Standard for Robotic Localization*

NISHIO Shuichi
nishio@ieee.org

ATR Intelligent Robotics and Communication Laboratories

Abstract. Location information is an essential factor required in every robotic systems. However, up to now, there exists no standard interface for treating location information suitable for robotics. In this paper, an ongoing standardization activity to specify interface for accessing localization results is represented.

Key words: Localization, Standardization, Network Robot

1 Introduction

Location information is a crucial factor in providing robotic services of every kind. Typically, a robotic system is defined as an apparatus equipped with the function of interacting with physical entities in the environment. Navigation, manipulation and human-robot interaction are typical features that require physical interaction with the environment, which distinguish a robotic system from information appliances. On performing such tasks, robots require geometric association between physical entities of interest and the robot itself for implementing and/or performing the given service scenario. Besides these examples, number of location-based robotic tasks is continuously increasing as working field of personal or service robots gradually expand from the controlled, stable factory environments to indeterminate, uncertain daily environments. Treating location information is thus a critical element in composing robots. However, currently there exists no standard means to represent location-related information in robotics, nor any common interface for building localization related software modules.

Although localization is still one of the main research topics in the field of robotics, the fundamental methodology and elements necessary are becoming established. Standardizing localization result representation and related interfaces in a generic form, independent to specific algorithms or equipment, are significant for decreasing costs and accelerating the market growth of robotic services. Moreover, clarifying what types of information are required in the field of robotics shall be useful for equipment vendors such as sensor manufacturers.

Figure 1 illustrate a typical robotic service situation where localization of various entities is required. Here, a robot in service needs to find out where a

* This research was supported by the Ministry of Internal Affairs and Communications of Japan.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 604-615

missing cellular phone is, by utilizing information from various robotic entities (robots or sensors) in the environment. These robotic entities have the ability to estimate the location of entities within their sensing range. Thus, the problem here is to aggregate the location estimations from the robotic entities, and to localize the cellular phone in target.

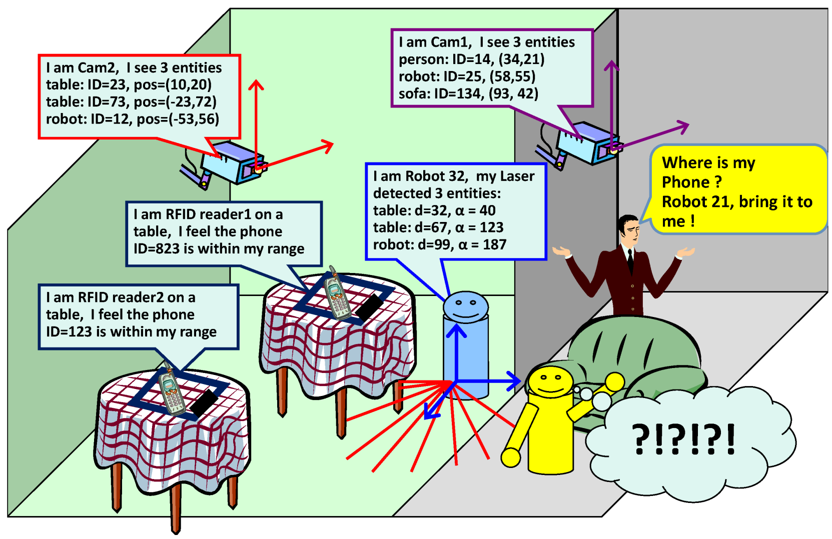

Fig. 1. Example of a typical robotic service situation requiring localization of an entity

Note that, the word "localization" here means "to locate some physical entities through analysis of sensor data." The word "locate" may include not only measuring position in the spatio-temporal space, but also heading orientation or pose information of the entity or obtaining additional information such as error estimation or time of measurement. Also, the word "physical entity" (or "entity" in short) is used to describe the target to be localized, including robots, humans or other objects, and not just the robot itself, in contrast to the common wording in the field of robotics.

This example shows several factors that makes localization in robotics a difficult, challenging issue.

Some sensors only provide partial location information. For example, the camera sensor can only provide 2D information, and RF tag reader can only provide proximity information.

Sensor outputs are not always correct. Sometimes, they might measure two or more entities as a single object, or even miss it. These erroneous reports occur frequently when sensors are used in an uncontrolled daily environment. In or-

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 604-615

der to tackle this erroneous situation, sensor outputs are usually treated to be probabilistic, accompanied with error estimation.

Efforts are required to match observations between results from different sensors. Imagine you are viewing two photographs of a crowded street corner, taken from different angles but on the same instant. The issue here is to match every single person in one photograph to another. Even much difficult efforts are required when matching the results from sensors of different modality, such as matching the results from a wall camera to that from a laser range scanner installed in a robot. As these two sensors measure different aspects of objects, these combinations are much robust in general, but harder to match. This issue, the identity association problem, happens every time multiple sensors are used. Thus, target identity information shall also be treated to be probabilistic.

In order to define standard interfaces for robotic components and representations for data in robotics usage, we have been working at the Object Management Group (OMG). OMG is an international standardization consortium, and is widely known for standards such as CORBA or UML. As a first step, we are now in the process of forming the "Robotic Localization Service (RLS)" standard [1]. This standard aims to define a common framework for treating localization results related to robots, including positions of robots, people or objects. The current specification not only targets the robots available today, but also make concerns of some near future systems under research or development. These include systems such as environmental sensor network [2] or collaboration of multiple robots [3], as in figure 1.

2 Related Standards

Although there exists numbers of standards for industrial robots, there are no standard suitable for robotic services that aim to serve people. Some industrial robot standards, such as [4], defines coordinate systems for treating location and pose information. This, however, is neither sufficient nor flexible enough for treating complex information required for service robots, such as positions of people the robots need to serve.

Another related field is Geographic Information System (GIS). GIS is one of the most popular and established systems that treats location information. In the International Organization for Standardization (ISO), many location related specifications have been standardized (for example, [5]). There already exist versatile production services based on these standards such as road navigation systems or surveying database. However, current GIS specifications are also not powerful enough to represent or treat information required in the field of robotics. As described previously, operations in robotics require much detailed and complex information compared to traditional systems. Still, interoperability with the current GIS systems shall be supported, as robots are expected to work in both indoor and outdoor scenes, and as the existing broad location-based datasets are useful for robots to interact with people. Moreover, trends are that many forms of equipments such as cellphones, home appliances and robots are being coordinated together to server us in our daily life. Considering these kind of ubiquitous system, holding interconnectability with GIS is essential.

3 Robotic Localization Service Specification

Modern robotic algorithms related to localization require not only simple spatial positioning information. Generally, various types of information related to spatial position are also required. In order to obtain precise results, measurement time and error estimation is crucial, for integrating measurements from multiple sensors. One obvious example can be seen when combining outputs of LRF and odometer installed in a mobile robot. When the robot turns around, LRF measurement values quickly change. Thus, if the two sensors are not temporally synchronized, the combined output will result in a mess [6].

Pose information is also important, either for obtaining the coordinate systems that describe each of the robotic body part, for grasping entities, or even for estimating people behavior in the surroundings. When sensors in use can perform measurements of multiple entities at once, target information is also necessary in order to distinguish and relate measurements each other. As such, there are numbers of information to be expressed in combination with simple spatial positioning. In order to make various robotic services treat and process these versatile information easily and effectively, our idea is to represent these heterogeneous information under a common, unified framework.

In this section, a new framework for representing location information required in the field of robotics is presented, by extending the existing GIS specifications. Three extensions required in robotics usage are defined. And then a generic framework for representing structured robotic localization results is defined.

In addition to representation of localization results, the Robotic Localization Service specification defines several functionalities required for accessing and controlling localization data flow. These include the service API, data format, ability description and filtering function. In the following, however, description is mainly focused on representation issue which is the essential part of the specification.

3.1 Relative and Mobile Coordinate Reference Systems

In general, spatio-temporal locations are represented as points in space. Cartesian coordinate is one typical example where location is defined by a set of numeric values that each represent the distance from the origin, when the location is projected to each axis that defines the coordinate system. As described in this example, locations are defined by a combination of information: a coordinate value and the coordinate system where the coordinate value is based on.

Before going further, let us clarify the terms used in the following sentences. *Coordinate system* is a system for assigning an n-tuple of numbers or scalars to

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 604-615

each point in an n-dimensional space (definition cited from Wikipedia). Mathematically, a scalar is in general an element of commutative ring, but we do not apply this definition here. Instead, each of the tuple is allowed to be taken from arbitrary set of symbols, as explained later. A *coordinate value* denotes the n-tuple of scalars assigned with respect to a coordinate system. In this document, we will assume that every coordinate value is related to a coordinate system, through which it was assigned. That is, every coordinate values are typed to be a member of an element set that belongs to a certain coordinate system. Note that, there exists no uncertainty with a coordinate value itself. The uncertainty (or error) with the localization result, if any, is represented by another value. We will call this an error coordinate value (or error in short).

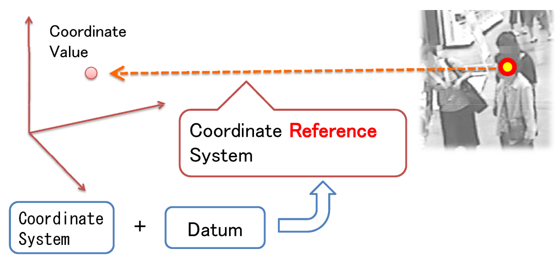

Fig. 2. Coordinate Reference System (CRS)

The basic idea in GIS specifications is that coordinate systems used for representing positional data are fixed relative to the earth (i.e. referenced). There exists descriptions for relative coordinate systems in GIS standards (*Engineering Coordinate Reference Systems*), but they are hardly used and the usage is not clear. In robotics usage, however, coordinate systems are not always fixed, and in many cases they are also mobile. It is possible to express every data in global, fixed coordinate systems, but in most cases, it is much more convenient to treat various data in a relative form. There exists a GIS specification recently published [7] which specifies a method for describing moving entities, but the method is for car navigation and is difficult to use in mobile robots.

Especially in mobile robots, coordinate systems defined on the robot moves in the space with the change in time. For example, imagine that there are two rooms, room A and room B, and a mobile robot equipped with a 3-degree-of-freedom hand. When this robot grasp and move some objects from room A to room B, at least one coordinate systems that represents the area including two rooms, and one coordinate system that moves along as robot navigates, are required. In some cases, each room may have individual coordinate space, related to the global coordinate space representing both rooms in common. Moreover, in order to represent the gripper location at the end of the robotic hand, several coordinate systems are defined over the robotic hand, each related to other coordinate systems by some means such as D-H convention. The object to be moved by the robot may also hold some coordinate systems that indicate the position or the pose of the object. When the object is carried by the robot, these coordinate systems also shift in space as the robot moves. That is, in such a case, it must be better way that the parameters in the coordinate transformation between the coordinate system located on the ground and that on the robot will be changed along with the robot motion, as similar to the D-H convention.

In this specification, a relative coordinate reference system is defined as a coordinate reference system where the relation with the fixed world is not known at some instant, or users have no interest in referencing it to other coordinate reference systems. A mobile coordinate reference system is defined as a relative coordinate reference system, with an dynamic datum (translation parameter) referring to output of different localization output (figure 3).

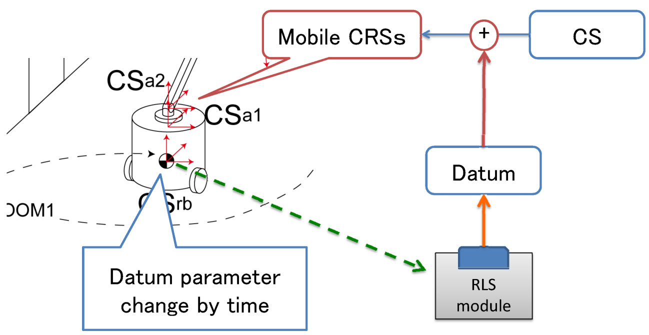

Fig. 3. Mobile Coordinate Reference System

3.2 Identity Representation

Identity information (ID), which is assigned for localized targets, can also be treated as a value on some coordinate system. For example, MAC addresses used in Ethernet communication protocols can be represented as a coordinate value on a two-dimensional coordinate system, vendor code and vendor-dependent code. Electric Product Code (EPC) or ucode, used for identifying RF tags, are another example of identification systems defined by multi-dimensional coordinate system. There also exists some ID systems, such as family names, that is usually not explicitly defined over some mathematical structure.

In general, sensors hold their own ID system, and each observed entity are assigned an ID from this local ID system. This is because, at least on the initial stage, there are no means to identify an observed entity and assign it a global ID. Thus, when multiple sensors are in use, there exist multiple local ID systems independent to each other, and it becomes necessary to properly manage and integrate these ID systems (ID association problem). Also, as we saw in

ISBN 978-88-95872-01-8
pp. 604-615

the overview section, ID assignments are probabilistic, just like other location information.

From these considerations, we can say that ID information requires representation or access methods similar to other types of location information. Thus, we propose to treat ID information in the same manner as spatial positions or other location information, as a value on a coordinate system. Since in GIS specifications coordinate systems cannot handle axis defined over a set of symbols or discrete set of numbers, we extend this point. Note however, that some operations such as comparison is not always defined over this axis, as symbols in ID systems do not form an ordered set in general. Also, transformation between ID coordinate systems will likely to be defined as a conversion table, not an mathematical operation.

Roughly speaking, two types of IDs will be treated in the Robotic Localization Service. The first is the IDs assigned to measured entities and the other is which are assigned to information treated in the service. This latter is used to identify various types of information such as measured coordinate value data or coordinate system definitions. By providing some standard, systematic means on naming these information IDs, constructing repositories containing common definitions such as coordinate systems or inter-coordinate transformation, or even interconnecting multiple repositories will be possible. This can serve for the purpose of enhancing the reusability of servicing modules, and to increase efficiency and decrease costs in the development of various robotic systems. GIS specifications define and utilize such naming system, based on the Universal Resource Name (URN), a notation standardized by the W3C. A common repository based on this naming system results not just in enhancing the reusability of modules. Effects such as decreasing ambiguity in coordinate system definitions, or semi-automatic construction of inter-coordinate transformation module can be attained. This useful usage can be inherited as well in the proposed robotic localization service.

Note that this specification does not force the developers or users to use some fixed naming systems. End users always have the freedom to use whatever naming system they want, at least on the range where specified here. However, if reusability or inter-operability in heterogeneous systems becomes a matter, a standard naming system will be a powerful mean for enhancing these features.

3.3 Error Representation

Error (or uncertainty, reliability) information plays a important role in robotic algorithms. This is especially important when the localization results are used for further processing such as fusion with other results. Thus, representing errors in location information is one of the essential features for the robotic localization service. In GIS specifications, only static information of expected error concerning inter-coordinate transformation can be stated. Thus, here we extend the GIS specification in the following points:

- Each measured location information can be attributed an error information.

– Allow various types of error representations, such as simple reliability, co-variance or distribution approximation by particles.

Normally, error information is combined with one main localization result. However, in certain cases, there is a need to hold an integrated error related to multiple location data. One commonly used example can be seen in a typical Kalman filter usage, where spatial position and velocity are used to form a state vector, and the co-variance matrix is defined over a composed space of these two elements. In such case, the error information (here, the co-variance matrix values) shall be explicitly related to the two localization results. Details are described in the next section.

3.4 The Robotic Localization Architecture

The Robotic Localization (RoLo) Architecture defined here is a unified framework for organizing and representing complex data set required in robotic localization. Similar to the relation between GIS location data and coordinate reference system, every localization data is *typed* to some RoLo Architecture. Figure 4 shows an example of RoLo Architecture definition. In this example, three elements are considered: measurement time, target ID and spatial position. Note that each element contains an error attribute (note: error attribute is not mandatory).

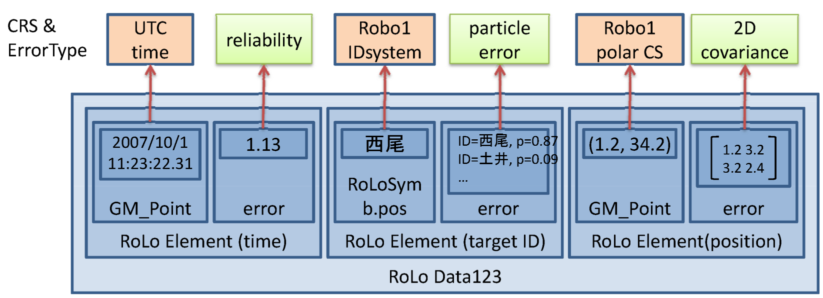

Fig. 4. Example definition of RoLo Architecture

Normally, error information is combined with one main localization element. However, in certain cases, there is a need to hold an integrated error among multiple location data. For example, in a typical Kalman filter usage, multiple main location information such as spatial position and velocity are used to form a state vector. When the elements of the state vector are not independent, which is the usual case, the corresponding error, the covariance matrix, is related to multiple main elements. In such case, the Error Element Specification (derived from RoLo Element class) instance specifies which main information slot the

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 604-615

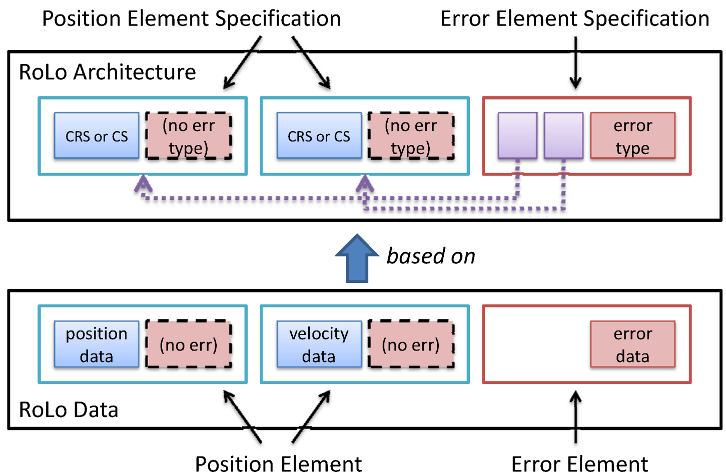

Fig. 5. Sample of Complex RoLo Error definition

error is related to, and the actual error data is contained by the Error Element class (derived from RoLo Element class) instances.

In definition of RoLo Architecture, coordinate systems that each required values are based on, are kept independent to each other, and individual values remain just as its original form. This means that, definition of each values are kept in their original form, and further, their relation to be treated in combination is expressed. In other words, multiple information are represented in combination, to be suitable for certain usage, still remaining the 'meaning' of each individual values. Note that, this specification does neither oblige the users to specify information of some 'meaning', nor restrict the 'meaning' of information expressed by RoLo Architecture. For example, the spatial coordinate in the above example may represent the centroid of the robotic body, or it may represent the position of a robotic arm. The meaning of each coordinate information contained in RoLo Architecture definitions are out of the scope of this specification. Only the users and provider of the information needs to agree in how each coordinate information will be interpreted.

4 Service Interface

Several types of modules are commonly used in robotic localization services in general. The simplest form of module is that which receives data from sensors, calculates location and outputs the results. However, this type of interface strongly depends on sensor interfaces or sensor output formats. Strong dependency on specific products or vendors is not suitable for standardization. Moreover, when a location is calculated, many kinds of resources such as map data, specific to each sensing system, are required. It is impractical to include each of these resources into the standard specification. Thus, we decided to embed and hide the individual device or localization algorithm details inside the module structure.

On the other hand, if we focus on functionality required to localization modules, we can classify them into roughly three classes:

- Calculate localization results based on sensor outputs (measurement)
- Aggregate or integrate multiple localization results (aggregation, fusion)
- Transform localization results into different coordinate reference systems (transformation)

These functionalities differ in their internal algorithms or the number of input / output streams. However, in all of these, the main data to be exchanged is localization results. As we are focusing on the interface of RLS modules, and not on their functionalities, we decided to abstract these different types of modules into a single form of module (figure 6). This abstract module holds n input streams and one output stream. By abstracting various types of modules and assuming a uniform interface, complex module compositions such as hierarchical or recursive module connections can be easily realized.

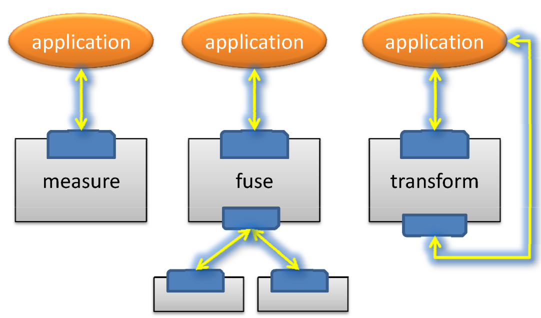

Fig. 6. Sample of Complex RoLo Error definition

Each stream owns an ability description, which shows how this stream can perform and be configured. This includes the list of RoLo Architecture or data formats that this stream can handle, or other configurable parameters specific to the module. Each service module also owns an ability description for the service it provides, besides the ability description for its output stream. The configurable parameters defined in the ability description can be specified values via the module interface, restricted by the ability description held by the service module or the belonging stream modules.

If each module can represent what it can perform, or provide information on available configurable parameters, a large amount of development efforts can be reduced. By defining the "meaning" of parameters, the ambiguity in functional

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 604-615

definition or parameters can be eliminated, resulting in increase of developing efficiency. Moreover, advanced features can be implemented such as verification of inter-module connection, automatic search of specific modules or semi-automatic parameter negotiation between modules. In cases where sensors or robots distributed in the environment cooperate with each other, namely the Network Robot environment, it becomes essential to register each module's capabilities in repositories and make them searchable.

5 Conclusion

In this paper, current status of robotic localization standardization was introduced. This standard is scheduled to be released on end 2009, and comments are still welcomed. In Japan and Korea, several national projects have started to use this specification for their robotic systems. One such system is the *Kansai Environmental Platform* [2], which is a sensor-equipped environment built in a shopping mall near the Universal Studio Japan (figure 7). As in this implementation, the standard described here can be easily extended to treat and exchange symbolic information, such as what a person is doing or what kind of place a person is staying now. As these symbolic information makes the description of robotic service flow much easier, compared to raw location data, further generalization of localization service may be a future issue.

Fig. 7. Symbolic and numerical localization data measured in a shopping mall. The measurement is done by a environmental sensing system, and is provided to service robots.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 604-615

One of the issues regarding standards of this kind is how vendor-specific functionalities or raw data can be represented. Although in this paper we have focused on describing the representation of localization data, the RLS specification includes elements such as data format or vendor-specific parameter specification, that can be utilized for treating vendor specific data or interfaces. However note that most of the existing outputs (including raw sensor data) follows some rule. This means the data is defined over some system, and this fact allows the data to be represented by means of coordinate systems in most of the realistic cases. Although the specification will generally not allow users to describe data based on grammar, in realistic cases this limitation can be avoided by defining symbolic axes based on codebooks such as the symbolic information described above.

The localization standard is only one of the fundamental specifications required to build robots with common interfaced modules. Profiling on robotic devices, or descriptions on robotic service flows may be other candidates for standardization. Also, as stated briefly in the last section, by defining and exchanging module description metadata, much flexible and dynamic configuration of robots, composed of distributed modules over networks will be available. This will also make the interconnection with heterogeneous network systems such as ubiquitous sensing systems or GIS systems much easier. Such architecture and metadata repository shall be also of concern.

Acknowledgement Standardization of the Robotic Localization Service has been worked at Robotic Localization Service working group in OMG Robotics Domain Task Force. I would like to thank the co-chairs and the members of the working group.

References

1. Nishio, S., Han, K., Kim, Y.-H.: Revised submission for the Robotic Localization Service RFP. robotics/08-05-02, Object Management Group (2008)
2. Nishio, S., Hagita, N., Miyashita, T., Kanda, T., Mitsunaga, N., Shiomi, M., Yamazaki, T.: Robotic Platforms Structuring Information on People and Environment. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (2008)
3. Hagita, N.: Communication Robots in the Network Robot Framework. In: Int. Conf. Control, Automation, Robotics and Vision, pp. 1–6 (2006)
4. International Organization for Standardization (ISO): Manipulating industrial robots - Coordinate systems and motion nomenclatures. 9787:1999 (1999)
5. International Organization for Standardization (ISO): Geographic information - Spatial Referencing by Coordinates. 19111:2007 (2007)
6. Ueda, T., Kawata, H., Tomizawa, T., Ohya A., Yuta, S.: Mobile SOKUIKI Sensor System -Accurate Range Data Mapping System with Sensor Motion-. In: Third International Conference on Autonomous Robots and Agents, pp.309–314 (2006)
7. International Organization for Standardization (ISO): Geographic information – Schema for moving features. 19141:2008 (2008)

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 604-615