should not contain too few nor too many works of art, the windows and the doors should not be obstructed by works of art, and so on. More precisely, the constraints we considered are the following: the maximum and minimum occupation of the area and the perimeter of the rooms, the maximum number of works of art that can be placed in a showcase, the fraction of the area that must be left free on the top surface of a showcase, the order (clockwise or anti-clockwise) in which the works of art are placed in the rooms, and the distances governing the collocation of the showcases according to the location of the doors, windows, and other showcases to allow the passage of the visitors. The values of some of these constraints (e.g., the maximum and minimum occupation of the rooms) can be set by the user.

The allocator agent exploits the knowledge inserted in its JESS knowledge base about where to place the works of art in a given environment
in order to satisfy the constraints. An example of JESS rules used in the allocation process follows:

(defrule Allocate-A-Group-in-A-Room
  (min-Perimeter-Occupation ?mip)
  (max-Perimeter-Occupation ?map)
  (min-Area-Occupation ?mia)
  (max-Area-Occupation ?maa)
  (group ?group ?area-g ?perimeter-g)
  (room ?room ?area-r ?perimeter-r)
  (and (> ?area-g (* ?area-r ?mia))
       (> ?perimeter-g (* ?perimeter-r ?map))
       (<= ?area-g (* ?area-r ?maa))
       (<= ?perimeter-g (* ?perimeter-r ?map)))
  =>
  (assert (allocation ?group ?room)))

This rule determines if a group of works of art can be allocated in a room according to the constraints on the occupation of the area and the perimeter.

The groups of works of art are allocated in the rooms in the order established by the user. The allocator agent tries to allocate each group in a different room. If it fails, the allocator agent spreads the group over more rooms or asks the user if more groups should be allocated in the same room.

Let us explain in more detail how the allocation process is performed. Initially, the allocator agent is provided with the taxonomic tree and retrieves the descriptions of the rooms of the museum from the DB manager agent. These descriptions include, among others, information about the dimensions of the rooms, the location of doors and windows, the nature of the room: passage or expositive, and in this last case, the position of the room in the ordered sequence of rooms accessible from the main entrance of the museum. In summary, the allocation process starts from an ordered sequence of groups of works of art \(G_1, G_2, \ldots, G_n\) (extracted from the taxonomic tree) and from an ordered sequence of rooms \(r_1, r_2, \ldots, r_m\).

For each room \(r_i\) of the museum (starting from the first in the sequence), the allocator agent considers the first group of works of art \(G_j\) not yet allocated and performs the following operations:

- **Allocation in the showcases.** Many archeological works of art are small objects, like pottery fragments, which are conveniently displayed in showcases. The objects are displayed on the top surface of the showcase
(see Figure 5). In the current implementation, showcases have fixed dimensions. A showcase can display more objects. Each object in $G_j$ is associated to a showcase. The allocator agent adds the works of art in $G_j$ to a showcase until the number of objects in the showcase is larger than a parameter ($\text{MaxObjectsInShowcase}$) or until the objects fill the space available in the showcase (the free area on the top surface of the showcase must be larger than the fraction $\text{FreeAreaFraction}$ of the available area). Note that some archeological objects can be viewed from all sides (e.g., statues), while others should be viewed from one preferred side (e.g., steles). In allocating the works of art to showcases, the allocator agents considers this fact and puts in the same showcase objects that share the same visibility characteristics. The output of this step is a set of showcases $S_j = \{s_{j,1}, s_{j,2}, \ldots, s_{j,q}\}$ that have been obtained from the group $G_j$.

- **Allocation in the large.** In this step, the allocator agent determines if the set of showcases $S_j$ obtained from the previous step can be contained in the room $r_i$. The allocation in the large considers only the geometrical constraints connected to the space occupied by the showcases, to the space available in the room, and to the space that must be left free between the showcases to allow the passage of visitors. When allocating the set $S_j$ in the room $r_i$, the allocator agent can face the following three situations:

  - All the above constraints are satisfied. In this case, the allocator agent goes to the next step, the allocation in the small.

  - The set $S_j$ is too large to be allocated in the room $r_i$. In this case, the allocator agent splits the group $G_j$ of works of art (from which $S_j$ has been obtained) in two new groups $G_{j1}$ and $G_{j2}$ (that substitute $G_j$ in the group sequence). $G_{j1}$ and $G_{j2}$ are said homogeneous because they derive from the same group $G_j$. Then the process starts again from the allocation in the showcases that is performed for $G_{j1}$, obtaining a new set $S_{j1}$. Note that we split the group of works of art $G_j$ and not the set of showcases $S_j$. This opens the way to more interesting possibilities (that we have not yet explored) to automatically allocate more objects in a room, for example, by increasing the number of objects in the showcases.

  - The set $S_j$ is too small to completely fill the room $r_i$. In this case, if the group $G_{j+1}$, successor of $G_j$ in the order of the taxonomic tree, is homogeneous with $G_j$ (i.e., has been derived from splitting the same original taxonomic group, see the previous case), then the allocator agent moves a work of art from $G_{j+1}$ to $G_j$ and starts again from the allocation in the showcases with an augmented $G_j$ (and a diminished $G_{j+1}$), obtaining a new set $S_j$. Conversely, if $G_{j+1}$ is not homogeneous with $G_j$, then the allocator agent asks the user (via the corresponding
client agent) if he or she wants to leave the room under occupied or to allocate in the same room two different groups. When the user answers, the process starts again from the allocation in the showcases, if the user wants to fuse two groups in the same room, or the process continues with the allocation in the small, if the user wants the room under occupied. The output of this step is a (possibly updated) set of showcases $S_j$.

- **Allocation in the small.** This step places the showcases of $S_j$ into room $r_i$ according to the visibility constraints and to those related to doors and windows. Around each showcase adequate space must be guaranteed for allowing the visitors to view the works of art in a showcase according to their preferred side of view (see the allocation in the showcases step). Moreover, the showcases should not be placed in proximity to windows and doors. To address the first kind of constraints, during the allocation in the small, the allocator agent augments the occupation of each showcase to take into account the space that must be reserved for the visitors to view the objects in the showcase. Note that the areas reserved for the visitors of two different showcases can overlap. According to the preferred side of view of its objects, a showcase can be placed along a wall (when the objects are preferably viewed from one side) or in the center of the room (when the objects are preferably viewed from all sides). During the allocation in the small, it could happen that a set of showcases $S_p$ which were supposed to fit the room according to the allocation in the large, cannot be placed in the same room. In this case, a work of art is removed from the group $G_j$ and inserted in the next group $G_{j+1}$, if $G_{j+1}$ is homogeneous with $G_p$, or in a new group $G'_j$, if $G_{j+1}$ is not homogeneous with $G_j$ (in this case, $G_j$ is inserted in the group ordering just after $G'_j$ and before $G_{j+1}$). In any case, the process restarts from the allocation in the showcases. To avoid cycles (the allocation in the small removes an object from $G_p$, the allocation in the large adds an object to $G_j$, . . .), we force the allocation in the large to tolerate a small under occupation of the room.

The allocation process terminates when there are no more groups of works of art to allocate or when there are no more rooms available. When the allocation process has been performed, the allocator agent notifies the supervisor agent of the termination and sends the final result. Then, the allocator agent kills itself.

**The VRML Builder Agent**

A VRML builder agent is created by the supervisor agent when a client agent requests to view the virtual museum resulting from an allocation. Each VRML builder agent $v_c$ is univocally associated to a client agent $c$. The virtual
museum is represented in the VRML language. Every work of art and every environment stored in the databases of Minerva has a field that contains the VRML code that allows its 3D representation. The VRML builder agent retrieves these VRML codes from the DB manager agent and adds the code for positioning the showcases in the rooms and the works of art in the showcases, according to the allocation to be displayed (that is the input of the VRML builder agent). Moreover, the VRML builder agent creates and links to each work of art the corresponding HTML page that contains the detailed information about the object (see Figure 6). The data for the HTML page are retrieved from the DB manager agent. The final results are written in some files (a .wrl file and a number of .html files) on the server side of Minerva and the path of the .wrl file is sent to the client agent that made the initial request. Finally, the VRML builder agent kills itself.

The DB Manager Agent

The DB manager agent is the only interface to the databases of Minerva. It translates the requests of the other agents, formulated as ACL query messages in SQL queries for the databases. Currently, the databases are implemented with Microsoft Access. The main databases of Minerva follow:

- the works of art database, which contains all the information about the works of art, including the dimensions, the sides for viewing, the best distance for viewing, the VRML code, and the artistic data (like date, place of origin, culture, function, and kind);
- the environments database, which contains all the information about the museums that can virtually be reproduced and their rooms, including the position of the walls, doors, and windows, the nature (passage or exposive) and the position in the sequence of the rooms, and the VRML code;
- the allocations database, which contains the allocations performed by the users of the system.

EXPERIMENTAL RESULTS

We extensively tested the Minerva system we implemented in order to assess its robustness and the quality of its results. For example, we tested the fault recovery procedure to recreate agents discussed in a previous section. Moreover, we let some archeological experts interact with the system in order to evaluate the virtual museums produced. In almost all cases, they judged the results produced by the system “reasonable.” In particular, they appreciated the possibility of interacting with it in order to finely tune the way in which the works of art are displayed. This gives
the users the feeling of having the control over the organization process, while leaving the system space for autonomous operations.

Following, we present some examples that evidence the possibilities of Minerva. Figure 9 shows a set of 29 works of art that were selected from the Minerva databases for the experiments presented later. It includes some busts, steles, bas-reliefs, and pottery. We also selected the Ca-ser-ma Napoleonica as the environment in which to display these works of art.

In the first experiment, we built a taxonomic tree in which the works of art of Figure 9 are classified according to their kind. We chose to order the groups in the following way:

- Bas-reliefs, composed of eight works of art (A9880203, A9421, A21700, A18498, A17923, A15494, A10599, A13497, see the codes reported in Figure 9);
- Drinking vessels, composed of a single work of art (A0910859);
- Busts, composed of a single work of art (A1157);
- Plates, composed of a single work of art (A0910853);
- Funereal steles, composed of seven works of art (A096623, A096799, A096625, A096743, A096626, A096577, A096621);
- Torsos, composed of eight works of art (A1153, A1161, A1178, A1174, A1163, A1425, A4036, A4050);
- Olla vases, composed of two works of art (A0910854, ST7416BIS);
- Other vases, composed of a single work of art (ST7416).
Then, we requested Minerva to allocate the groups in the rooms of the Caserma Napoleonica and we requested (by answering the questions of the system during allocation) that different groups were allocated in different rooms. The result is schematically represented in Figure 10. Some

FIGURE 10 An allocation of works of art with different groups in different rooms.
issues are worth pointing out. In this case, only 24 out of the 29 works of art of the initial set could be allocated in the given environment: there were no rooms left for the remaining works of art (two torsos and three vases). Having allocated different groups in different rooms and given that some groups are composed of a single work of art, some rooms contain a single object, like rooms 4, 5, and 6. Note also that room 9 contains a single stele, but this is the result of splitting the group of steles over rooms 7, 8, and 9.

In the second experiment, we built a taxonomic tree in which the works of art of Figure 9 are classified according to their material. We chose to order the groups in the following way:

- Clay, composed of four works of art (A0910859, ST7416, ST7416BIS, A9421);
- Stone, composed of nine works of art (A096623, A096799, A096625, A096743, A096626, A096577, A096621, A21700, A15494);
- Glass, composed of two works of art (A0910853, A0910854);
- Schist, composed of five works of art (A9880203, A18498, A17923, A10599, A13497);
- Bronze, composed of a single work of art (A1157);
- Marble, composed of eight works of art (A1153, A1161, A1178, A1174, A1163, A1425, A4036, A4050).

Then we asked Minerva to allocate the groups in the rooms of the Caserma Napoleonica and we allowed the allocation of different groups in the same room (by answering to mix groups to all the questions of the system). The result is schematically represented in Figure 11. In this case, differently from the previous one, all the works of art of the initial set could be allocated in the given environment. The allocation of different groups in the same room produced, for example, the allocation of the bronze bust in room 7 alongside the marble torsos. It can be noted that there are some empty rooms. Rooms 2 and 6 contain a single work of art because their dimensions do not allow other objects to be placed alongside.

In the third experiment, we built a taxonomic tree in which the works of art of Figure 9 are classified, at a first level, according to their culture. Then we further chose to classify the Roman works of art according to their kind. We chose to order the groups in the following way:

- Unknown, composed of five works of art (A0910859, A0910853, A0910854, ST7416BIS, ST7416);
- Oriental, composed of eight works of art (A9880203, A9421, A21700, A18498, A17923, A15494, A10599, A13497);
- Roman, further divided in the following groups:
- Torsos, composed of eight works of art (A1153, A1161, A1178, A1174, A1163, A1425, A4036, A4050);
- Funerary steles, composed of seven works of art (A096623, A096799, A096625, A096743, A096626, A096577, A096621);
- Busts, composed of a single work of art (A1157).
Then we asked Minerva to allocate the groups in the rooms of the Caserma Napoleonica and we allowed for allocating in the same room only the group of the Oriental objects and the group of the Roman torsos. The result is schematically represented in Figure 12. In this case, the system
allocated all of the works of art occupying all the available rooms. Room 5 contains an Oriental work of art and a Roman torso.

The time required by Minerva to perform the above allocations strongly depends on whether we allow the allocation of different groups in the same room. For example, in the first and third experiment, the system was able to find the allocations in about 100 seconds. The allocation of the second experiment (where different groups were allocated in the same rooms) required about 450 seconds. This can be explained in the following way: when trying to allocate more groups in the same rooms, the system is solving a complex problem because it tries to fill each room to the maximum capacity, while satisfying the constraints on the occupation of the room. Several iterations of the allocation steps may be needed to find an optimal allocation. This, of course, requires more time than a suboptimal occupation of the rooms. Note that, at this stage of the project, we have not made any attempt to optimize the performance of the system.

Finally, in Figure 13 we present some experiments that illustrate the allocation of more works of art in the same showcase. In Figure 13(a), all of the works of art can be allocated correctly. In Figure 13(b), not all of the five works of art can be allocated because of the large value of the parameter FreeAreaFraction. In Figure 13(c), all of the five works of art can be allocated because the value of the parameter FreeAreaFraction is smaller. In Figure 13(d), some free area is left on the top surface of the showcase because of the small value of the parameter MaxObjectsInShowcase. Acting on the values of the parameters MaxObjectsInShowcase and FreeAreaFraction, it is possible to modify the number of works of art in a showcase. Consequently, the number of showcases in a room can be modified and, accordingly, the allocation produced by the system.
RELATED WORKS

To the best of our knowledge, Minerva represents a unique example in the current research into computational tools applied to museums and cultural heritage. What distinguishes Minerva from other systems are both the extensive adoption of AI techniques and their original application to museum organization. In practice, computational tools are mostly applied in collection management (they are basically database management systems) and for allowing the virtual visits of museums.

In literature, many efforts have been devoted to virtuality. There exists countless implemented systems that offer so-called virtual tours, where the basic idea is that the user can visit a virtual model of a museum (or of a part of it). These tours can enhance the experience of visiting a physical museum or can be made remotely via the web with different degrees of interactivity and personalization. For example, in The British Museum (2006) and The National Gallery (2006) the user can follow a personalized path to view the 2D representations of the works of art. In these cases, the works of art are displayed in HTML pages and there is no connection with a physical museum. In the Museo National del Prado (2006) and the National Music Museum (2006), the addition of a map of the physical museum allows the creation of connections between the works of art and the rooms in which they are actually displayed. Another way to create this connection is to use web-cams (Centre Pompidou 2006) or, more often, VRML representations (Center for Arrhythmia Research 2006; Virtueel Museum Arabesk 2006). All these systems differ from Minerva since they offer the user the possibility to visit predetermined museums (or, at best, museums that can be slightly customized), while Minerva offers the user the possibility, firstly, to build (from scratch) his or her own curatorial organization of a museum’s collection and, secondly, to visit it.

There are systems in which virtuality is adopted for purposes other than virtual tours. For example, Perkins, Bailey, and Price (2003) describe a system devoted to the reconstruction of an historical apartment: in this case, the computational tools are applied not only to the virtual visit but also to the study and evaluation of possible configurations of the apartment. As another example, in Abe et al. (2003), computational tools are used for restoration of relics from slice images. In both cases, however, the degree of autonomy of these systems is not comparable to that of Minerva.

There are some systems that are closer to Minerva. These involve the active role of a user in choosing the works of art. In Virtual Museum of Canada (2006), the user has the possibility to choose some of the works of art to be added to his or her personal collection when visiting the virtual museum. This collection is sequentially visitable in a 2D setting. In
the virtual museum of the Associated Colleges of the South (2005), the user can build a personal collection by choosing both the works of art and the environments in which they are to be displayed. In this case, the “personal” museum is created by the user who manually places the works of art in the environments. Even if the user experiences a high degree of interactivity, the difference with the more powerful partially automatic features of Minerva is evident.

CONCLUSIONS

In this article, we have illustrated a version of Minerva for archeological museums, where traditional and recent AI techniques have been successfully applied to cultural heritage, and innovatively, to the organization of works of art. We implemented Minerva as a modular multi-agent system, in which agents perform inferential activities. We extensively tested the version of Minerva presented here. Archeological experts considered that the results produced by Minerva are acceptably good because they satisfy formal constraints and are sound with aesthetic sensibility.

Future work will be devoted to improve some technical aspects of Minerva. Besides an effort to better the performances of Minerva that is underway, we aim at working on the communication and negotiation aspects of agent interaction and on the automatic retrieval of information concerning the works of art from the web. Furthermore, we aim to apply the same kernel of Minerva to the organization of other museums and collections, where different criteria are employed during the organization process. This extension of the system has three main purposes: to validate Minerva in different contexts, to assess the generality of its architecture, and to enlarge its use to other users beyond museum curators and institutional professionals.

REFERENCES


NOTE

1. The archeological experts with whom we cooperate are Glauco Mantegari and Lucio Perego, who supported us in developing the new version of Minerva presented in this article.