

SPOT-IT: Going Beyond the Vision Loss Boundaries

Michal Tvarožek

Martin Adam

Michal Barla

Peter Sivák

Mária Bieliková

Faculty of Informatics and Information Technology,
Slovak University of Technology
Ilkovičova 3, Bratislava, Slovakia

+421 902 802 058

+421 904 655 209

+421 903 843 276

+421 904 824 827

+421 2 654 295 02

tvarozek@gmx.net

martin@atrip.sk

crutch@seznam.cz

psivak@netcomga.sk

bielik@fiit.stuba.sk

ABSTRACT

The volume of available information is growing daily and it thus becomes increasingly difficult to organize, search and access this information. In this paper we present a novel concept which allows for acquiring, organizing and finding relevant information, related to a particular physical environment efficiently. We describe the SPOT-IT system, which uses the devised concept to offer aid to the many visually impaired or completely blind people living in the world today. The purpose of the SPOT-IT system is to provide blind and low sighted people with contextual information about their surroundings, thus enabling them to know (i.e., “to see”) what is around them. Such newly gained information will in turn enable them to be more independent and self-sufficient, since they will no more have to rely on others to provide them with this information or perform tasks based on this information. We present basic principles of the devised concept and the design of the SPOT-IT system together with a discussion of its marketability. The system employs several on-coming technologies such as RFID, data transfer through the human body and introduces several notable innovations including a novel way of associating information with real-world entities.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *information filtering*; H.4.3 [Information Systems Applications]: Communications Applications; H.5.4 [Information Interfaces and Hypermedia]: Hypertext/Hypermedia

General Terms

Algorithms, Design, Experimentation, Security, Human Factors

Keywords

Information acquisition and providing, RFID, RFID Name Service, RFID Tag Service, Contextual Information Description Language, web service, blind aid, contextual information, personal area network, human body transmission.

1. INTRODUCTION

In order to respond to the challenge of present and future information processing needs, we devised a concept which allows for acquiring, organizing and finding relevant information, related to a particular physical environment efficiently. Moreover, it enables us to supply this information to people who need it, at places they need it and at the exact time they need it – to anyone, anywhere and anytime. In this way, it eases off the threat of excess information and thus surpasses the current limits of information processing capabilities.

A system based on the proposed concept uses RFID (Radio Frequency Identification) tags to tag various entities in the environment with the purpose of providing people with contextual information. A mobile device is used to read data from these RFID tags and to retrieve meaningful information associated with them, which is presented to the user in an appropriate form (e.g., in spoken form for the visually impaired).

To demonstrate the real-world impact, which a system capable of advanced information processing and access can provide to society, we looked for something that could cause a small “revolution” within its field of application. We designed and developed a prototype of the SPOT-IT system, which uses the described concept to offer aid to the many visually impaired or completely blind people living in the world today. Alone in the U.S. there are 750,000 blind people while some 8.9 million people suffer from low vision [5]. The situation is similar in other developed countries and even worse in developing countries. Worldwide there are close to 161 million people suffering from significant visual disability, 37 million of whom are blind [9].

These people many a time have to rely on others to help them perform various daily tasks. This is mostly because blind or low sighted people lack information about their surroundings, which sighted people receive naturally by simply looking around (even unconsciously). The purpose of the SPOT-IT system is to provide blind and low sighted people with contextual information about their surroundings, thus enabling them to know (i.e., “to see”) what is around them. Such newly gained information will in turn enable them to be more independent and self-sufficient, since they will no more have to rely on others to provide them with this information or perform tasks based on this information.

Ideally, the ultimate objective of the SPOT-IT system would be to make visionless people completely independent, but this would hardly be possible unless one truly restored their sight and enabled them to see the same way as visually unimpaired people see. Thus, the prime objective of our design is to increase the

level of independence of blind and visually impaired people by going a step beyond the current boundaries of seeing aids.

SPOT-IT is only one example of the concept's overall potential, since it is applicable to a whole family of systems designed for a particular application domain. At the same time, the functionality of the SPOT-IT system is extensible to other applications also for sighted people thus providing new possibilities of information acquisition and processing resulting in a variety of socially valuable applications.

The rest of the paper is structured as follows. First, we give a short overview of the SPOT-IT system structure, provide an example scenario of the system usage and list key requirements considered during design and prototype implementation. In section 3 we describe the basic principles of the proposed concept. Section 4 presents the system architecture. We describe the design and implementation details of individual modules in consequent sections. In section 8 we briefly discuss marketability issues. The paper concludes with a summary.

2. OVERVIEW OF THE SPOT-IT SYSTEM

The ultimate purpose of the SPOT-IT system is to provide an alternative way of information acquisition for visually impaired people. To accomplish this goal the system utilizes different components (see Figure 1). RFID tags are used to tag various entities. During operation, the user carries a mobile device (e.g., a PDA or a Smartphone equipped with additional add-ons) with the corresponding application software. The input is gathered by an RFID reader, which searches for RFID tags near the user and forwards the acquired data to the mobile device for further processing. Depending on the actual data, the system retrieves additional information from RNS (RFID Name Service) and RTS (RFID Tag Service) servers over the Internet. When all data retrieval and processing is complete, the results are presented to the user by means of a synthesized voice, which is transmitted to headphones via a wireless personal area network or a wired connection.

Each RFID tag contains data, which uniquely identify a message that is presented to the user when a tag is detected nearby. If an RFID tag contains only a reference to the data, then the SPOT-IT system uses a distributed group of RNS and RTS servers to interpret the reference into meaningful information which is presented to the user. The conversion is a two-stage process, in which first the RTS server that contains the actual tag data is identified by means of the RNS web service. Thereafter the data is downloaded onto the mobile device for further processing from the respective RTS server.

The SPOT-IT system software is based on a client-server architecture. The client side comprises the mobile device software, whereas the server side comprises the RNS and RTS web services. Both the RNS and the RTS web services feature two different interfaces. The default web service interface allows for direct access from other applications (e.g., the mobile device), whereas an additional web-interface facilitates access to data associated with RFID tags and allows for its simple modification by RFID tag owners (e.g., changing the content of a message). System administrators can use the web-interface to manage system settings (e.g., adding a new RTS server).

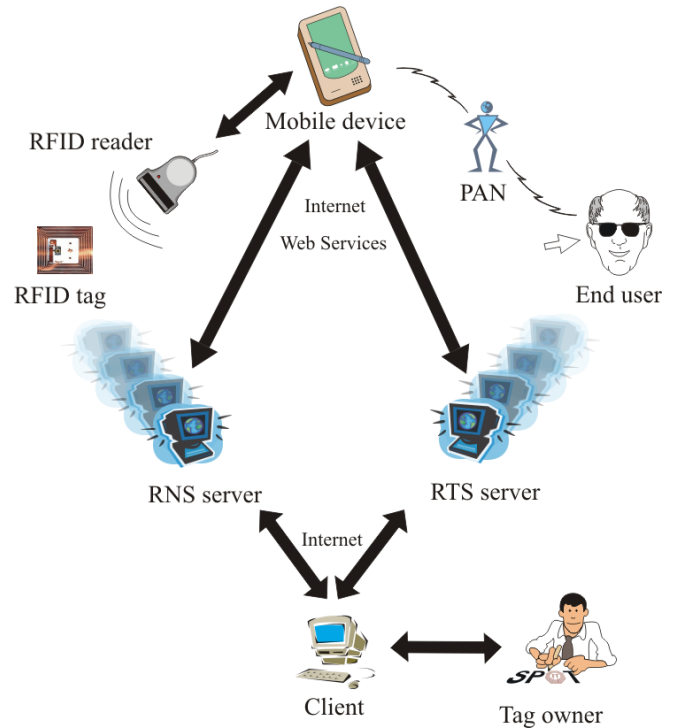


Figure 1. Overview of the SPOT-IT structure.

For the end user, the client side is essential, while the presence of the server side of the system remains mostly hidden. After all, it is the client software and the mobile device that accompanies him wherever he goes. As such, the client software's purpose is to find and identify RFID tags near the user, retrieve the associated messages and determine the message content to be presented to the user. The user has the option to affect message playback by means of an easy-to-use interface. This includes skipping, pausing or replaying the message, replaying it at a slower pace or even replaying it letter by letter.

2.1 Usage scenario

Rebecca is a blind woman accommodated in a hotel, which employs the SPOT-IT system. As a result, finding her own room poses no problem for her as compared to other hotels. As she walks by doors, she can use SPOT-IT, to tell her room numbers. In each room, there is a radio with an alarm clock on a table next to a bed. When Rebecca comes closer to the radio, SPOT-IT provides her with information on how to operate the radio and set the alarm. Next morning, when she comes close to a window in her room, SPOT-IT tells her about the weather outside – it is raining. Thereafter, she takes her umbrella and leaves for breakfast. When she leaves the elevator, she receives information about directions to important places in the hotel, which include the hotel restaurant.

John is a coordinator of the hotel's congress rooms. From his office, he updates announcements about conferences, which are due in the hotel. Therefore, when Rebecca walks near a congress room, she gets up-to-date information about the events inside.

The hotel itself is located near a busy road intersection, what makes it a dangerous place for sighted people, not to mention blind people who are really endangered. As Rebecca leaves the

hotel and comes to the proximity of the intersection, she receives a warning message from the system that she should pay extra attention to the situation around her. Local authorities had RFID tags with warning messages placed near dangerous places of the town, which helped to reduce the number of traffic accidents caused by visually impaired citizens.

Rebecca intends to stay in the city for a longer period. Over time, she becomes quite familiar with the hotel and its vicinity, thus she no longer needs to hear every information as she did the first day. Because she set a filter on some tags and decreased the priority of some message categories, SPOT-IT does not bother her with information she already knows. While staying in this hotel, Rebecca now often uses SPOT-IT in the on-demand mode. This way, she only receives additional information when she truly wants to.

Rebecca's acquaintance Michael is on vacation in a national park together with his friends. During a walk near their hotel they find a tourist information guide (board) with information about local tourist trails to places with beautiful views, which goes through several villages. Michael notices a SPOT-IT logo near the guide and since he has a mobile device capable of reading RFID tags with SPOT-IT software (e.g. cell phone), he tries to search for SPOT-IT tags. In a few moments he finds out, that he can download a text guide about the tourist trail with the description of nearby landmarks that can be seen along the way (e.g. what is the name of the town that can be seen from a hill on the way even with a small photo of the view). He also discovers that information about catering opportunities along the way is also available and some restaurants also have web pages where the daily menus can be "explored". He also downloads timetables of busses going back to the hotel from nearby villages in case they decided not to go back on foot.

2.2 Key Requirements

During the design stage of the SPOT-IT system, it was imperative to satisfy several key requirements, which had to be met to make the system useful:

- The system should be able to associate information with real-world entities.
- The system should offer an uncomplicated way of message management.
- The system should be capable of retrieving associated messages with low latency.
- It should be possible to use dynamic content of messages that could change depending on different conditions. The message format should be easily extensible.
- The client side of the system should be a light, small and convenient mobile device.
- The system should communicate with visually impaired people in an easily comprehensible and discreet way and should provide an easy-to-use user interface.
- The power requirements of the mobile device with add-ons should be low.
- The overall cost for the end user should be low.
- The system must be able to operate in densely populated urban areas, including city streets, high-rise buildings, households, topologically complex buildings like hotels, and shopping malls, as well as underground. Seamless

coexistence with other electronic and communication equipment is necessary.

- Deployment of the system should take advantage of existing infrastructure and avoid excessive costs.
- The overall system design and implementation should facilitate the incorporation of changes.
- The system's implementation should be platform independent.

3. BASIC CONCEPTS

3.1 RFID tag and the SPOT-IT data format

One fundamental function of the SPOT-IT system is to associate information with entities in a real world environment, which requires these entities to be identified. After evaluating the available technologies, we proposed RFID (Radio Frequency Identification) tags as a solution for the SPOT-IT system, since an RFID tag can store additional data, which can provide more than the sole identity of an object. Furthermore, it may not require a power supply and its range is ideal for providing contextual information related to entities located nearby. At present, RFID is widely used in commercial applications [7], but its purpose consists mainly of tracking goods, animals or even people and it is often envisioned as a replacement for barcodes (www.epcglobalinc.org/index.html).

Several kinds of RFID tags are presently available and differ in the radio frequency used for communication, in their form, in the amount of stored data and in the used power source. While active tags use their own batteries, passive tags use the energy supplied by the electromagnetic field emitted by an RFID reader and thus require no maintenance [8]. In the prototype we used passive tags, which comply with the ISO 15693 standard. These tags have 8 bytes of ROM memory with a unique code used to address the tag and can hold up to 256 bytes (divided into 4-byte blocks) of user data and thus offer ample space for data storage.

In addition to identifying an object with a unique code, we also use the chip's memory to store data about what kind of object it is and in some cases even to provide all necessary information without obtaining any further data from a server in the Internet.



Figure 2. SPOT-IT tag data format; colors indicate 4-byte memory blocks.

The SPOT-IT system needs at least 16 bytes (4 blocks) of the chip's memory (see Figure 2):

- *Version* – determines the version of the SPOT-IT tag data format.
- *Reserved* – reserved for future use.
- *Category* – a code identifying what kind of message is associated with the SPOT-IT tag (e.g., public transport, health care, navigation).
- *OwnerID* – a unique code for a tag owner that is used in RNS server queries.
- *TagID*, *OwnerID* – determine the message associated with the tag.

- *Checksum*, which also serves for identification of SPOT-IT tags, while ignoring all others.
- *Length* – specifies the size of additional data contained directly in the SPOT-IT tag.
- *Data* – if Length is greater than zero, this contains the additional data encoded in UTF-16 Unicode character encoding. If data are stored in the tag, they are presented to the user without querying an RTS server.

We assigned 6 bytes to the *OwnerID* and only 2 bytes to the *TagID*, because we assume that most tag owners will not need more than 65,536 SPOT-IT tags (i.e., private or small business users are unlikely to use that many tags). If some tag owners need to use more SPOT-IT tags, they can register additional *OwnerIDs*. The reason why we used a separate *TagID* instead of the tag's unique 8-byte code, is the fact that the SPOT-IT system must be able to work with several RFID tags which are associated with the same message and thus must have the same *OwnerID* and *TagID*.

Although we used only the SPOT-IT tags in the prototype application we see no critical problems that would prohibit the use of other RFID tags by the SPOT-IT system. Thus also data stored in “generic” RFID tags placed on various objects (e.g. groceries) could be processed and if their type and the corresponding information was identified also presented to users.

3.2 Message categories and their priorities

When considering a system working with data, which provides various information to its users, we also have to consider the possible consequences of providing too much information. Solving the issue of information overwhelming is even more important in case of a system for visually impaired people since the perception via the sense of hearing is much slower than perception via visual sense and cannot be interrupted easily. Even though blind users will speed up the speech synthesis process substantially since they devote most of their cognitive resources to hearing and none to visual processing, there are still concerns that must be accounted for. First, hearing is less “parallel” than sight in that only one source (or few sources) of information can be processed at a time. Furthermore, if information of little interest is presented to sighted users they learn to “overlook it” over time. Ideally a similar way to ignore irrelevant information must be found for speech in the case of blind users.

The SPOT-IT system offers several solutions of this problem, which take effect on different levels of information processing. As an essential aspect of the solution, the SPOT-IT system introduces an ontology of categories, which describe the meanings of messages. Categories are hierarchically organized so that each category can be divided into sub-categories ultimately creating a tree of categories, which is stored on a mobile device and retrieved from an RNS Server in the case of an update. On the server side, the ontology of categories is stored using Resource Description Framework (RDF) syntax [2], [3]. The client side uses the ontology to build its own tree of categories by means of an RDF parser thereby adding one important attribute to each category – its priority.

Every category represents a group of messages with similar kinds of information. Thus, if a user knows one message, he can guess at the content of other messages from the same category, and accordingly personalize the settings of SPOT-IT to better reflect

his preferences by increasing or decreasing the priority of a certain category of messages. Only messages with priority above a defined threshold are retrieved from an RTS server and presented to the user. If multiple messages with a high enough priority are encountered simultaneously, they are presented to the user in descending order of their priorities. To enable the user to affect the behavior of the system for specific tags, which are in fact the end nodes of the hierarchy of categories, SPOT-IT allows the user to define tag specific filtering rules.

Prior to designing the SPOT-IT system, we consulted the use cases of the system with visually impaired people. The detection of specific obstacles and/or hazards e.g. on the streets, which are hard to detect with a cane proved to be of special importance. Quite often such obstacles can (easily) be avoided if the visually impaired are aware of their presence, but can prove very dangerous if unnoticed. Typical examples of such hazards include uncovered manholes, or obstacles in head height. Therefore, we added a special category of messages to the SPOT-IT system – critical messages that have to be presented to the user as soon as possible, because they warn the user about imminent dangers. SPOT-IT tags containing critical messages can be placed in the proximity of construction sites, dangerous roads or in places where other visual hazard warning signs are used (e.g. high voltage). If a critical message is present near a user, it is immediately presented to allow for the longest reaction time possible. Additionally, to prevent the abuse of the system by adding random critical messages, only a central authority (see section 3.4) can create new critical messages.

3.3 Message content representation

In case of simple static information (e.g., numbers of hotel rooms), the SPOT-IT device acquires it directly from a SPOT-IT tag, in which case it is stored in plaintext format. If the nature of the information is more complex, it is retrieved from an RTS server. The latter approach imposes no limitation on the message length and brings many possibilities of message content management to SPOT-IT tag owners.

```
<?xml version="1.0"?>
<context xmlns="http://www.fiit.stuba.sk/spot-it/cidl">
  <spot xmlns="http://www.fiit.stuba.sk/spot-it/cidl">
    <title> Washington-Centerville Library </title>
    <interval text="1">
      <from> <h>10</h> <m>0</m> </from>
      <to> <h>21</h> <m>0</m> </to>
    </interval>
    <interval text="2">
      <from> <h>21</h> <m>0</m> </from>
      <to> <h>10</h> <m>0</m> </to>
    </interval>
    <text name="1">
      <t/> Open </text>
    <text name="2">
      <t/> Closed </text>
    </spot>

    <part2 xmlns="http://something.other">
      <!-- ... -->
    </part2>
  </context>
```

Figure 3. Example of a message represented in CIDL.

Figure 3 depicts a simplified example of a message stored on an RTS server. In order to represent extended contextual information we developed a specification of Contextual Information Description Language (CIDL). CIDL is a simple, yet powerful XML for the representation of additional information about real-world entities. The main idea is to encapsulate information represented by other XML-based languages into a single message associated with an entity and allow for its fast, single-pass processing. Each part is independent and is distinguished by a namespace (the *xmlns* attribute). This allows for the development of new devices intended to process additional information (represented by new namespaces) while not affecting existing devices. Each message has its own title that is represented by the title element. Within the content of the message, the *t* element can be used as a substitute for the message title so that it need not be repeated several times.

In order to enable a user to control the way messages are presented (e.g., requesting further information), messages can carry information with different levels of detail (hierarchically structured). Furthermore, CIDL can express the following additional information about individual message parts:

- Periods of time during which the enclosed information is valid (e.g., a notice “library is open” could be valid between 9:00 AM and 6:00 PM). They are represented by means of interval elements with the text attribute that points to the message part text (text element).
- ID of a user, which allows for user specific information (non-confidential since there is no authentication). It is represented by the *userid* attribute in the text element.
- The priority of the information to allow the tag owner to control the amount of presented information according to user preferences. A message content part is only presented when its priority is not lower than the user assigned priority of the respective message category, which can be further increased or decreased by the user. Furthermore, priorities define the reading order of message parts on equal levels of detail.

3.4 RFID Name Service and RFID Tag Service

To maintain the flexibility of the complex system, we employed a service-oriented architecture (SOA) [1]. SOA achieves flexibility by decomposing the whole system into services and application front-ends and by decoupling them from a particular technology [4]. We applied the concept of web services to the server side of the SPOT-IT system. The introduction of web services makes it possible to separate the interface from the functionality of both the RFID Name Service (RNS) and RFID Tag Service (RTS), what gives users the possibility to access both web services with their own custom made clients thus allowing seamless integration with other systems.

The RTS server is designed to provide CIDL encoded messages associated with a SPOT-IT tag, which are identified by an *OwnerID* and a *TagID*. In order to reduce the required Internet connection bandwidth, a list of usable CIDL namespaces can be included in each request. The RTS server then only returns appropriate sections, which use these namespaces.

RNS allows clients to determine which RTS server stores messages associated with SPOT-IT tags. The service requires the *OwnerID* from a SPOT-IT tag and returns the URL address of the desired RTS server, where the information associated with the SPOT-IT tag is stored.

In the SPOT-IT system, both RNS and RTS are deployed on servers in the Internet and are accessed by several types of users. All users utilize the RNS to resolve addresses of RTS servers, which provide end users with information associated with SPOT-IT tags and allow tag owners to manage this information. Administrators can access the RNS and RTS to manage owner accounts (*OwnerIDs*) and RTS server addresses. While applications access the RNS and RTS directly by exploiting the methods offered by both web services, human users can access the RNS and RTS via a simple web interface.

SPOT-IT can be used with multiple RNS servers, because each server has complete information about all RTS server addresses. Mirroring of RNS servers prevents overloading of individual servers and increases overall reliability. All RNS servers are administered by a central authority, which also issues *OwnerIDs* to new users. Having a single authority supervising the content of all RNS servers ensures that the SPOT-IT system’s network of RTS servers is never broken into several isolated parts. The authority maintains a list of existing critical messages and provides a web service directory where each RNS mirror is registered. The directory can be queried for RNS server mirrors using a standardized format.

4. SPOT-IT ARCHITECTURE

The SPOT-IT system uses a client-server architecture (see Figure 4). The client side of the system is illustrated on the left and shows the successive application layers in which data processing is performed during operation. Data are collected by a *TagProvider* module, which communicates with SPOT-IT tags. Low-level communication is performed by the *RFID reader API*, while the *SPOT-IT API* provides a high-level interface to SPOT-IT tags. The resulting data are processed by the *TagProcessor*, which determines the order of further processing for each encountered SPOT-IT tag. Tag processing has two phases, where first simple filtering rules are applied by a *TagFilter* and more sophisticated category based processing is performed by a *CategoryBasedFilter*.

For every processed tag with a high enough priority, the associated data is retrieved by the *MessageProvider*. To prevent frequent and eventually unnecessary Internet communication, messages are stored in a cache. If a message is not present in the cache when necessary, the *MessageProvider* retrieves it from an RTS server and stores it in the cache. When the URL address of the respective RTS server is unknown, the *MessageProvider* obtains it from an RNS server. Once retrieved, the message is processed by the *MessageProcessor*, which determines the message parts to be presented to the user. The resulting text is presented to the user by the *MessagePresenter*. A speech synthesis engine converts it into speech, which is played by the *SoundLib* module as a regular waveform and transferred to headphones by the *PAN* module.

The server side of the SPOT-IT system consists of the RNS and RTS, both of which respond to client queries and store their data

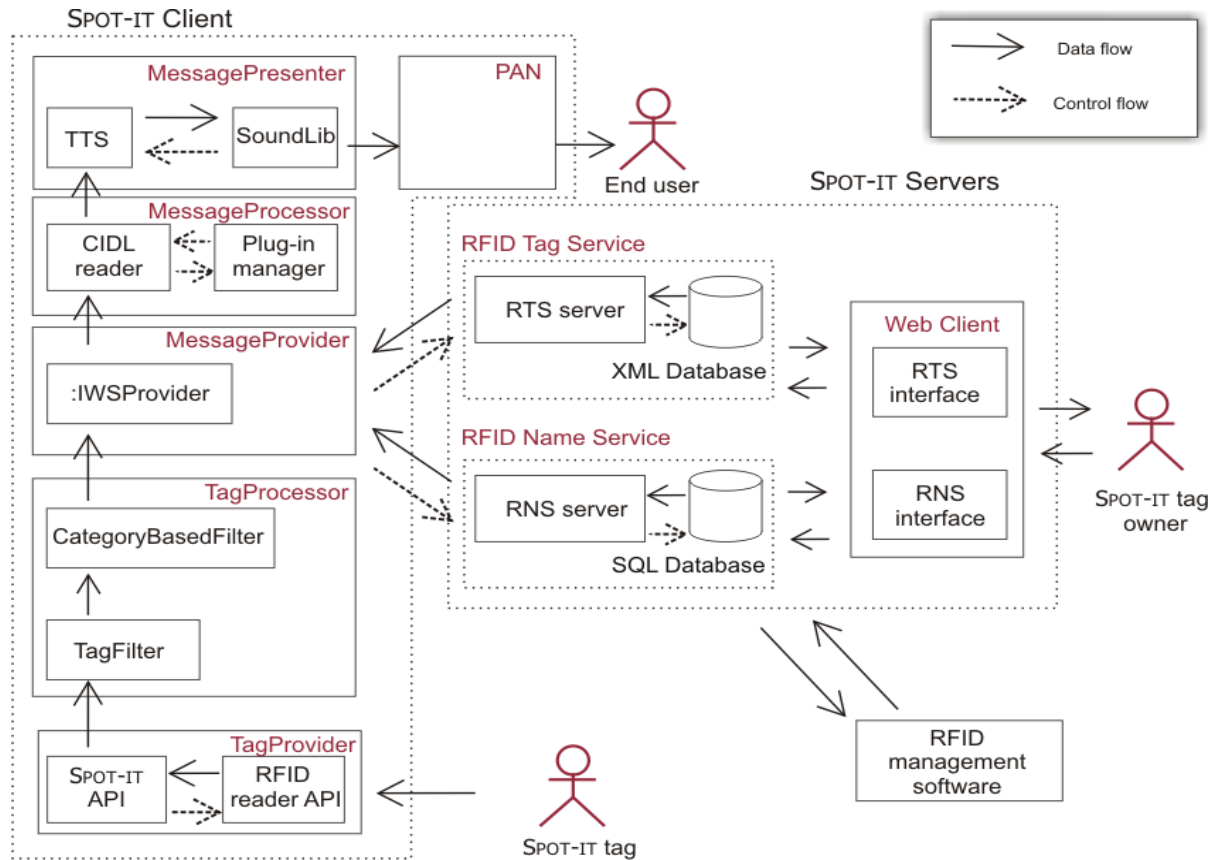


Figure 4. SPOT-IT system architecture.

in an SQL and XML database respectively. RNS and RTS are primarily accessed by SPOT-IT client devices and by web clients, which provide access to these services for tag owners. Additionally, they can be accessed by external systems, which perform automated management of SPOT-IT tag data on behalf of tag owners.

The SPOT-IT system prototype follows the described architecture and implements all software and hardware modules as illustrated in Figure 4. We describe the design and implementation details of individual modules in sections 5 (SPOT-IT client), 6 (SPOT-IT servers) and 7 (SPOT-IT hardware).

5. SPOT-IT CLIENT SOFTWARE

5.1 Client application and user interface

The client application provides the end user with the necessary functionality by realizing the individual layers of processing as defined by the system architecture. It performs all tasks related to the operation of the SPOT-IT system by utilizing the functionality of individual modules based on user settings and handles user input. The SPOT-IT client operates through three relatively independent parts – the searching for SPOT-IT tags, their processing and presenting. These are realized by three concurrently running threads, which communicate primarily via events.

During normal operation, the *TagProvider* periodically searches for SPOT-IT tags (each 250 ms). If a new tag is found or an old

tag gets out of range, it sends appropriate events to the processing thread, which consists of the *TagProcessor*, the *MessageProvider* and the *MessageProcessor*. This thread handles user input, processes new tags, retrieves and processes the associated messages and initiates their presentation, which can be affected by the user (e.g., repeating, skipping a message). Finally, message playback is asynchronously performed by the *MessagePresenter*.

Since the message presentation is an adaptive process, the decision whether to present a message or not is made according to its assigned priority and the number of messages, which are to be presented to the user. Messages with priorities lower than the priority of the highest priority message minus a user defined value are not presented. For example, in the presence of eight tags, the system reads only five of them, because the other three have low priorities. The system notifies the user about various events by means of event notifications (specific sounds), which are raised for example when a new tag is discovered.

The user has the possibility to personalize the settings of the system to better suit his needs. The order in which messages are presented can be affected by setting message priority related options, while event notifications can be enabled, disabled or the associated sounds can be changed. The user also has the option to select one of the following modes of operation:

- *Normal* – the system presents all messages associated with SPOT-IT tags nearby. Messages are processed and ordered in descending order of their priorities, thus preventing the user from being overwhelmed by information. The user is

notified about various events (e.g., a missed tag) by specific sounds (e.g., a specific beep).

- *Silent* – the system presents only critical messages.
- *On-demand* – the system ignores all tags. The user has the option to initiate a one-time search for nearby SPOT-IT tags.

We discussed the needs and requirements of potential SPOT-IT users with visually impaired people. Based on the feedback we received, we designed the interface of the SPOT-IT client with a special screen specifically for their needs (see Figure 5). The standard interface is graphical, capable of displaying messages in textual form and suitable for performing system wide settings. The interface for blind people is much simpler and utilizes the touch screen as four additional soft-buttons. To enable the user to recall interesting information later, SPOT-IT offers the option to store a message and listen to it later. Navigation in stored messages is easy thanks to the categorization of messages and their identification by their titles.



Figure 5. SPOT-IT user interface for the blind (left) and for the sighted (right).

5.2 Tag provider

The RFID API implements the functionality of the *TagProvider* and serves as a communication interface for the RFID reader. It is designed as a set of two separate DLL libraries, so it can be easily replaced or upgraded. The first one is the RFID reader API that implements functions related to the RFID reader, which communicates with the mobile device via a serial port. For each command it is necessary to generate and compare an Longitudinal redundancy check (LRC) to ensure error free communication. If we wanted to use a different RFID reader only this DLL library would have to be changed.

Thanks to the aforementioned module, the SPOT-IT API is independent from the used RFID reader and provides functionality, which is related to SPOT-IT tags. It continuously scans for SPOT-IT tags and sends notification events to the Client application if new tags are found or if old tags get out of reading

range. Additionally, it also provides functionality required for writing data into RFID tags.

5.3 Tag processing

The *TagProcessor* receives an array of SPOT-IT tags as input. After processing, it returns an array of these tags ordered in descending order of their priorities. In the first stage of processing a *TagFilter* checks the tags against a set of filtering rules and applies all matching rules. These are stored in a hash table and can be defined for individual SPOT-IT tags identified by the *TagID* and *OwnerID*. The user can create a filtering rule for the currently presented tag or modify existing filtering rules, which are distinguished by the title of the filtered message. Filtering rules can be of various types:

- filter always,
- filter for a certain period of time or for a number of passes around the SPOT-IT tag,
- filter messages from a certain owner, based on the *OwnerID*,
- filter until the message changes.

The *CategoryBasedFilter* assigns the priorities of corresponding message categories to all tags and sorts them in descending order of these priorities. The categories with their respective priorities are internally stored using an abstract data type – a tree. An ontology of categories is used to change the behavior of the system related to the groups of messages. By changing the priorities of categories, the user affects the order in which multiple tags in the range of the reader are presented. When the priority of a certain category becomes lower than a defined threshold, the messages belonging to that category are no longer presented to the user.

5.4 Message provider

After initial processing of SPOT-IT tag data, the associated messages are retrieved from an RTS server in the message providing layer. The client application retrieves the messages by exploiting the *IMessageProvider* interface, which is implemented by the *MessageProvider* module. In order to decrease the latency of queries, the *MessageProvider* module employs several caching mechanisms. It maintains a hash table of *OwnerIDs* along with the matching RTS URL addresses and also stores messages downloaded in the past, thus being able to provide them later without downloading them again.

The validity of each message is limited by a deadline, which is determined by the RTS server at the time of download. It is based on the date of its creation and the *UpdateFrequency* attribute set by the tag owner. Every time the *MessageProvider* is queried for a message that is present in cache, it checks its validity and downloads it if necessary.

During message retrieval first the URL address of the respective RTS server is determined from the table, or if it is not present in the table, it is retrieved via RNS. The RTS server is queried for the message, which is downloaded, stored in the cache and returned to the client application. If all RNS servers are offline or the RTS server is unavailable, an error is returned.

The handling of critical messages is different, because they need to be presented with low latency. A critical message, once retrieved from a RNS server is stored permanently in the cache,

thus preventing the *MessageProvider* from querying the RTS for these messages. This allows SPOT-IT to present the critical message with almost no delay. Critical messages never completely fill the cache, since there is only a limited number of critical messages and their average size is relatively small.

5.5 Message processing

The *MessageProcessor* performs the task of message content processing in the SPOT-IT system. Its purpose is to process the CIDL representation of a message retrieved from an RTS server into hierarchically structured plaintext information, which can be presented to the user.

The *MessageProcessor* consists of two parts – a CIDL reader and a Plug-in manager. The CIDL reader parses the message's content using the Pull XML Parser provided by the .NET framework. It is a state machine, which by default traverses the message's hierarchical tree structure by means of a breadth-first traversal algorithm. The user can affect the traversal by skipping certain parts of the message or by instructing the CIDL reader to provide detailed information about the current part of the message by processing the next layer of the message's tree.

When the CIDL reader starts processing a message, it converts the first layer of a message's tree into corresponding lists of *Interval* and *MessagePart* objects. The Interval objects describe the periods of validity for one or more *MessagePart* objects, which encapsulate the priority, the user's id and the text of the message. The result of the traversal is the text property of the *MessagePart* object, which is presented to the user,

- if its priority is not lower than the priority of the message's category,
- if the userid in the MessagePart is equal to the user's id (if present) and
- if the associated Interval object corresponds to the current date and time.

In order to enable other developers to extend the capabilities of the SPOT-IT system, we incorporated the support for additional plug-ins, which contain a parser used to process specific parts of a CIDL encoded message. Each time a new CIDL namespace is introduced, developers can create new plug-ins (distributed as DLL libraries) to take advantage of its new features. Prior to use, all plug-ins must be registered with the Plug-in manager.

5.6 Message presentation

The purpose of the *MessagePresenter* is to convert processed textual data into a form, which a visually impaired person can understand – into speech, and present it to the user. The synthesis of speech is performed by a free text-to-speech (TTS) engine called Flite provided by the Carnegie Mellon University (www.speech.cs.cmu.edu/flite/index.html). For the SPOT-IT system, we used this engine with minor modifications, which were mostly related to its encapsulation into a dynamic link library (DLL) for Windows Mobile and the use of this library from within the main SPOT-IT application written in C#.

The *SoundLib* module plays the resulting audio data using the PAN via the human body, but is naturally capable to use the standard audio output for analog headphones if necessary. The *MessagePresenter* extends the basic functionality of the Flite TTS

engine by implementing playback oriented features in the *SoundLib* module, which is able to handle several sounds simultaneously. New sounds can be played, existing ones can be paused or stopped. For this, the SoundLib module maintains an internal array of currently playing sounds and their attributes. Based on feedback from visually impaired users who find it difficult to understand synthesized speech, we added support for playing the text one word or one letter at a time in addition to the standard mode which plays the text continuously. Finally, to ensure smooth sound playback a separate thread with high priority is used to play the sounds (i.e., update sounds buffers).

6. SPOT-IT SERVER SOFTWARE

6.1 Web Services

To fulfill the key requirements of a service oriented architecture as presented in section 2.2, we used the .NET Framework to implement the RNS and RTS [6]. One feature of web services is the use of XML and standard protocols. We chose HTTP to be the transport protocol so that it can be easily secured by using Secure Socket Layer (SSL). As an "operating environment" for web services, we used the Microsoft IIS server with ASP.NET.

Both RNS and RTS store data on database servers. These data are accessed by several web service methods, which constitute the interface of RNS and RTS. Each method performs the authentication and authorization process, which compares the supplied username and password hash against the data in the database. Once this process is complete, the web service method accesses the database by calling the appropriate stored procedure on the database server and processes the resulting data. Finally, the processed data are sent to the web service client. This process is the same for both RNS and RTS, with the only exception being the lack of stored procedures for the RTS, which are not supported by the used XML database (see section 6.2).

6.2 Database Servers

The SPOT-IT system uses two types of database servers. While RNS uses a standard relational database, we opted for a native XML database for RTS, which is fully extensible and allows us to make use of all advantages of CIDL and XML.

The SQL server of the RNS uses stored procedures to maximize the processing speed. This solution offers superior performance, because stored procedures are compiled and stored in executable form on the database server itself. For each RNS method there is an appropriate stored procedure on the database server.

The XML server of the RTS stores messages in CIDL representation. We used XMS a native XML database by Xpiori (http://www.xpiori.com/developers/dotnet_developers.html), which allows to process message content directly on the database server. Thus, it is not needed to implement a CIDL parser on the RTS server in order to extract message parts containing only the namespaces specified in the RTS query. Our prototype utilizes XPath and XQuery languages to extract and manipulate XML data.

6.3 Web Client

The use of web services on the server side implies that there can be many different client implementations, which enable users to

access data stored on RTS and RNS servers. To provide tag owners with a means of direct access to RNS and RTS without the need to install additional software, we have designed and developed a web client for each service. Both web clients are implemented in ASP.NET 2.0 and can connect to any RNS/RTS server on the Internet with either administrator or tag owner privileges.

For the implementation we chose ASP.NET, because both RTS/RNS web services are implemented using the .NET framework. This allowed us to use the same components for handling data and did not require the implementation of additional functions for parsing data.

7. SPOT-IT HARDWARE COMPONENTS

7.1 Mobile device

The most important aspect of every system is how well the end user can make use of its functionality. In the case of the SPOT-IT system, this is directly related to the mobile device, which the user carries with him and which provides him with all the benefits of the system. For the prototype, we decided to use a PDA with phone capabilities manufactured by HTC – the XDA II. It is a PDA with a built-in tri-band (GSM 900/1800/1900) mobile phone and can thus be used in different wireless networks.

We chose this device out of many because it offered sufficient processing power with enough memory while still having a built-in GPRS modem for Internet connectivity. The support for the .NET Compact Framework allowed us to use the same programming environment to develop all components of the system and make them cooperate easily.

Since the SPOT-IT system is built around RFID technology and thus includes two essential components – an RFID reader and at least one RFID tag. Even though we assume, that an RFID reader will be integrated into mobile devices in the future, we had to use a separate reader for our prototype the S-4100 reader included in an evaluation kit from Texas Instruments (<http://www.ti.com/rfid/docs/products/evalKits/RF-MFR-RNLK-00.shtml>). The main advantage of the reader is that it can read multiple kinds of RFID tags, including low frequency as well as a number of different high frequency tags.

The SPOT-IT system however is not limited to the hardware nor software that we used in the prototype. It is also possible to write custom client applications for SPOT-IT in other languages such as Java, which is widespread on a range of available mobile devices.

7.2 Personal area network via human body transmission

The last stage of message presentation involves its transmission from the mobile device to headphones for which we use the human body as a physical layer. We believe that this kind of transmission can be especially useful when used to interconnect small devices like PDAs, headphones or watches, because it has low power requirements, causes little interference and is hard to intercept. We constructed a prototype, consisting of a transmitter and a receiver, based on electrostatic coupling [10], which transmits the synthesized voice from the mobile device to headphones through the human body.

8. MARKETABILITY

We consulted the design of the SPOT-IT system with blind people, to ensure its feasibility and we incorporated several additional features, based on their advice. This made us believe that it would have a significant impact on their everyday lives, thus finally going beyond the boundaries of their disability.

The SPOT-IT system takes the widespread use of RFID tags into consideration. This poses an important requirement on their price. Presently RFID tags cost only approximately \$0.50 each and this price is expected to fall down to \$0.05 in the near future. This low price means that there is no real limitation on use of RFID tags for SPOT-IT system. For the end user of the SPOT-IT system, the mobile device amounts to the majority of expenses. With the advent of today's low-cost PDAs and Smartphones this is no more a serious issue. Presently, RFID readers are already becoming integrated into mobile phones (<http://www.nokia.com/nokia/0,,55738,00.html>) and we also expect PAN via the human body to be integrated into mobile devices in the future.

Infrastructure-wise, the RFID Name Service (RNS) would ideally be maintained by some international authority just like DNS servers are. An unlimited number of private and public RTS servers could exist and the RFID Tag Service could be offered as a hosted service similar to web-hosting services for "ordinary" tag owners. For a simple tag owner who only wants to use a couple of tags, the whole complexity of the SPOT-IT system can be hidden. For this, some kind of tag dispenser machines can be used, where users input messages (structured and/or dynamic thanks to CIDL) and receive appropriate SPOT-IT tags (simple RFID encoding machines are already available, e.g., Intellitag® PM4i by Intermec). The machine would take care of registering the tags with an RTS server belonging to an RTS hosting company, leaving the user with a ready-to-be-placed SPOT-IT tag.

We believe that many users would appreciate the availability of contextual information about objects in their proximity. Specifically visually impaired users would have the opportunity to become more independent by using the SPOT-IT system. We identified several examples of organizations and businesses that are likely to use SPOT-IT tags to spread information. Associations of visually impaired people could cooperate with public authorities to spread SPOT-IT tags at the important places of towns (e.g. offices, hospitals, or near hazards). For businesses, the use of SPOT-IT tags could become an issue of prestige while simplifying the navigation of the visually impaired on their premises (e.g., hotels, restaurants or conventions). SPOT-IT however is not limited to visually impaired people and we believe that businesses would also exploit the concept for sighted people. One example are shopping sessions – potential customers could have direct access to information about products such as websites without the need to manually search for them. This could be accomplished by additionally storing links to such pages in SPOT-IT tags and opening them in a (built-in) web browser on a mobile device.

Based on the low acquisition costs, easy use and many benefits of the SPOT-IT system, we estimate its widespread use by end users as well as tag owners.

9. SUMMARY

RFID is a revolutionary technology, which is very likely going to change our lives dramatically in the near future. In this paper we show that this technology need not be seen only as a tracking technology, but can be also useful for the society in different ways. We take advantage of RFID technology to associate information with real-world entities and utilize this association to provide the user with relevant information anytime, anywhere and in an appropriate form. SPOT-IT in particular does this for visually impaired people thus allowing them to go beyond the boundaries of vision loss by providing them with an additional "sense", which allows them to "see" their surroundings.

During the development of the SPOT-IT system, we have constantly discovered new aspects of the SPOT-IT system and came up with ideas of many additional extensions. Further work might include the incorporation of multiple languages. This could be done on different levels – on the client device by adding additional speech synthesis engines and on the server side by being able to provide information in the requested language. Another interesting extension also for sighted people, which could be primarily used in commercial applications, would be the incorporation of individual user and group accounts on RTS servers. This would improve the personalization capabilities of the SPOT-IT system.

The deployment of the SPOT-IT system has the potential to significantly influence the information providing capabilities presently available. The described system brings several major benefits to society and innovations:

- A novel way of associating information with real-world entities by exploiting the capabilities offered by RFID tags with the purpose of providing people with contextual information. The described concept presents a significant design difference compared to the common use of RFID tags for tracking goods, animals or people.
- Taking advantage of the ability to associate information with real-world entities, we devised a unique method of discovering and processing information, which is to a great extent independent from the user. As a result it provides an additional sense for a (visually impaired) person and thus endows him with an alternative way to detect (i.e., "to see") the entities and the associated information.
- A way of communication with the visually impaired by means of a fully featured text-to-speech engine, while using only the limited computing resources of a mobile device.
- An original approach to wireless data transfer in a personal area network (PAN) via data transmission through the human body, which requires little power and ensures privacy.

The SPOT-IT system has additional distinctive features. The most prominent one is its open architecture, which allows for reusability and easy extensibility of the system. To achieve platform independence and portability, we used the Microsoft

.NET platform, which together with its primary programming language C# makes developing applications and extensions easy, fast and more reliable. The service-oriented architecture based implementation of RNS and RTS separates the interface and functionality and provides services to a broad audience of users, while allowing anyone to implement their own client. Furthermore, the SPOT-IT system supports a flexible message representation, which makes creating dynamic messages possible and allows RFID tag owners to extend the base message format. The messages themselves are stored in a native XML database what makes working with messages and incorporating changes relatively straightforward.

10. ACKNOWLEDGMENTS

This work was partially supported by Science and Technology Assistance Agency under the contract No. APVT-20-007104. We would also like to thank Microsoft Slovakia for providing us with the RFID reader evaluation kit and Orange Slovakia for providing us with the Orange SPV M1000 GPRS PDA.

11. REFERENCES

- [1] Barry, D.K.: *Web Services and Service-Oriented Architecture*. Morgan Kaufmann Publishers, San Francisco 2003.
- [2] Decker, S., Mitra, P., Melnik, S.: *Framework for the Semantic Web: An RDF Tutorial*. In *IEEE Internet Computing*. Vol. 4, No. 6 (2000) 68-73.
- [3] Ebenhoch, P.: *Legal Knowledge Representation using the Resource Description Framework (RDF)*. In *Proceedings of 12th International Workshop on Database and Expert Systems Applications*.
- [4] Krafzig, D., Banke, K., Slama, D.: *Enterprise SOA: Service-Oriented Architecture Best Practices*. Prentice Hall, 2004.
- [5] Microsoft® Encarta® Reference Library 2005. © 1993-2004 Microsoft Corporation. All rights reserved.
- [6] Prosize, J.: *Programming Microsoft® .NET*. Microsoft Press, Redmond 2002. ISBN 0-7356-1376-1.
- [7] RFID Journal, Case Studies. <http://www.rfidjournal.com/article/archive/4> (2006-01-12)
- [8] Stanford, V.: *Pervasive Computing Goes the Last Hundred Feet with RFID Systems*. In *IEEE Pervasive Computing*. Vol. 2, No. 2 (2003) 9-14.
- [9] WHO: *Magnitude and causes of visual impairment*. Fact Sheet N°282, November 2004. <http://www.who.int/mediacentre/factsheets/fs282/en/> (2006-01-12)
- [10] Zimmerman, T.G.: *Personal Area Networks (PAN): Near-Field Intra-Body Communication*. MIT, 1995.