DALICA: Agent-Based Ambient Intelligence for Cultural-Heritage Scenarios

Stefania Costantini, Leonardo Mostarda, Arianna Tocchio, and Panagiota Tsintza,
University of L'Aquila

Agent technology can help users experience cultural assets or monitor the transportation of cultural assets.

Villa Adriana is an enormous archaeological area where ancient artifacts and modern technology have found an unexpected equilibrium. The old artifacts are huge stone monuments; the modern technology includes PDAs and signals from the Galileo satellite combined with intelligent software agents.

As part of the European CUSPIS project (Cultural Heritage Space Identification System, www.cuspis-project.info), we exploited intelligent agents for two ambient-intelligence scenarios applicable to Villa Adriana. One scenario involves *cultural assets fruition*—the possibility of accessing and enjoying cultural assets. This scenario concerns the dissemination of information about cultural assets; for example, users can visit a museum or archaeological site and receive on their mobile devices appropriate, personalized information about that place. The other scenario involves *cultural assets monitoring*, which concerns securely transporting cultural assets from the owner organization to a renter organization and back.

To improve these scenarios, we designed the DALICA multiagent system. In the CAF scenario, an agent assists and guides each user during the visit by suggesting routes and proposing suitable information. The agent works from a standard user profile that it later improves by eliciting the user's cultural awareness, habits, and preferences. The enhanced profile lets the agent update the route and propose customized information, including suggestions for future visits to that or other locations. In the CAM scenario, agents monitor the transportation to enhance security. Sensors in packages containing cultural assets detect environmental data, which agents analyze to infer possible theft. Agents also monitor other practical considerations such as routing.

We've fully implemented DALICA and successfully tested it at the University of L'Aquila and the Villa Adriana area.

Agents and distributed monitoring

Agents and multiagent systems (MASs) are a powerful technology for addressing a variety of complex scenarios that require autonomy. Several industrial applications[1] demonstrate the advantage of using agents. However, agent systems haven't achieved widespread deployment in operating en-

1541-1672/08/$25.00 © 2008 IEEE
Published by the IEEE Computer Society

vironments because the technology has yet to move from pure research to development. DALICA is an interesting example because we've implemented it in a nontrivial, real-world scenario.

Platforms for building autonomous software require dedicated basic concepts and languages. At the individual-agent level, they require representational elements such as observations, actions, beliefs, and goals. Every agent-oriented framework must provide certain functionalities, including

- *reactivity*—an agent's ability to perceive its external environment and respond appropriately to its perceptions,
- *proactivity*—an agent's ability to exhibit goal-directed behavior by taking the initiative according to suitable conditions, and
- *social ability*—an agent's ability to communicate with other agents with suitable modalities.

At the next higher level, new applications need intelligence (the ability to exhibit, compose, and adapt behaviors) and must be able to learn without being instructed how to appropriately perform a task.

Among potential applications, distributed monitoring and control systems (DMCSs) appear to be a natural realm for agents because controllers are autonomous. You could consider the DALICA scenario to be a distributed monitoring system. For example, in the CAF scenario, the object of the agent monitoring activity is to assist the user during the visit. The degree of control here is loose—the system can advise the user to follow a route or view a piece of art, but the user remains autonomous. (The degree of control could increase in the future when we extend the system to assist disabled people.) However, the system must check that users don't trespass forbidden areas, crowd into limited areas, or violate public visiting hours. It must also inform users of interesting events, such as a concert, and unlucky events, such as a closure for security reasons or inclement weather.

Our DMCS needs intelligence to interact with users in a flexible, customizable, and evolving way, rather than through predefined, rigid, or unalterable patterns. Computational logic can be relevant in this sense—it's a good tool for building intelligent agents. Logic languages are also good candidates for such advanced applications owing to their fast prototyping, efficient implementation, and embodiment of new concepts.

Modeling the environment

We put our system to work in a complex, heterogeneous environment; here, we discuss how to model such an environment. The environment's distributed nature led us to accept Jacques Ferber's definition—that you can model an environment as a set of cells assembled into a network.[2] So, in DALICA, we interpreted the environment as a set of specialized cells. For instance, a cell consists of the ontological description of the site where the agents are put to work (for example, Villa Adriana), where monuments

Among potential applications, distributed monitoring and control systems (DMCSs) appear to be a natural realm for agents because controllers are autonomous.

and services are mapped. But, for the CAF scenario, we also had to consider the perceptions (positions and preferences) from each visitor's device as belonging to a particular cell. We could also consider as specialized cells the global database containing visitors' profile information and, in the transport scenario, the perceptions coming from the devices that control the artwork's safety. The abstraction levels of the information coming from different cells trigger suitable activity in the DALICA agents.

We can formalize this multiple-layer concept using an approach that decomposes the environment into building blocks called *environment abstractions*.[3] Each environment abstraction is an entity encapsulating some function or service for the agents. From this perspective, the cells we defined earlier form several different environment blocks or layers. The MAS middleware layer lets us formalize the underlying infrastructures.

The environment can provide three levels of support:

- The *basic level* includes external resources that interact with the MAS. For us, these resources include the visitors' PDAs, the repository of transport digital certificates, the global database, and all other hardware components of the overall CUSPIS system (including the MAS).
- The *abstraction level* bridges the gap between the agent abstraction and low-level deployment details. This level includes, for example, the MAS ontology, the authentication infrastructure, and the communication infrastructure.
- The *interaction-mediation level* encompasses the mechanisms for mediated interaction. This level includes the MAS infrastructures for organizing the visitors' activities in the CAF scenario or for coordinating the cultural-assets transport in the CAM scenario.

Above the environment levels are the working agent applications. We model the environment using a synthesis of previously described approaches: we see the environment as consisting of several contexts, each providing not just data but rather (or also) functions and services, at various abstraction levels.

Figure 1 shows the infrastructure for environment abstractions in the CUSPIS MAS application, for the CAF scenario. The physical layer specifies the hardware components exploited by the CUSPIS MAS (that is, the visitors' PDAs, the Galileo infrastructure, and so forth); we've integrated DALICA into this layer. The PDA receives the Galileo signal and transmits the visitor's position to CUSPIS, which delivers it to the CUSPIS MAS.

The execution platform specifies the operating systems, the virtual machine, and other middleware.

The CUSPIS MAS application consists of the application agents and the application environment. There are three types of application agents. The *generator agent* automatically generates the user profile agents when a user initiates a visit. The *user profile agent* deduces visitors' interests and monitors their behavior. The *output agent* manages communication between DALICA and external infrastructures.

The application environment has three components. The *ontology interface* gives agents information about the Villa Adriana context. The *visitor data interface* communicates to the CUSPIS MAS the visitors' positions and data (such as responses to

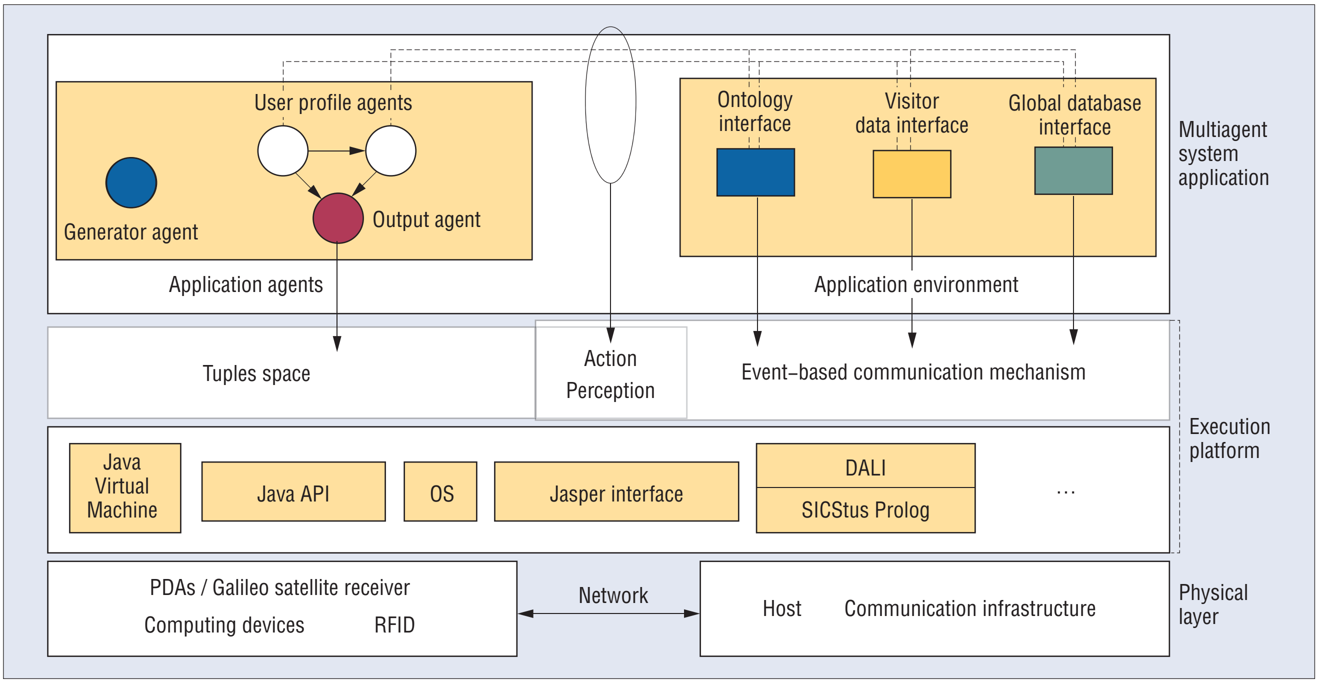


Figure 1. Multiagent-system environment abstraction layers for the cultural-assets-fruition (CAF) scenario.

questions, interests, and so forth). The *global database interface* contains the user profiles and other global information.

Improving cultural-assets fruition

To help agents understand Villa Adriana, we've represented the domain knowledge as a set of *points of interest*. Each POI represents a specific cultural asset or some public place (such as a restaurant) near a cultural asset. A POI includes these fields:

- *Identifier* is a string that uniquely identifies the POI.
- *Latitude* is the POI's latitude defined through the Galileo satellite.
- *Longitude* is the POI's longitude defined through the Galileo satellite.
- *Radius* is the radius of a circle that identifies the POI (we explain this in detail in the next section).
- *Keywords* is a list of the POI's characteristics—for example, "mosaic" if the POI contains a mosaic or "water" if it includes a fountain or water basin. A POI might have several keywords, each weighted according to its relative importance (expressed as a percentage and determined by experts). For example, assuming that visitors are usually attracted to the Pecile for its water basin, while its mosaic has a marginal role, the keyword list would be [(*water*, 60), (*garden*, 30), (*mosaic*, 10)].
- *Time for visit* is the average time a user should dedicate to the POI (again, experts provide this information).

The Dalica ontology

We collected the POI descriptions in an appropriate ontology (developed for the Cuspis project by the University of Rome Tor Vergata's Artificial Intelligence and Natural Language Processing group). For instance, the following string defines "Pecile":

poi('*VA_PecileV* 1', 41.942012, 12.774035, 80, [('*mosaic*', 10), ('*water*', 40), ('*statue*', 20), ('*garden*', 10), ('*column*', 20)], 10).

Keywords let us establish the possible similarities between POIs and, consequently, let agents understand if the visitor is interested in a particular feature common to them. For example, if in Villa Adriana a person visits the Pecile, Teatro Marittimo, Canopo, Piccole, and Grandi Terme, you could assume that the visitor is interested in POIs with water. In all these POIs, water has a greater weight than the other keywords.

Constructing and enhancing user profiles

As we mentioned earlier, users provide a basic profile before starting the visit—often via the website where they booked the visit—with basic data about themselves (such as name, age, and profession) and the visit. The user expresses preferences as POI characteristics (such as "I like gardens"). When the visit begins, the system generates a *user profile agent*. The agent elaborates the data from the initial profile, then reelaborates the profile according to new data derived from the user behavior.

New enhanced fruition profiles can replace a user profile while the visitor proceeds on the route.

Deducing visitor interests. Dalica agents are reactive, proactive, and communicative. They can perceive and react appropriately to data from the environment, such as satellite coordinates or the visitor's POIs. Reactivity lets the agents adopt a specific behavior in response to the external perceptions. Proactivity has a main role because the reasoning process that leads to interest deduction is based on the correlation of data coming from the environment, the ontology, and inferential processes.

The agent deduction process comprises three phases: the first represents a basic deduction level that involves six possible indictors of a visitor's interest.

The first indicator is the amount of time a visitor spends at one location. The algorithm for this assumes that visitors might be interested in a POI if they observe it longer than average. Agents can modulate the meaning of "longer" according to the visitor's profile. Clearly, a longer stay could be due to several reasons unrelated to the visitor's interest. So, agents must confirm that interest through other means (including asking the user).

Agents use the Galileo satellite signal to determine which POI a user is visiting. The system identifies each POI by a circle (whose center is defined by a latitude and a longitude) having a certain radius. If the user's PDA shows a position in the circle related to a specific POI, we can assume that the user is visiting that POI. If two or more POIs are close enough that their circles intersect, and the visitor is located within the intersection, the algorithm presumes that the visitor is interested in all those POIs (because the algorithm can't capture the user's intention). The algorithm considers the keywords of all selected POIs to extrapolate the most frequent ones. These keywords represent the assumed user interests that, once deduced, must be confirmed both by subsequent user behavior and other deduction mechanisms.

The second indictor is the visited POIs. The algorithm considers the user's chosen POIs to improve reliability and precision as the user visits more POIs. The algorithm extracts each POI's keywords and asserts the most frequent ones as "deduced interests."

The third indictor is the chosen route. If a visitor chooses a system-proposed route, the agent tries to capture the visitor's interests by studying the POIs along that route.

The fourth indictor is similarity. This algorithm employs a similarity measure to match the interests the user expressed on the website with the ontology. It selects as deduced interests the ontology elements that it considers similar enough.

The fifth indicator is responses to explicit questions. The system can occasionally ask questions about the POIs near the visitor.

The sixth indicator is responses to cultural questions. This strategy considers the visitor's (self-declared) cultural level. This process includes elements such as the visitor's job and age. The agent compares the data acquired from questions and other methods and elaborates it to determine the appropriate specificity of information. We have provisionally identified three degrees of accuracy.

To further refine the user profile definition, the second deduction phase repeats the steps of the first phase and compares the results to those of the first phase. This process selects the most frequent interests that appear in both phases.

The third phase sends the list of most frequent interests to the visitor for partial or total confirmation. The agent manages the selected interests for updating the user profile. Moreover, the agent communicates them to a central system that manages the visitor's information to propose (through the agent) information and POIs closer to the user's desires and expectations.

Monitoring visitor behavior. Dalica agents use their reactive and proactive capabilities for three basic monitoring tasks.

The first is detecting trespassing. If a visitor enters a forbidden area, the agent sends an alert to the visitor's PDA and informs the authorities about the violation.

> Totally eliminating the risks involved during transport is difficult—maybe impossible. This task's unpredictability hides its risks, delays, anxieties, and difficulties.

The second task is monitoring the visitor's route. The agent can track the visitor when he or she chooses a predefined route. The agent's activity adapts to the visitor's speed. When the visitor tours Villa Adriana along a predefined route but finishes it quickly, leaving time to visit other POIs, the agent proposes additional places to visit according to the user profile and the distance. Alternatively, if the agent realizes that the visitor won't be able to visit all the POIs along the route in the predefined time, it sends a warning to the visitor's PDA.

The third task is creating a list of POIs. The agent collects all POIs that a user has visited into a file with texts and images, letting the visitor remember the visit to Villa Adriana.

Coordinating users' activities. We designed Dalica under the assumption that visitors might have common interests and decide to share their time. Dalica user profile agents can detect common interests among users easily: they construct and manage the user profiles and can exchange information about them. During the visit, when an agent has collected enough information about the user, it cooperates with the other user profile agents. This activity aims to discover whether visitors with similar profiles would like to share the rest of their visit with others nearby. This negotiation could be extended to other users' activities such as dining at a particular restaurant or visiting other museums.

Monitoring cultural-assets transport

Totally eliminating the risks involved during transport is difficult—maybe impossible. This task's unpredictability hides its risks, delays, anxieties, and difficulties.

Previously, we combined the Galileo infrastructure with security mechanisms to create the Geo Time Authentication system.[4] The GTA identifies and authenticates cultural assets, protects cultural-asset information, and securely transports cultural assets. We achieve secure transport using the GTA monitoring component. Each package containing a cultural asset includes a device that contains this component and is connected to temperature, humidity, and light sensors. At runtime, the monitoring component

- controls sensor data variation to detect possible opening of the package,
- checks the mutual position among packages to detect possible theft, and
- uses the Galileo signal to check the correct transport routing.

Experiments revealed that the GTA monitoring component can raise false alarms—in particular, owing to unexpected environmental conditions (such as rapid weather changes or sharp vehicle braking) that require intelligent deductions the GTA implementation lacks.[5] So, we enhanced GTA by exploiting intelligent agents' deduction capabilities.

During transport planning, the cultural assets' owner, renter (the entity who desires the cultural assets), and third-party entities (those who vouch for the content and the transport routing) cooperate to produce different certificates. We focus here on the authorization and the transport certificates

```
posE(Lat,Lng,Time,Date,Integrity,_):>
def_position(Lat,Lng,Time,Date,Integrity).
def_position(_,_,_,_,Integrity):-
   Integrity=0,no_correct_signalA.
def_position(Lat,Lng,Time,Date,Integrity):-
   Integrity=1,positionA(Lat,Lng,Time,Date,1).
def_position(Lat,Lng,Time,Date,Integrity):-
   Integrity=2,positionA(Lat,Lng,Time,Date,2).
```

Figure 2. Integrity signal identification via the DALI reactive rule.

because the Dalica agents use them. Each transport package has an authorization certificate listing all the cultural assets in that package, which the system uses to check that the assets are actually in the package. When a transport has been planned, exactly one transport certificate (which contains the correct routing) is generated. The system defines the routing in terms of

- the starting transport area and the related date (the day and hour),
- the areas the transport must travel through and the related dates, and
- the destination area and date.

In our experiments with the Cuspis project, the cultural assets' owner was a museum in Rome, the renter was a museum in Florence, and the third-party entities were the Ministry of Cultural Heritage, the insurance company, and the transport company.

```
check_forbidden_area(Lat,Lng):-
   positionP(Lat,Lng,_,_,_).
check_forbidden_areaI(Lat,Lng):>
   findall(X,clause(forbidden_area(X,_),_),L),
   examine_forbidden_area(Lat,Lng,L).

examine_forbidden_area(_,_,[]).
examine_forbidden_area(Lat,Lng,[A|_]):-
   clause(forbidden_area(A,Li),_),
   belong_forbidden area(Lat,Lng,Li),
   genera_code(I),
   clause(agent(S),_),
   clause(message_forbidden_area(Mfa),_),
   clause(user_terminal(UT),_),
   messageA(transfer,
   send_message(xinfotransfer_message(I,S,UT,Mfa),S)),
   clause(system_address(SA),_),
   messageA(transfer,
   send_message(xinfotransfer_message(I,S,SA,Mfa),S)).
examine_forbidden_area(Lat,Lng,[A|B]):-
   clause(forbidden_area(A,Li),_),
   not(belong_forbidden_area(Lat,Lng,Li)),
   examine_forbidden_area(Lat,Lng,B).
```

Figure 3. Forbidden area checking via the DALI proactive rule.

In the packaging phase, the aforementioned entities, along with the person responsible for transport (RT) and the packaging expert (PE), supervise the packaging of assets. Each package contains

- a set of assets identified by an RFID tag;
- an Asset Board Unit; and
- the temperature, humidity, and light sensors.

The ABU is a powerful mobile device that hosts both the GTA monitoring component and a control device agent. The GTA monitoring component provides security through traditional mechanisms (cryptography, certificates, and a detection algorithm), and the control device agent enhances security through intelligent deductions. In the real implementation, both the GTA component and agent can implement the same security checking, thus providing additional security.

During the journey, each control device agent proactively checks that the route is correct, verifies the cultural assets, checks the sensor data, and verifies the package position. It checks the route by exploiting both the Galileo signal and the transport certificate. In particular, the agent uses the Galileo satellite to check that the package travels through each area at the right time. When verifying the cultural asset, the control agent loads all the cultural-asset IDs in the authorization certificate and verifies their presence in the package.

These activities repeat over time. The GTA monitoring component also checks the routing and verifies the assets. So, the agent provides redundant checks that enhance the system's fault tolerance.

Checking the sensor data determines its variation over time. This variation must not exceed a given threshold that adapts dynamically through agent cooperation. This check ensures that a package is neither opened nor in a dangerous environment.

Verifying the package position ensures that all packages are in the correct position. This process involves agent cooperation. Occasionally, each control device agent sends a message to the other agents requesting their position. It then computes the distance and verifies that the (relative) position doesn't vary. As we mentioned earlier, a variation of the packages' relative positions can imply theft but can also be due to a sudden stop. This situation requires agent reasoning to enhance the system's effectiveness by detecting and avoiding GTA false alarms. When Dalica detects an anomaly, it sends a warning message.

The control device agents can adapt the thresholds to new environmental conditions. If, for example, the environmental temperature exceeds the threshold for all packs, the agents can communicate with each other and conclude that no package has been tampered with because each signals the same temperature. In this case, the agents don't send an alert but rather cooperate with the GTA monitoring component to adapt to the new temperature threshold.

Implementing Dalica

We implemented Dalica in DALI,[6–8] a logical-agent-oriented language. Here we look at our implementation of the user profile agent. This agent receives the Galileo satellite signal via a DALI reactive rule (see figure 2), where `posE(…)` is an external event (that is, the perception of something that happened in the external world). In this case, it's information about the user's position, where `Lat` and `Lng` are the latitude and the longitude, `Time` and `Date` are the time and date, and `Integrity` indicates how well the signal has been received. To define the reaction, we use a reactive rule whose head contains that external event. The special token :>, used instead of :-, indicates that reactive rules perform forward reasoning.

This rule filters the Galileo signal according to its integrity value (indicating the signal's quality). The signal is acceptable if its integrity is different from 0. The system then records the action `positionA(Lat,Lng,Time,Date;_)` and makes the position data available to the proactive rules for subsequent inferential activities. Subsequent detections of the satellite signal will update the position data.

The trespassing check illustrates the agent's proactive capabilities. This check employs an internal event—a pair of rules that let DALI agents reach internal conclusions according to acquired knowledge and experience (see figure 3). The system automatically attempts the first rule in each pair from time to time. If it succeeds, possibly returning some values for input variables, the system then executes the body of the second rule (the reactive one) after assigning the values to the variables. The rule check_forbidden_area(Lat,Lng) is an internal event that is triggered each time the agent receives a new correct position.

The procedure belong_forbidden_area(Lat,Lng, Li) verifies whether the position is in a forbidden area. A positive response forces the agent to send a message to the user's PDA and to the central system for alerting the authorities. In the communication primitive xinfotransfer_message(I,S,SA,Mfa), Mfa specifies the kind of communication act that the agent is performing (in this case, a forbidden-area alert).

Agents at work

Suppose a user is walking near the Pretorio. The ontology describes the Pretorio as

poi('*V A_IlPretorio*', 41.939503, 12.775775, 25, [('*columns*', 0.30), ('*opus*', 0.30), ('*fresco*', 0.30), ('*arch*', 0.10)], 8).

That is, the center of the circle describing the Pretorio is defined by the tuple (41.939503,12.775775) of Galileo coordinates. The circle's radius is 25 meters, and we describe the Pretorio with the keywords *columns*, *opus*, *fresco*, and *arch*. A relevant parameter for deducing that the visitor might be interested in this POI is the time for a visit—assumed to be about eight minutes.

The user's movements are concentrated in the area described by these coordinates:

(41.93948,12.775775);
(41.939476,12.775773);
(41.939487,12.775773);
(41.939487,12.775772);
(41.939495,12.775768);
(41.939503,12.775775);
(41.939503,12.775784);
(41.93945,12.7755765);
(41.93944,12.775759);
(41.939438,12.775752);
(41.93943,12.7757435);
(41.9394,12.77573).

The user's PDA communicates these positions to the user profile agent. The time is the afternoon, after 1:30 p.m. Some positions can repeat because the user might stay in one place.

Figure 4 illustrates the user profile agent's behavior. The agent has already deduced an interest in Grandi Terme and Pretorio Vista, which leads the agent to assume that the visitor is interested in *thermae* (thermal baths), an interest belonging to the POI Grandi Terme. Then, the agent sends the interest *thermae* to the transfer output agent, which dispatches the message to the visitor's PDA. Finally, the agent sends questions about Pretorio Vista and waits for the user's reply to understand his or her preferences and send personalized information.


```
make(deduce_interest_time_in_poi(http://ontologies/heritage#VA_GrandiTerme2)
make(deduce_interest_time_in_poi(http://ontologies/heritage#VA_IlPretorioVista)
make(position(41.93948,12.775775,time(13,38,40),date(15,11,06),1))
make(deduce_interest_time([(thermae,0.25)]))
make(position(41.939476,12.775773,time(13,38,45),date(15,11,06),1))
make(position(41.939487,12.775773,time(13,38,50),date(15,11,06),1))
make(position(41.939487,12.775762,time(13,38,55),date(15,11,06),1))
make(position(41.939495,12.775768,time(13,39,00),date(15,11,06),1))
make(position(41.939503,12.775775,time(13,39,05),date(15,11,06),1))
make(position(41.939503,12.775784,time(13,39,10),date(15,11,06),1))
make(position(41.93947,12.775784,time(13,38,15),date(15,11,06),1))
make(position(41.93945,12.775765,time(13,39,20),date(15,11,06),1))
make(position(41.93944,12.775759,time(13,39,25),date(15,11,06),1))
make(position(41.939438,12.775752,time(13,39,30),date(15,11,06),1))
make(position(41.93943,12.7757435,time(13,39,35),date(15,11,06),1))
make(position(41.93942,12.77573,time(13,39,40),date(15,11,06),1))
send_message_to(transfer,send_message(question(8,1,1,8,VA_IlPretorio),1))
```

(b)

Figure 4. The Dalica multiagent system at work in the CAF scenario: (a) Photos show the satellite overview and details of the Pretorio and Grandi Terme artifacts in Villa Adriana, including the user's route around the buildings. (b) Code for the User Profile agent shows some steps of that agent's deduction activity. Given the time the user stays in Pretorio and Grandi Terme, the agent deduces that the user might be interested in *thermae* (thermal baths).

Although no significant statistics are available yet, experiments have shown that Dalica behaves well and can deduce the visitors' interests adequately. User feedback has been positive—users were impressed by how the system understood their needs. They would have liked better interaction to control the system behavior and ask more questions. We'll consider this feedback in future developments. The users also indicated that they would pay a reasonable price to use the system but wouldn't pay for the information.

We've also successfully demonstrated Dalica in Villa Adriana to European Community officers and to various stakeholders, including local institutions, Cuspis partners, representatives of the Italian Ministry

Related Work in Monitoring Cultural Assets

Several projects have proposed using technology to enhance cultural assets. Doyun Park and colleagues propose an "immersive tour post" system in which a post fixed in one location reproduces the images and sounds of the historical events that occurred there.[1] Giovanni Pilato and his colleagues have experimented with mobile applications in a mobile environment.[2]

MAGA (Mobile Archaeological Guide at Agrigento), a user-friendly virtual-guide system, assists visitors in their routes at the Parco Archeologico della Valle dei Templi, an archaeological area with ancient Greek temples in Agrigento, Sicily. MAGA exploits speech recognition and location detection.[3] Minerva organizes virtual museums, starting with collections of objects and the places in which they must be displayed.[4] Unlike our Dalica system (see the main article), MAGA and Minerva don't try to deduce the visitors' interests.

In the DramaTour methodology, a virtual spider assists visitors by monitoring their behavior and reactively proposing historical information and amusing anecdotes.[5]

These systems build on traditional methods by using new, amusing techniques for making the human-machine interface friendlier and more intuitive. Arvind Bhusate, Lloyd Kamara, and Jeremy Pitt go further by giving each visitor a PDA associated with noninvasive sensors that measure physiological context data such as the user's skin conductance and temperature.[6] The system reports the sensor readings to a control module that determines, according to other data, the visitor's mood.

Oliverio Stock and his colleagues helped develop the Peach (Personal Experience with Active Cultural Heritage) system through experiments in the Castle of Buonconsiglio in Trento, Italy.[7] Peach employs "interest propagation;" it propagates a positive or negative degree of interest toward parts of the presentation templates to all related templates. As in Dalica, user profile evolves on the basis of the visitor's interest level for various concepts and instances, but the system focuses on the presentation while the Dalica MAS focuses on the users' monitoring.

The KORE system considers parameters such as age, cultural level, artistic preferences, and preferred historical period to fine-tune user-provided information.[8] Its architecture is based on a distributed system of servers, installed in various areas of a museum, which host specialized agents. We believe KORE is closest to Dalica because agents manage the information by studying a user profile. However, KORE doesn't exploit the Galileo signal, and its agents aren't written in a logic language.

References

1. D. Park et al., "Designing an Immersive Tour Experience System for Cultural Tour Sites," *CHI '06 Extended Abstracts Human Factors in Computing Systems*, ACM Press, 2006, pp. 1193–1198.
2. G. Pilato et al., "An Intelligent Multimodal Site-Guide for the Parco Archeologico della Valle dei Templi in Agrigento," *Proc. 1st European Workshop Intelligent Technologies for Cultural Heritage Exploitation, 17th European Conf. Artificial Intelligence*, 2006.
3. A. Augello et al., "MAGA: A Mobile Archaeological Guide at Agrigento," presented at Mobile Guide 06, 2006; http://mobileguide06.di.unito.it/pdf/Augello&al.pdf.
4. F. Amigoni, S.D. Torre, and V. Schiaffonati, "Yet Another Version of Minerva: The Isola Comacina Virtual Museum," *Proc. 1st European Workshop Intelligent Technologies for Cultural Heritage Exploitation, 17th European Conf. Artificial Intelligence*, 2006, pp. 1–5; www.airlab.elet.polimi.it/index.php/publications/yet_another_version_of_minerva_the_isola_comacina_virtual_museum.
5. R. Damiano, C. Galia, and V. Lombardo, "Virtual Tours across Different Media in DramaTour Project," *Proc. 1st European Workshop Intelligent Technologies for Cultural Heritage Exploitation, 17th European Conf. Artificial Intelligence*, 2006, pp. 21–25; http://dramatour.di.unito.it/pdf/2006-ECAI-WS.pdf?DRAMATOUR_PHP=15c2229910b63eece1d48328dc8d8337.
6. A. Bhusate, L. Kamara, and J. Pitt, "Enhancing the Quality of Experience in Cultural Heritage Settings," *Proc. 1st European Workshop Intelligent Technologies for Cultural Heritage Exploitation, 17th European Conf. Artificial Intelligence*, 2006, pp. 1–13.
7. O. Stock et al., "Adaptive, Intelligent Presentation of Information for the Museum Visitor in Peach," *User Modeling and User-Adapted Interaction*, vol. 17, no. 3, 2007, pp. 257–304.
8. M. Bombara, D. Cal, and C. Santoro, "KORE: A Multi-Agent System to Assist Museum Visitors," *Proc. Workshop Objects and Agents* (WOA 03), 2003, Pitagora Editrice Bologna, pp. 175–178.

of Cultural Heritage, and a delegation of the Chinese Ministry of Cultural Heritage. For the demonstration, we based the Cuspis system deployment on a server and three user terminals equipped with Egnos (European Geostationary Navigation Overlay Service) receiver and communication devices. We demonstrated the CAM scenario on 31 January 2007 with secure transportation from the Scuderie del Quirinale Exhibition in Rome to the Villa Adriana Museum. For more information on the Cuspis demonstrations, see www.cuspis-project.info/demonstrations.htm.

Future improvements will deal with both the algorithms and the data representation. In particular, we plan to make the POI descriptions more complete and informative to support useful inference about the POI interconnections. Because a Cuspis central server collects the Dalica data, data mining could give agents significant information for use in initial profiles—but also goes beyond a single-user visit to be used in future visits or for statistical aims. We plan research tied to social computing to better relate different users' activities and to supervise and coordinate group visits. Also, agents might exploit their planning capabilities to suggest alternative routes to users who are late or in a hurry.

Dalica should be available for general use in the near future. For a description of other systems for monitoring cultural assets, see the sidebar. ■

Acknowledgments

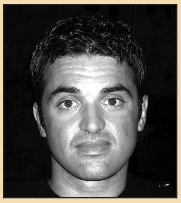
The Cuspis (Cultural Heritage Space Identification System) project GJU/05/2412/CTR/CUSPIS partially supported this work.

References

1. H. Van Dyke Parunak, "Practical and Industrial Applications of Agent-Based Systems," Industrial Technology Inst., 1998; http://agents.umbc.edu/papers/apps98.pdf.

The Authors

Stefania Costantini is a full professor in the University of L'Aquila's Dept. of Computer Science. Her research interests include computational logic and AI, agents and multiagent systems, and answer set programming. She received her master's degree (the highest possible degree in Italy at the time) in computer science from the University of Pisa. She was a cochair for the Answer Set Programming 2007 conference. Contact her at the Dept. of Computer Science, Univ. of L'Aquila, Via Vetoio, Loc. Coppito, I-67010 L'Aquila, Italy; stefcost@di.univaq.it.



Leonardo Mostarda is a postdoctoral student in computer science at the University of L'Aquila and a research associate in Imperial College London's computing department. His research interests include distributed systems, monitoring systems, and security. He received his computer science degree from the University of L'Aquila. Contact him at the Dept. of Computer Science, Univ. of L'Aquila, Via Vetoio, Loc. Coppito, I-67010 L'Aquila, Italy; mostarda@di.univaq.it.

Arianna Tocchio is a postdoctoral student in computer science at the University of L'Aquila and a research assistant at the university. Her research interests include logic agents and multiagent systems and learning. She received her computer science degree from the University of L'Aquila. Contact her at the Dept. of Computer Science, Univ. of L'Aquila, Via Vetoio, Loc. Coppito, I-67010 L'Aquila, Italy; tocchio@di.univaq.it.

Panagiota Tsintza is a PhD student at the University of L'Aquila. Her research interests are multiagent systems, computational-logic agents, and automated negotiation. She received her master's degree in computer science from the University of L'Aquila. Contact her at the Dept. of Computer Science, Univ. of L'Aquila, Via Vetoio, Loc. Coppito, I-67010 L'Aquila, Italy; panagiota.tsintza@di.univaq.it.

2. J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, 1999.
3. M. Viroli et al., "Infrastructures for the Environment of Multiagent Systems," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 1, 2007, pp. 49–60.
4. L. Mostarda et al., *A Geo Time Authentication System*, Springer, 2007.
5. "Cultural Heritage Space Identification System (CUSPIS)," *European Commission 6th Framework Program—2nd Call Galileo Joint Undertaking*, www.cuspis-project.info.
6. S. Costantini and A. Tocchio, "A Logic Programming Language for Multi-Agent Systems," *Logics in Artificial Intelligence, Proc. 8th European Conf.*, LNAI 2424, Springer, 2002, pp. 1–13.
7. S. Costantini and A. Tocchio, "The DALI Logic Programming Agent-Oriented Language," *Logics in Artificial Intelligence, Proc. 9th European Conf.*, LNAI 3229, Springer, 2004, 685–688.
8. A. Tocchio, "Multi-Agent Systems in Computational Logic," PhD thesis, Dipartimento di Informatica, Università degli Studi di L'Aquila, 2005.