HyperGuide: a context-aware semantically interoperable multimedia application for the fruition of cultural heritage

Giorgio Ventre
Università degli Studi di Napoli “Federico II”
Napoli, Italy
giorgio.ventre@unina.it

Francesco Gragnani and Vincenzo Masucci and Claudio Nardi and Vincenzo Orabona
Consorzio Campano di Ricerca per l’Informatica e l’Automazione Industriale
Piazza Enrico Fermi 1, Portici, Italy
c.nardi@criai.it

Abstract

In this paper we present a semantically interoperable software application, codenamed HyperGuide, for supporting the fruition of cultural heritage through proposition of context-aware multimedia content. Semantic languages have been used to achieve semantic interoperability for data and for some subsystems. The application was designed and developed at the CRIAI research lab in Portici (Naples), Italy within a research program funded by the Italian Education, University and Research Ministry. The program is currently entering its experimental phase when the application is being adopted in one of the archeological sites of the Soprintendenza archeologica of Pompei. In HyperGuide context is a multi-dimensional entity which refers to three aspects: the user’s profile, the user’s device capabilities, and the user’s real-time geographical position. Multimedia content is described and stored through standard ontologies and languages. The paradigm of device independence is exploited to provide a standard description of the device’s capabilities while the user’s preferences are captured through a lightweight registration process. When the user accesses the system, his/her position is determined through a fine-grained indoor and outdoor location system. At this point, multimedia content is retrieved and content presentation is automatically adapted based on the context.

1. Introduction

Cultural tourism is a growing segment of the tourism industry and the one that offers the best opportunities to local communities in terms of economic development and preservation of local landscapes and lifestyles. However, the visitor to cultural heritage sites is often frustrated at the lack or inaccuracy of the relevant information. As a result the visit is not tailored to the tourist sensibility and education and this, in turn, may take him/her away from this kind of experience. In recent years though, rapid advances in technology, from telecommunications to multimedia, hardware, speech synthesis and recognition, and graphics have proven useful in documenting, preserving, and disseminating our valuable culture heritage. In this paper we present a semantically interoperable software application, codenamed HyperGuide, for supporting the fruition of cultural heritage through proposition of context-aware multimedia content. The application was designed and developed at the CRIAI research lab in Portici (Naples), Italy within a research program cosponsored by the Italian Education, University and Research Ministry [1]. The program is currently entering its experimental phase when the application is being adopted in a real-life scenario, in collaboration with the Soprintendenza archeologica of Pompei. Several ICT research projects all over Europe are dedicated to cultural heritage sites [2–5]. However, HyperGuide features many aspects of novelty. First, its multimedia repository is based on the semantic Web specifications. Second, its location system works well both indoor and outdoor. Furthermore, vocal technologies are exploited to make the user’s interaction with the system more natural and content presentation is adapted to the end-device. Finally, the adoption of open source software and of standard protocols and recommendations has been a key factor in the design and development. The paper is organized as follows. Section 1 (this chapter) gives a brief introduction. Section 2 presents the overall system architecture. In section 3, our approach to the mul-
timedia digital repository is explained. The location and tracking system is described in section 4. Section 5 is about content presentation in HyperGuide, with speech technologies and device independence being key drivers. Conclusions are drawn in section 6 where future work and directions are also mentioned.

2. HyperGuide system architecture

A logical view of the HyperGuide application and its main interactions is given in figure 1. The application main logical components are distinguishable: a three-tiered, J2EE-compliant Web application; a Geographic Information System (GIS) for maps visualization and user’s position rendering; a multimedia digital library for content store and retrieval; a location and tracking system; a voice engine with a text-to-speech (TTS) system and an Automatic Speech Recognition (ASR) system. Given its multichannel nature, we represent HyperGuide different access mediums (Internet, WLAN, UMTS, GPRS, etc.) and techniques with an access gateway. The diagram also represents two different types of users: the remote user and the on-site user. The former mainly logs on to the system through a desktop/laptop PC with a standard browser and can surf the Web application viewing content like pictures, video, and of course text. Audio files are also available with the proper browser plug-in. Remote PC users with a soft-phone may also exploit the vocal interaction. On-site users present static as well as dynamic characteristics (position, interest in some place or theme, route within the site, etc.). Their position can be determined with high accuracy and their mobile devices’ capabilities can be described in a standard way, thus resulting in a richer user’s context and ultimately in a better user experience.

To some extent, HyperGuide is a Web application implementing a virtual visit to an archeological site. For our demonstrator, we chose the archeological site of Pompeii because of its richness, popularity, and complexity of experimental scenarios, but the application may easily be adapted to other sites. When a user accesses the system through a standard browser, a personal assistant guides him/her through a lightweight registration process which allows him/her to express his/her preferences in terms of language, education, age, and level of details for the visit. Other options like fruition mode (push or pull) and type of tour (e.g. generic tour, religious tour, etc.) are also selected at this time. However, if the user is visiting the site and his/her device fulfils some hardware requirements, a new scenario unfolds. A thin software program on the mobile station sends the system sensing information for it to determine the device’s location. Simultaneously, another software module provides the system with a standard description of the device’s capabilities, including hardware components for the location determination (e.g. wireless/GPS receiver). Hence, the user’s context is fully developed and the application engine may query the location system for the user’s location, the digital library for the relevant multimedia content and the GIS system for the user’s rendering on the site map. Before content can be presented back to the user, content adaptation also occurs. From a static, generic tour, the visit is transformed to a dynamic, customized one where the user can freely move around the cultural heritage site and receive the pertinent information. Users unable or simply unwilling to look at a device’s screen and read on it may resort to the vocal interaction, which is handled by a voice over IP (VoIP)-based, bilingual speech system including an ASR and a TTS. Here, a vocal dialog is started, which first enables the user to make his/her choices of language and level of detail. The user can then move around the site and ask the virtual guide for more information. As a result, the user is presented with a list of nearby cultural resources from which to choose by just uttering their name or part of their name. Again, the recognition engine will understand the user’s choice and select the content from the digital library much in the same way as a Web server does for a Web browser client. Finally, all the available information on the relevant resource, characterized by the previously selected language and level of detail, is provided to the user in the form of synthesized speech. The vocal system supports advanced features like barge-in, which allows to listen to the client utterances while still synthesizing speech. Hence, the user may interrupt the system at any time, for instance because he/she want to move on to another cultural resource in the list, by simply saying the right keyword.
3. Multimedia digital library

Our repository of multimedia contents is the Multimedia Digital Library (MDL). Instead of the traditional database-like approach to data management, we chose the semantic Web paradigm [6]. Multimedia data is stored and accessed through an Ontology Storage and Querying System (OSQS). This has the multiple advantages of semantic expressiveness, effective data querying and retrieval, and increased flexibility in that heterogeneous storing systems may be supported more naturally. In addition, this architecture allows the MDL to smoothly evolve towards a Web service model. Our first objective was to design a new ontology for the conceptualization of our domain of interest. We selected several knowledge domains: the user visiting a cultural heritage site, the user’s electronic device, the cultural resources (artifacts, manufacts, buildings, wall paintings, etc.) contained in the site, the multimedia contents (images, audio, video, etc.) associated to the cultural resources and presented to the user’s device, the themes (religion, buildings, food, etc.) to which cultural resources can be related, and finally the cartographic data, which indicate with great accuracy the position of cultural resources in the site. The next step was to create an ontology for each domain. Therefore, we determined the relations among entities and we associated a metadata standard to each domain. MPEG-7 [7] is particularly suitable to represent multimedia data related to cultural resources [8]. Dublin Core is also fit to cultural resources representation [9], and its open and property-centric features give it a high degree of flexibility. Hence, we chose MPEG-7 for multimedia data, Dublin Core application profiles for data about cultural heritage, user, and thematic tours, and GML for cartographic data [10]. The integrated ontology was obtained by adding custom properties and joining the relevant concepts of each ontology. Open source platforms with Java API were considered to select a suitable OSQS. We experimented with both the Sesame framework [11] (rel. 0.9.x) and related Sebor reasoner [12], and the Jena framework [13] (rel 2.1). The latter has the big advantage of supporting OWL extensions [14] and custom defined inference rules, thus dramatically extending the knowledge base. Following releases of Sesame, enhanced with a reasoner engine, are under evaluation. In fact we realized prototypes of the MDL with both platforms. Multimedia data, of different formats and sizes, were stored on a file system and described through RDF [15], while the ontology defined above was represented in RDFS [16]. The query engine of the prototypes could access and retrieve multimedia data. Both the OSQSs adopted have shown good performance and high stability and reliability. Finally, we conducted some simple performance tests, based on a common RDF knowledge base. The results showed that the RDQL query engine was faster than the RQL one on the Sesame platform. They also showed a better performance for the Sesame RDQL query engine than for Jena’s.

3.1. Integration between MDL and GIS

In HyperGuide, the main goal of the GIS is to provide the user with geographically enriched context-aware information about cultural resources. A map may be rendered on the user’s device screen with selectable points of interests, and the relevant information may be obtained. In addition, the GIS exports functionalities to obtain the URIs (Uniform Resource Identifier) [17] of all the cultural resources in a geographic surround of the user, for instance a circle or a rectangle centered in the user current location, as provided by the location system. We adopted a freeware GIS, namely ALOVMap [18], because of its support of standard formats (e.g. shapefile) and because it is compliant with the Web Map Service (WMS) [19] specifications. WMS allows access to maps and other GIS functionalities from any device, including mobile ones, through the HTTP [20] protocol. Moreover, it is very useful to assign some appropriate significance to points and curves drawn on the map, e.g. a line may represent the boundary of a building or its access doorway. Likewise, a cultural resource may be related to one or another particular theme or tour. To capture the variety of aspects and the semantic significance of our knowledge domain, we represented spatial data in a standard XML-based encoding called Geography Markup Language (GML) [21], introduced by the Open GIS Consortium (OGC) [22]. Then we converted those data to RDF for succeeding storing to the DML. As a result, it was possible to load geographical information to the DML and make that information available to other applications, e.g. the virtual visit. Data stored in the DML was also made accessible through the GIS interface with its associated tools. The tight integration of the DML with the GIS has several advantages: first, spatial data are given semantic significance while still keeping the accuracy and precision typical of a GIS. Second, it is possible to extend that significance by making inferences or even by defining custom inference rules. Third, all data in our knowledge domain present the same common interface since they are stored in the same repository. Finally, the adoption of GML guarantees flexibility, interoperability among heterogeneous systems, and independence from presentation.

4. Location system

The location system provides location and tracking of the user’s device both indoor and outdoor. The system has been designed from start to support any number of sensing technologies. In fact it currently leverages on two different
technologies to get the best available location estimate: the Global Positioning System (GPS), which is available in external environments, and radiofrequency, in the form of an IEEE 802.11b [23] wireless local area network (WLAN) infrastructure. A conceptual view of the location system is provided in figure 2. The mobile stations communicate with a location server by sending it the relevant sensing information. This may be an NMEA [24] string from the on-board GPS receiver, or the signal strength read at the 802.11b receiver. Either case, the location server, which is technology-neutral, does not elaborate the information itself, but instead forwards it to the specific technology-aware server responsible for the actual position determination, e.g. a GPS location server or a WLAN location server. The position estimate, represented in a standard format (e.g. latitude and longitude), is sent back to the location server, which keeps track of each mobile device’s. For some devices (e.g. those equipped with both receivers), several position estimates may be available, in which case the location server selects the best one. In this way, any LCS client, willing to know the position of a mobile station, may query the location server with the device’s MAC address as input. The Service Location Protocol (SLP) [25] is used in our system as a lightweight discovery mechanism for retrieving the location server transport protocol address (IP address and TCP port). The adoption of SLP gives the application a great flexibility, since no hardcoded information or static configuration files are required on the mobile stations or anywhere else. Furthermore, the distributed nature of the system’s architecture improves scalability and performance.

4.1. Components of the location system

The main components of the location system are: the GPS location server, the WLAN location server, the location agent, and the LCS client. Let’s see each of them in more detail. The GPS location server receives NMEA strings from a mobile device equipped with a GPS receiver, it makes consistency checks, and extracts the device’s absolute position as a tuple latitude and longitude, together with the exact time of the fix. The server then uploads this information to the location server. The WLAN location server converts real-time Received Signal Strength Information (RSSI) coming from several access points, to the receiving device’s absolute position estimate. This estimate is then sent to the location server, in the same way the GPS server does. The location agent is a lightweight software program running on the mobile station, whose main task is to read sensing information from the on-board GPS/WLAN receiver(s) and to transmit it to the previously discovered location server. Finally, the LCS client is the entity concerned with a device’s position. Through SLP, it discovers the location server and obtains the absolute coordinates of the input mobile device.

4.2. Experimental results with the WLAN Location system.

In our work we investigated several techniques, which we call filters, to determine the user’s position inside a building with a IEEE 802.11b wireless network infrastructure. We designed, implemented and tested several such filters. We first built a database of received signal strength (RSSI) which we call fingerprints map or simply map. In addition, we implemented the well-known K-Nearest Neighbour algorithm in the signal strength space [26]. For $K \geq 2$, this algorithm computes the barycenter of the K-Nearest points belonging to the fingerprints map, where “near” is associated with a particular metric. We chose the Euclidean distance but other metrics may also apply. In its simplest form, our system works as below. RSSI information from a mobile device is sent to a server which is also given the fingerprints map as input. The server then estimates the device’s location using the K-Nearest Neighbour (KNN) algorithm with K set to a known value. The system can freely combine more filters. For instance it may use the circle filter and/or the signal strength filter to achieve a better accuracy. The latter focuses on the signal strength spectrum by comparing the RSSI spectrum in the map and the one in the real-time sample. The comparison is limited to a subset of cardinality M of access points, hence M is a parameter of this algorithm. This filter only considers those points in the map for which the M highest and the M lowest RSSI values are relevant to the same access points as

![Figure 2. logical view of the location system](image-url)
the M highest and the M lowest RSSIs in the real-time sample. In other words, for $M = 2$ let us suppose that AP1 and AP2 are the access points with the two highest signal strength values for the real time sample (order is unmeaningful here), while AP5 and AP6 are the ones with the two lowest values. This filter selects the locations in the map where AP1 and AP2 have the two highest signal strength values and AP5 and AP6 the two lowest values. The circle filter is a software implementation of the triangulation technique using the RF signal propagation model in a vacuum. For each access point the algorithm draws a circle whose radius depends on the RSSI sample and whose center is at the access points itself. A system of equations representing the circles is solved and a shared area is output. If no solution is found, then some kind of heuristics is used until a solution is obtained. To determine the device’s location we may use any triangulation algorithm but this is where the filters concept comes into play again. Starting from the shared area, we use the fingerprints map to filter out points outside the map. Finally the K-Nearest Neighbour filter is exploited to estimate the device’s position. In order to validate our approach, we conducted some experiments at the ground floor of the CRIAi building in Portici, Naples. The building is 28.80 metres long by 25.20 metres wide and is divided in several different-sized offices and corridors with a large open space in the center. We first installed six wireless access points. For our experiments we used a Toshiba e800 WiFi PDA (Personal Digital Assistant) with the PocketPC 2003 operating system. We finally implemented a program that would read RSSI values on the PDA and send them to the server where the location estimation algorithm was running. With this setup, we built a RSSI fingerprints map with pace 1.80 metres. We then ran tests using these maps and the same K-Nearest Neighbour filter and the signal strength filter with $M = 2$. We considered 16 locations equally distributed throughout the building. To assess the location accuracy, we computed the average error distance from the actual position for each of the 16 locations. The results hinted that the signal strength filter may improve accuracy substantially. In fact, the application of the KNN algorithm alone yielded overall average distance error of 1.88 metres and a standard deviation of 2.08 metres. Applying the signal strength filter resulted in an average distance error of 1.68 metres and in a standard deviation of 1.93 metres.

5. Content presentation

In HyperGuide context is a multi-dimensional entity which refers to three aspects: the user’s profile, the user’s device capabilities, and the user’s real-time geographical position. Once the context is acquired, and the appropriate multimedia content is selected, content adaptation occurs. This chapter is devoted to describing how and to which extent this happens.

5.1. Device independence

The driver to our content adaptation approach is the paradigm of device independence as proposed by the W3C CC/PP (Composite Capabilities / Preferences Profile) working group [27]. That provides an RDF framework for the extensible description of a device’s delivery context. This is often referred to as a CC/PP profile. Both static and dynamic characteristics are considered. In this project we introduced a new CC/PP-compliant vocabulary which we called ArcheoLocation. Here, in addition to standard device capabilities, we introduced two new dynamic attributes: GPSCapable and WiFiCapable. If GPSCapable equals zero, then the device does not have a GPS receiver or its GPS receiver is switched off, but if GPSCapable equals one, then the receiver is turned on and it is acquiring a valid fix. Likewise, if WiFiCapable equals zero, then the device does not have a WLAN receiver or its WLAN receiver is turned off, but if WiFiCapable equals one, then the device’s WLAN receiver is working. We implemented a C++ dynamically linked library which exports functionalities to read from the GPS/WLAN receivers on a PocketPC PDA. On the server side, we adopted the Deli [28] library, which can parse and store CC/PP-compliant profiles. We successfully tested the above configuration with an iPAQ 5550 PocketPC PDA running a Java Virtual Machine (JVM): the device sent an RDF description of the ArcheoLocation profile over the CC/PP-exchange protocol [29], and the server was able to capture it.

5.2. Content adaptation

In accordance with the W3C activity on content presentation, we used XML to collect and describe dynamic content and XSL (eXtensible Stylesheet Language) and XSLT (XSL Transformations) [30] to transform and to present that content. Let’s see in more detail what happens when an HyperGuide on-site user accesses the Web application through a standard browser on a mobile device. The personal assistant has helped him/her make the right choices for him/her, for instance an expert level of detail, the English language and the religious tour. Let’s assume the user is close to the cultural resource X, a temple, and to Y, a villa. The application selects expert content related to resource X only, and discards the other. Non-English texts are also filtered out. The application generates a XML document including this information, which a XSL file transforms for rendering on the user’s device. Long texts, which will not fit to the small device’s screen, will be split in several parts, while pictures will be properly rendered.
5.3. The speech system

In our project we deployed a bilingual speech system powered by the Loquendo ASR and TTS technology. We chose the VoIP technology instead of a more traditional telephony-like approach because of its flexibility, simplicity of integration with other Web applications, and scalability. We exploited the Loquendo system as a black box with the VoiceXML 2.0 [31] interpreter being the API layer. Grammars for the speech recognizer were developed in JSGF [32]. We implemented several vocal services. In particular, the vocalOnSite service showed the tight integration of the speech system with the digital library, with the GIS system and with the location system. Cultural resources in the user’s proximity only were selected so that context-aware, multimedia data could be retrieved and then synthesized to the user. A site map including the real-time user’s position was also shown to the user’s device. To make the user-system interaction more natural, the technique called barge-in was also exploited and the prosody level was fine tuned. Since a high degree of dynamicity was required, the vocalOnSite service was accurately designed so that the voice dialog and the grammar files were generated on the fly, based on the system status. Dynamic grammar files have the advantage of increased performance, higher confidence level and, as a result, they enable slimmer dialogues since there is no need to ask the user for confirmation.

6. Conclusions and future work

This paper provided a description of HyperGuide, a semantically interoperable application supporting context-aware, multimedia applications for the fruition of cultural heritage. Although work is still in progress, the project has already achieved important milestones, like the introduction of a multimedia digital library based on the semantic Web paradigm, a prototype location and tracking system for indoor and outdoor, a demonstrator of vocal interactions. We were especially pleased with the results obtained with the WLAN location system, since they demonstrate the validity of our approach. We expect more work ahead. As far as the digital library is concerned, we are experimenting on custom defined inference rules within the Jena framework. With regard to the location system, we are working on extending the WLAN system’s operations to a limited range around a building with a wireless network infrastructure. We are finally planning to realize a queryBuilder, that is a module for dynamic proposition of context-aware thematic tours.

References


http://www.miur.it/.


