IEEE IoT
IoT Scenarios & Use Cases: See What I See

Description

When an object (person, car, truck or other mobile object) is moving in a specific area (e.g., an unsecure area, a city, or other locale), See What I See (SWIS) can provide monitoring by utilizing the available resources along its route. Resources delivering some level of tracking could be simple (e.g., proximity objects, RFID tags or sensors that simply mark a moving object as it passes by) or more complex, such as cameras that can record the object passing by or even be movable and able to record an object for a short period of time. If the route is known in advance, resources could be ready to track an object without needing to guess where it might be moving next. However, for particularly casual routes, a more dynamic allocation of resources could be provided based on predicting the possible movements and pre-allocation of resources.

In general, SWIS offers a set of tracking functions, where followers can view a moving object's path and can also recommend or command the object to alter course. In addition, the tracking function can show the object's predicted route and followers can, once again, suggest alternative routes. In case a monitored object is lost, the last identified position is recorded and an event is generated, such as an alarm in a security related service offering. The event contains all relevant information available about a moving object up until its last monitored position.

For individuals, SWIS can be extended by means of communication. For example, a user being monitored can communicate with followers by means of messaging ("I'm here in Place de la Concorde, can you spot me?"), or they can exchange multimedia content. The service can also support the exchange of streaming video, either generated by the user or by the cameras tracking the user movements). Content providers offering information and multimedia content related to the path a user is following can further extend services for tourists. Privacy constraints are initiated by users who can enable or disable tracking, while followers can request users to activate tracking functions.

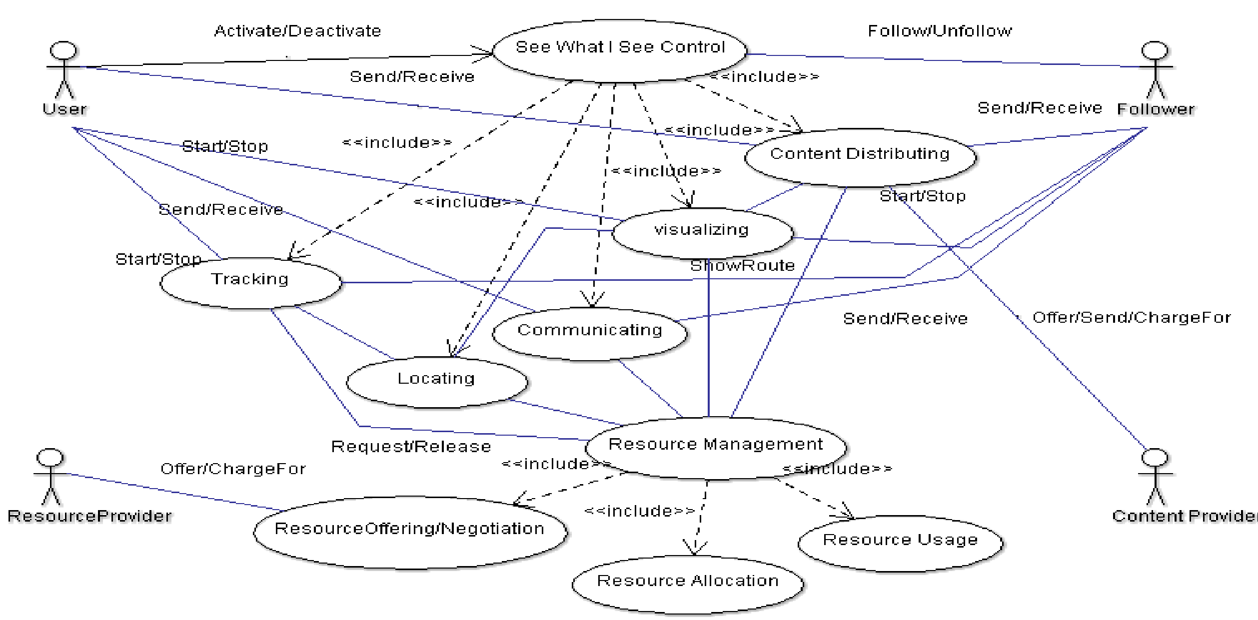
Of particular importance is the Resource Provider. This is an entity, such as a user, a company, an organization, or a smart city, that is able leverage local resources needed to track and visualize moving objects. Unlike the Content Provider, whose focus is information based, the Resource Provider enables resources useful to service execution, from cameras to proximity sensors.

Applicability Areas: Entertainment, Business, Security

SWIS service falls into two Categories: Smart City and Smart Business. On one hand, it is a service that can be used for entertainment, such as in a tourism application where a visiting user to a city can share what what's viewed with his followers. In another scenario, it could support a business application, as in a case where a company wants to follow the delivery of its goods by carriers in real time. It can also be used as a sort of security application when an object is moving in an unsecure area so that the resources available, such as cameras and sensors, track it.

The Participants

- The **User** is the entity that is moving and interacting in the space that is to be tracked and monitored. Information about its movement is processed in order to keep track of the actual path, and in order to forecast its possible future path.
- The **Follower** is the remote identity in respect to the area in which the user is moving. It tracks the user object and manages the information received. The Follower can also interact with the user by providing content about the places the user object is moving through and can also exchange messages with the user. Messaging is extremely important for SWIS, because it can be used for entertainment purposes by User and Followers, as well as be used to provide commands and events related to security issues. For example, messaging is used to deliver a command to avoid a specific road or the order to take a specific road, and similar scenarios.
- The **Resource Provider** is charged with offering local resources, such as cameras, local networks, local sensors networks, location resources and the like. Such resources could be pre-allocated by the SWIS service in order to provide a smooth tracking of user objects. Resources must be offered and negotiated before being allocated and integrated into the service infrastructure. The usage of resources could be paid for according to a negotiated agreement or, if many resources from several resources providers are available, bidding could be made. One of the goals of SWIS is to optimize the allocation of resources.
- The **Content Provider** enriches users' experiences by providing tailored content of interest for the user object. In the case of a SWIS service for tourists, the Content Provider could offer information on particular spots within a visited area and suggestions on things to see or to do. In the case of a SWIS service aimed at object tracking and security, information could relate to area to be avoided, traffic information, or city restrictions.



.

Figure 1: The SWIS Service

In Figure 1 a set of basic high-level functionalities that are needed to provide SWIS services are represented. Users and Followers interact with the service logic of the service in order to activate and deactivate it. From the User perspective, the Activate function could be interpreted as an invitation sent to one or more followers to start using the service. From the Follower perspective, the interaction with the service logic is used to initiate tracking and begin interactions with the User. By means of the Communicating Functionality, Users can send events to Followers (e.g., events related to the path or more complex ones), while Followers can send commands or suggestions to follow a specific path ("I've been there before and there is a nice restaurant around the corner").

The Visualizing functionality is used to show Followers what the User is seeing, either through the User's smartphone camera or via local cameras. In this case, the tracking function is not only useful for monitoring the movements of the User, but also for performing image recognition on the video streams of local available cameras. The Content Distributing functions how Content Providers can make relevant local information available to the User. These functions cooperate with the Location function in order to determine the actual position and direction of Users, and this information is then used to dispatch location related content to Users and Followers.

Tracking, Location, Communicating, Visualizing and Content Distributing functions refer to resources that are searched for, negotiated, allocated and integrated in a dynamic manner within the SWIS framework. The different local Resource Providers can offer dynamic resources and the infrastructure could negotiate them and allocate them based upon the perceived needs (and policies based on game theory) of the User.

SWIS Service Business Model

The basic offering of SWIS service is a set of intelligent tracking functions and the possibility to continuously communicate with moving objects. It allows for real time tracking of monitored entities' paths in order to anticipate, predict and influence future movements. In addition, the service can be extended beyond the set of basic functions with features oriented towards the entertainment.

The SWIS service value chain is a simple one. The SWIS service provider can offer a basic infrastructure plus some software for installation in simple communicating devices, such as M2M type device to more complex smartphones. The infrastructure can be enriched by local resource providers to extend the reach of the service by allowing for dynamic allocation of the available service infrastructure. On top of this dynamic infrastructure, the content providers can offer a wealth of local related information. Finally, the customer can choose between basic or extended functions and between spot or continuous local information. A tentative Value Chain is shown in Figure 2.

The revenue models behind the service are different according to the specialization of SWIS (e.g., tourism or logistics, or security). However, the costs for infrastructure could be shared among the different providers supporting the various SWIS implementations. In effect, the possibility to dynamically enrich the service by integrating local resources gives more flexibility to the investments in infrastructure (i.e., by creating a sort of community that shares resources **à** la FON).

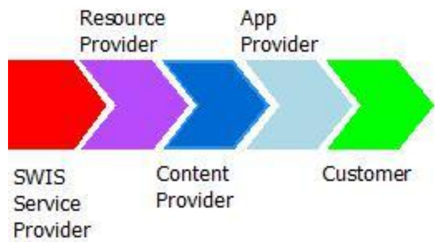


Figure 2: the Value Chain of the SWIS Service

IoT Architecture Requirements

At an object level, the SWIS service can be very complex because it aggregates many different functions. Figure 3 is an attempt to represent a (quasi) object model of the SWIS service. There is an evident mapping between the functions seen in Figure 1 and the objects represented in Figure 3.

At the object model level, the framework and the system functions are more clearly emphasized. For instance, the service logic is the object that has to coordinate the actions and the functions of Composite Virtual Objects (i.e., the virtualization and functional integration of real world objects) and System Objects like the session controller.

The Composite Virtual Objects collect functions and behaviors of simpler objects and they abstract the complexity of physical resources (see bottom of diagram). For instance, the set of objects needed to track the path of the User are quite explicative in this organizational layout. Sensors give low level detail and information on the current location of the User. This information can then be integrated in a virtual object that gives the actual location of the user and adds new information (e.g., speed and predicted direction).

As an example, cameras and an image recognizer work together (physical and virtual objects) in order to recognize in a stream of images a particular user that's being monitored. When this information is determined and the user is identified, the object location and information related to the user is used to determine the user's position, as well as their direction of motion. The aggregation of information and functions is a Composite Virtual Object called Actual Route. This object cooperates with others in order to get all the relevant real-time information on the movement of the monitored object. The Route Object is a Composite Virtual Object that abstracts and aggregates three different type of information about the motion, the actual route, the target route (i.e., the route that the object should have followed) and the predicted route (i.e., the most likely route that the object will follow). The Route object offers its methods and functionalities to the Service Logic that can utilize all the relevant information immediately and as needed.

Another example of objects that is relevant for IoT architectures are those related to the coordination of other objects. The Service Logic and the Session Control fall in this category. The Session Control is particularly interesting, because it is the object in charge for coordinating and keeping track of all the resources allocated to the service execution. In fact, the route and the other composite virtual objects are seen as resources and should also be governed and orchestrated by the session controller.

The Session Control should cooperate with the Service Logic in order to determine what local resources should be allocated. Generally speaking, resources should be intelligently allocated according to the dynamic behavior of the moving object and according to policies that try to optimize the usage of objects. Resources allocated from external providers could be more expensive or less reliable, so criteria should be followed to minimize costs and risks. On the other hand, a short allocation of a scarce resource can be very expensive, whereas the cost savings from a longer allocation could be justifiable in some cases.

In order to be have an intelligent allocation of both internal and external resources, two system objects are presented in Figure 3: the Negotiator and the Allocator. The Negotiator is intended to negotiate with the Resource Provider, perhaps through another external domain Negotiator, for the best possible deal, weighing criteria such as the best resource at the lowest price and for a longer duration. The Allocator allocates, integrates and releases the resources as fast as possible in order to keep pace in a highly fluid demand environment for resources. These two objects represent a set of cognitive functionalities that a cognitive IoT architecture should take into account. Moreover, the triumvirate (Session Controller, Negotiator and Allocator) could be tasked to work cohesively to launch bids between Providers when there are plenty of resources and the Providers are in competition to sell their objects.

The relationship between negotiator and allocator objects in the SWIS domain, and in the resource provider domain, could also take the form of Composite Virtual Objects. For instance, an Allocator object could be a Composite Virtual Object that encompasses information and behavior of Allocators from different resource providers. In this way, the Session Control could use a single Allocator that is able to coordinate and compile data from Allocators of several domains.

Another relevant aspect for IoT – one not emphasized by the object model – is events dispatching. Events and commands are essential for this architecture; there is a need for the right object to receive events at the opportune moment. Considering it's likely during a session that many resources will be allocated for a short period of time, dispatching events in a timely manner in relation to many associated objects requires cognitive support. To achieve this, a sort of "cognitive" PubSub engine could be extremely useful to simplify the programming of a dynamic service such as SWIS and others. The cognitive PubSub should be able to intelligently dispatch events to objects even if the specific target object has been deallocated and is not present in the system.

Another interesting IoT architectural issue is to determine, and accommodate, how intelligent terminals can interact and provide functionalities to the SWIS System. Figure 3 graphically illustrates the user terminal as a resource of the system, capable of supporting a large part of functions of the service. In effect, it is a location, visualization, messaging resource and much more. Executing cognitive behavior to the user terminal is a requirement and a big challenge to IoT architecture and systems.

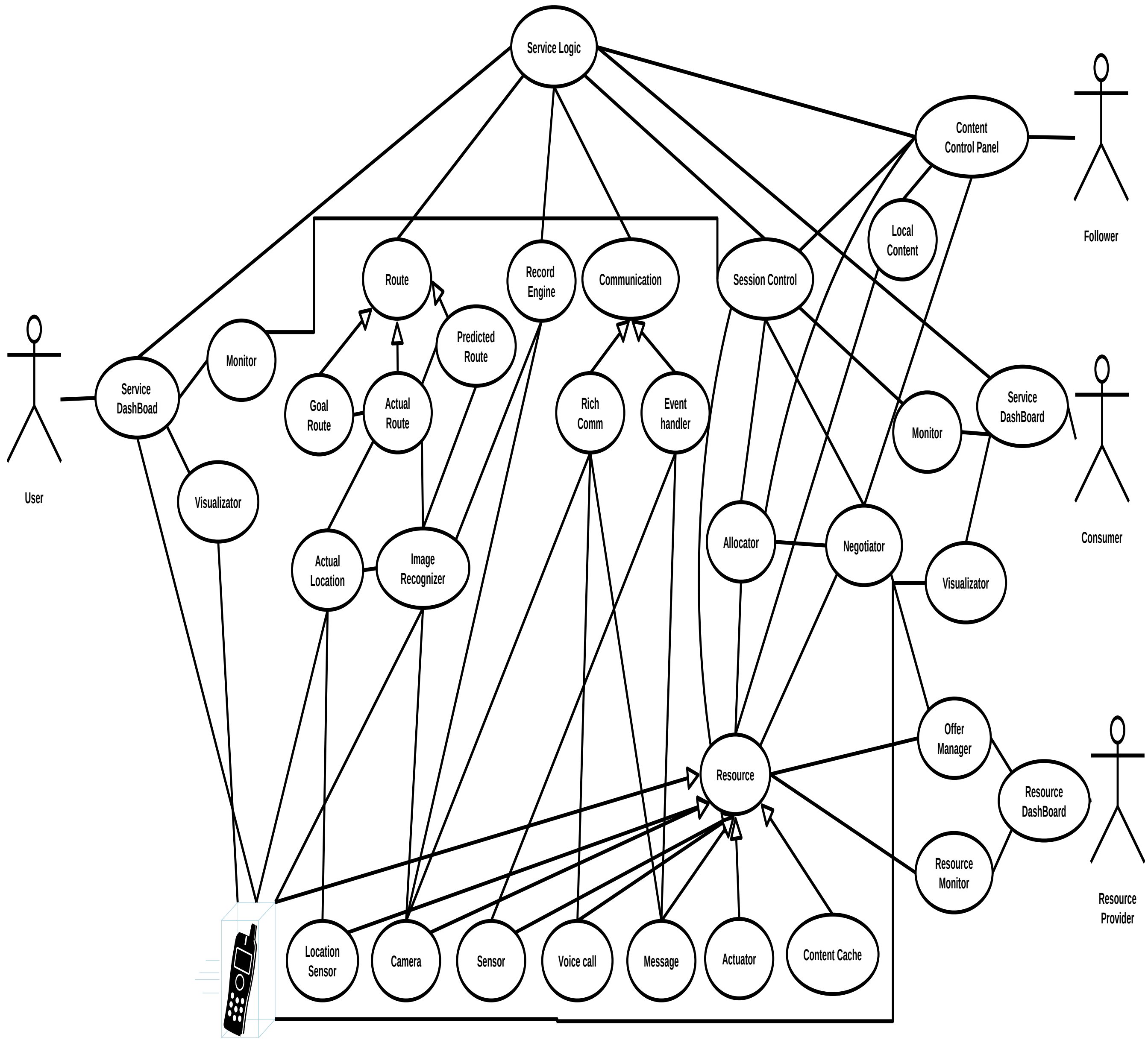


Figure 3: An object model of the SWIS Service

Requirements elicitation

The set of generic services and functions

- Those supported by an IoT infrastructure
  - Collaboration fabric between objects
  - Discovery of services/objects and their negotiation and allocation according to optimization policies
  - Communication and messaging infrastructure (the PubSub messaging)
  - Security/privacy issues
  - Interoperability of various systems with special focus on terminals and their integration in iCore

- Self-Organization of resources and VOs

- Those supported by upper layers (added functions)
  - Cognitive tracking of objects
  - Image recognition, rapid negotiation, bid and allocation of external resources
  - Ability to determine the context of moving objects and to dynamically create a supporting environment
  - Ability to take dynamic control over resources/objects made available by the ambient
  - The dynamic creation of VOs and the aggregation in VCOs

Summarizing, this use case is stressing a number of IoT architecture functionalities such as:

- Mobility of physical objects and behavior of associated VOs.
- Mobility of physical objects and dynamic creation of a meaningful set of resources around them

These two last requirements point to the very fast and dynamic creation and support of a changing "service context".

- Interoperability of objects pertaining to different administrative domains
- Self-management without user intervention

And, in addition, they refer to a powerful self-organizing infrastructure.

It is worth considering other functionalities for an IoT architecture and related systems:

- Specific intelligent functions like tracking of objects
- Intelligent dispatching of events, errors and alarm even if the intended recipient is no longer allocated
- Integration and usage of smart terminals.