Event Recording System for Smart Space Applications

Andrey Vasilev, Egor Krylov, Alexey Subbotkin, Ilya Paramonov
P. G. Demidov Yaroslavl State University, Yaroslavl, Russia
vamonster@gmail.com, Ilya.Paramonov@fruct.org

Abstract

Smart space applications consist out of several agents interacting with each other. Their operation form an ubiquitous environment assisting actions of the user. There could possibly be several applications running at the same time supporting different human activities. In this paper we address the issues of identification of meaningful event, context information gathering and visualization with the use of a developed tool. The application allows to display the current or recorded context information using the time line visualization approach.

Index Terms: Context Capturing, Event Capturing, Smart Space, Smart-M3.

I. Introduction

Smart space is a paradigm for developing applications in accordance with ubiquitous computing approach. The paradigm incorporates knowledge and spatial approaches [1] for implementing the next generation of user-centric applications. Smart spaces are usually attached to a specific location and allow to exchange information between applications and devices in the area, so they could cooperate for better user living or work operation. Future smart space applications shall be proactive and be able to act on behalf of a user to minimize involvement of human intelligence in all stages of service provision chain. The key enabler for that is efficient capturing and use of the context information.

The Smart-M3 platform [2] is the information sharing platform, which provides services to create smart space applications. The platform implements the idea of the local data storage of semantic web information [3]. The core of the platform is the semantic information broker (SIB), which provides services for data storage and modification. The information in SIB is stored in the form of RDF triples. The applications implemented on top of the platform consist out of several agents running on a suite of devices including embedded and mobile platforms. The agents can query, modify and subscribe to the data changes in the RDF store.

A set of information, which KPs of a smart space application use during operation, is usually defined by the ontology or a set of linked ontologies, but it he is not required to do so [4]. There is no common standard for description of agent operation, so each architect must design own description for interaction. This brings the problem of detecting end user identified events, because there is a variety of approaches for inter-agent communication. Also it is hard to determine the context of the event as arbitrary information could be used to describe the event.

In this paper we propose an approach for the event definition for smart space applications and the corresponding context data specification. Also the architecture for context information capturing application is given. Such application allows to view the current events and readings from the plain sensors placed in smart environments.

The paper is structured as following. In Section II the description of previous studies in current field is given. Section III defines the event description ontology and provides an example of ontology usage. In Section IV the description of context capturing application is given. Section V summarises the paper.

II. Related Work

One of the base information sources for context data is sensor readings. There are special tools for recording and playback of all gathered readings from the sensors. PyViz project [5] is devoted to capturing and playing back the sensor readings done in a smart house-like environment. The users need to create the structure of the house inside the application, place sensor visualizations on corresponding places and link them with data sources. Cheun at el. [6] propose a special framework for capturing mobile context of the user. The mobile context is generated using sensor inputs available for the mobile phone, e.g. accelerometer, absolute position of the person, light sensors and so on. The visualization application, implemented as a web service, allows browsing the sensing history of the single person or a group, or some extrapolation for a period of time. The approach presented in these works should help to capture useful information about end users for the future context-aware services. Though providing a lot of context information, such approach requires user to manually associate raw data with substantial events.

In our previous work we have created a system for capturing event information in the form of mind map on top of open source mind map editor HiveMind [7]. The main idea behind the aggregate information from the smart space into the hierarchical representation in the form of the mind map. Such structure allows to see the correlation of recorded events and their context. The context is defined as the data trees, which are formed from neighbour nodes and their child nodes. Though being descriptive such formation becomes bloated with the collected data and it is harder for user to perceive the information. Another disadvantage of such scheme as it hard to present the sequence of events without the introduction of additional nodes to the map, which take additional space on the view.

III. Event Description for Smart Space Applications

The Smart-M3 platform supports specifying the behaviour of agents, but there is no common approach to do it. In order to implement event and context capturing system we needed a way to specify the behaviour of the arbitrary application build on top of the platform. So, we decided to provide the description of events based on the information, which this application manages.

A. Description Requirements

The descriptive information of the event could possibly be relevant for a wide class of applications, therefore it is desirable to provide this information along with the application data. So we decided to give the description in the form of RDF triples and develop the corresponding ontology. The requirements for the ontology were formulated on top of Event [8] and TimeLine [9] ontologies.

Event ontology should describe the following information:

- Different event types:
  - State change events, e.g., by capturing corresponding event, the system changes the state. Such events can also be named as long events.
  - Notification events. They do not change the state of the system, but indicate that an event took place.

- Parent-child relationship for events, event hierarchy. Mainly interesting for state change events.
- The descriptive information of the event, e.g., which target ontology entities contain information.
- Event triggers: the triple, the modification type (addition, removal, update), check expression.
- The visualization template for the event, not all available information can be used. There could be two templates: the brief overview and the full description.

B. Ontological Model

The structure of the ontology is presented in Fig. 1

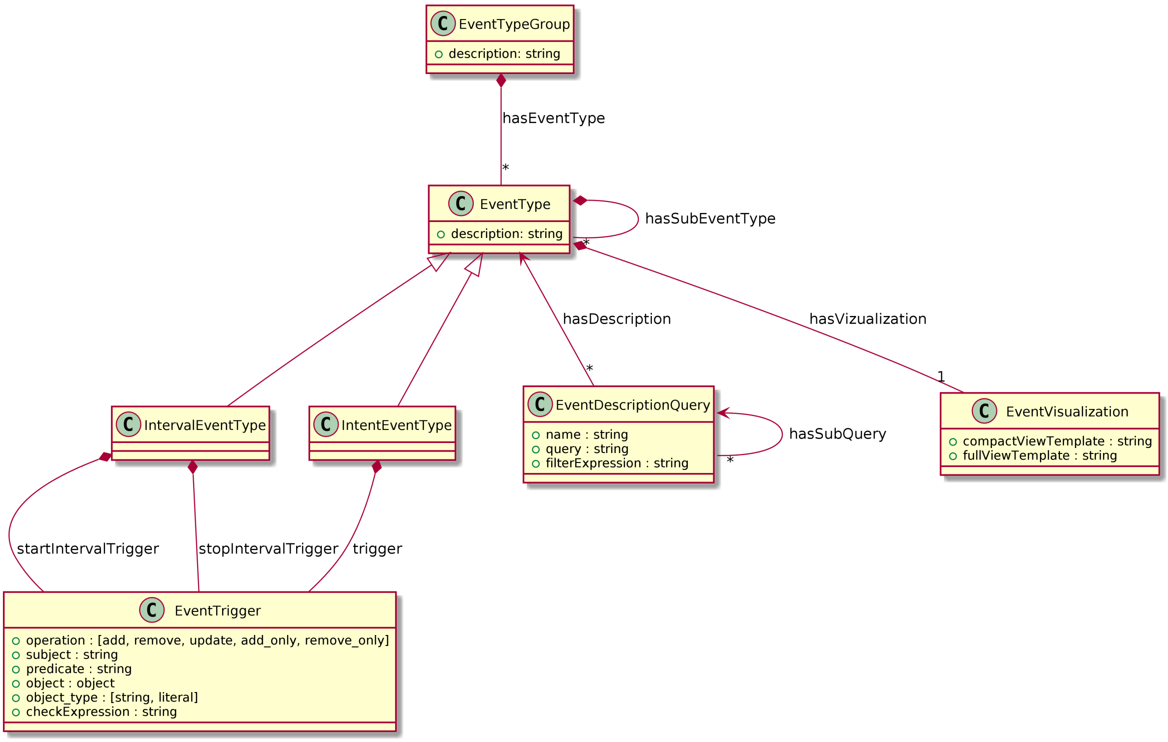

Fig. 1. Event description ontology structure

The EventTypeGroup class is the aggregative entity for all events that smart space application generates. This entity is required as there could possibly be several applications running at the same time and it would be harder to sense the context of the event if they come from different sources into a common visualization space. So each application should declare it's own EventTypeGroup entity.

There is a set of EventType entities, which correspond to each EventTypeGroup entity. The EventType class is a meta-class, which forms the hierarchy of events and should not be used directly. There are two derived classes: IntervalEvntType and IntentEventType. Entities of the first one should be used to describe the state change events, the second one are notification events. They differ in the number of triggers required to register the event. The notification event requires only one trigger and state change event demands two triggers, which indicate the beginning of the event and the end. If there is a set of internal states for the smart space application, then IntervalEventType could be nested.

The EventTrigger class is used to describe the pattern for information modification in the smart space that should notify the change in the event space of smart application. It describes the triple pattern and expected operation on top of that triple. There were depicted five distinct operation types. It is also possible to provide an additional check expression, which could filter out false triggers. There is no concrete description for the expression and it left as an implementation detail for event gathering system. In our system we will use python expressions to check trigger.

The context information about an event should be gathered on top of the queries defined by EventDescriptionQuery entities. Potentially all information in the smart space could be used for the event description, but it would not be resource-efficient to gather all information from the space. The queries could be nested, therefore it is possible to get complex data. There is a variety of query languages that could be used to retrieve data. The Smart-M3 platform supports plain triples queries and SPARQL query language. Each could be used to query data. Retrieved data could be filtered-out if it is not supported by query language.

Last element of the ontology EventVisualization allows to define the visualization of gathered information. We presume the use of HTML language to present information to end-user, so properties of EventVisualization entities contain an HTML template. There are supported two kinds of information views short one and full. First could be used if there is a lot of events to display and not enough visualization space available.

C. Example: Events in Smart Conference System

The Smart Conference System [10] is an intelligent system that assist the conference process. It consists out of whiteboard KP, which maintains the schedule, projector KP, which displays the slides of the presenter, and end-user KP, which allows to see all information on ones mobile device and control presentation flow. In fig. 2 the part of the event description for Smart Conference System is presented.

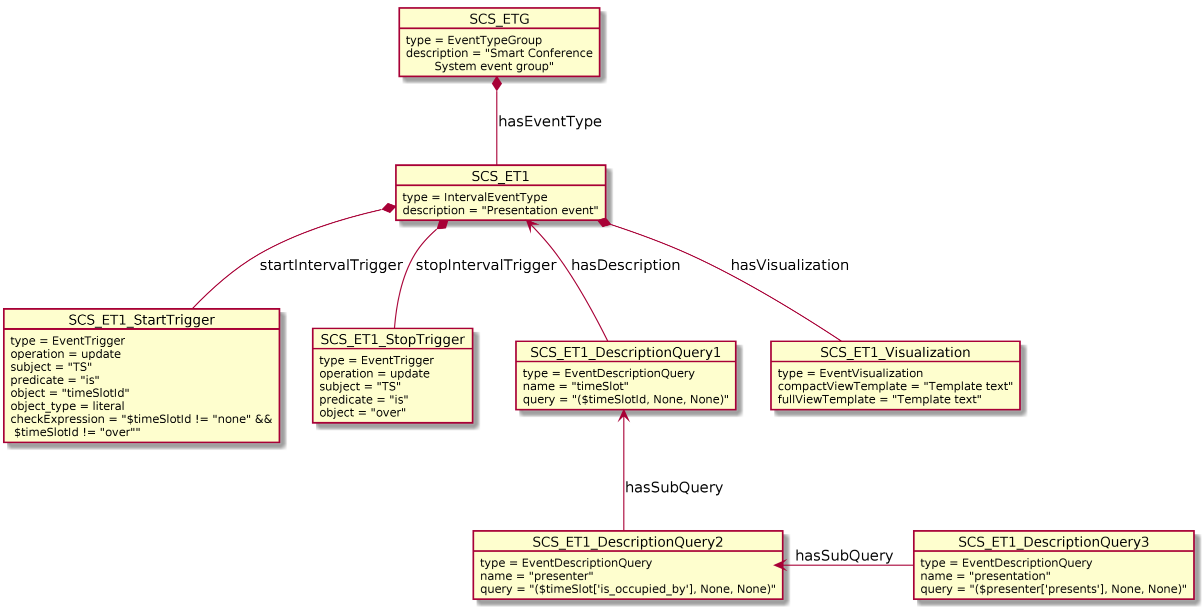

Fig. 2. Part of Smart Conference System event description

There is the description for one interval event, which describes the presentation. The event begins when whiteboard KP changes the triple *('TS', 'is', timeSlot)*. The end of event denotes the modification of the previous triple to *('TS', 'is', 'over')*. Event is described with the series of the triple-based queries, which retrieve information about time slot, presenter and presentation. The visualization templates are provided, but omitted in the figure.

IV. Event Recording Application

In current section the description of event recording and visualization application for smart space environments is given. The application is designed to capture event information from an arbitrary Smart-M3 application.

A. Event Sources

There are two main classes of event sources in smart environments created using the smart space approach, which differ by the source of origin. The first class consists of events generated by smart space applications during their work. The second class is constituted by the readings of standalone sensors or sensor networks. Key characteristics of each event type are presented further.

Smart space based events are defined as following:

- Event is some slice of information from the smart space at arbitrary time. In common case it is the information about the entity and linked entities, which describe perceived event in the real world.
- Event trigger is the modification of the triple in a smart space, satisfying predefined condition. For now, the addition, removal, modification operations are depicted. The check condition can be defined in an arbitrary way.

Sensor readings are described by the following information:

- Concrete location, where readings are taken;
- Type of the readings;
- Minimum and maximum values of readings
- Current value of the readings.

B. Application life cycle: the order of initialization and data collection

The application could operate as an interactive event logger and event review application. During the event capturing phase, the application consumes information and visualizes it in real time. During the log review phase user can see the recorded events. The additional information can be retrieved by clicking on a component visualization.

When the application connects to the smart space, it searches for event descriptions, which are done according to event description ontology. Then it subscribes for specified event triggers. When such trigger fires, the application initiates a series of queries to gather event description data. The captured event is further visualized for the end user.

There could possibly be several smart applications using distinct ontologies in one smart space. For each smart application the entities of Event Description ontology, describing the events and their triggers, are supplied. It is assumed, that only events, which are descriptive and important for the end-user are depicted by Event Description ontology.

There could be several ways to obtain sensor readings. Specialized message passing protocols could be used to transmit information from sensors or they could be delivered through the smart space. In current implementation, the direct communication with the provided sensor will be used.

C. Applied visualization techniques

One of the issues of context data gathering applications is the visualization of the collected information. In previous work we used the mind map representation, which worked well for small amount of information, but proved to be less useful for long-term data collection. In [11] and [12] authors show the approach to visualize timed and hierarchical data on the time line. We decided to use such approach to present information in our application. The sketch of the interface is shown in Fig. 3.

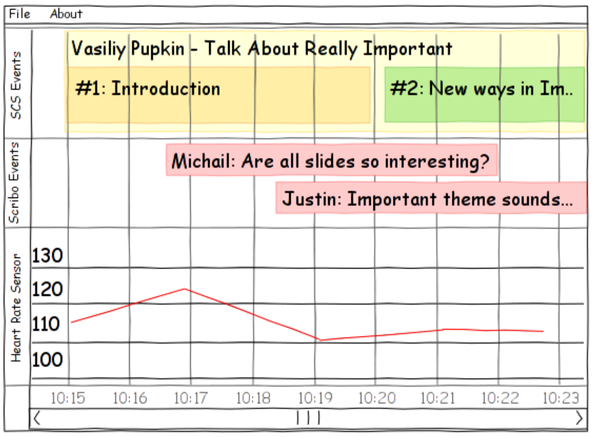

Fig. 3. The sketch of application main window

The visualization space of the application is split into multiple non-overlapping lines. Each line refers to either events from a smart space application or readings from a sensor in smart environment. On the bottom of the visualization space the time scale is presented. Readings from sensors are presented in the form of value graph. Sensor lines also contain the scale ranging from the minimum value to the maximum value of captured data.

Smart space events are presented as the filled rectangles, containing one or several lines from short description of event. The beginning of the instant event view corresponds with the time, when event was registered. The width of the rectangle is set in accordance with event description. The beginning and the ending of interval event view is set according to corresponding event time capturing. User may see the full information about event by clicking on it.

V. Conclusion

In current paper we presented an approach to describe events in the smart space environments. The developed ontology could be used to gather meaningful context information about noticeable events. Such approach was used to design the system for context information gathering. The application is planned to be implemented and tested as a part of smart space services on FRUCT 12 conference.

The use of ontology is not restricted to data collection, but could be used as an assessment for smart space application integration, as it provides additional operational description to ontological data description. Another point of application could be the base of information collection for different context reasoning techniques, because of preliminary notation of important moments in application operations.

VI. Acknowledgements

The authors would like to thank members of Internet of Things and Smart Spaces FRUCT working group for their feedback.

The paper is a part of the research carried out within the project "Cross-platform future services - technologies of smart space and internet of things" funded by grant of Federal Program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013" contract 14.B37.21.0876.

References

[1] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, A survey of context modelling and reasoning techniques, *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161180, 2010. DOI: 10.1016/j.pmcj.2009.06.002.

[2] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö, Smart-m3 information sharing platform, in *Proc. IEEE Symp. Computers and Communications*, ser. ISCC 10, Riccione, Italy: IEEE Computer Society, Jun. 2010, pp. 10411046, ISBN: 978-1-4244-7754-8. DOI: http://dx.doi.org/10.1109/ISCC.2010.5546642.

[3] I. Oliver, "Information spaces as a basis for personalising the semantic web", *Proc. 11th International Conference Enterprise Information Systems (ICEIS)*, SAIC, May 2009, pp. 179-184.

[4] D.G. Korzun, A.A. Lomov, P.I. Vanag, S.I. Balandin, J. Honkola, "Multilingual ontology library generator for Smart-M3 information sharing platform,"*International Journal on Advances in Intelligent Systems*, volume 4, 2012, pp. 68-81.

[5] B. Thomas and A. Crandall, A demonstration of pyviz, a flexible smart home visualization tool, in *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 304306. DOI: 10.1109/PERCOMW.2011.5766889.

[6] M. Kim, H. La, H. Bae, C. Keum, S. Kim, et al., A practical framework for comprehensive mobile context visualization, in *e-Business Engineering (ICEBE)*, 2011 IEEE 8th International Conference, IEEE, 2011, pp. 201206. DOI: 10.1109/ICEBE.2011.24.

[7] Oleg Kandaurov, Andrew Vasilev, Ilya Paramonov, "Integration of HiveMind Application into Smart Conference System", *Proceedings of the Congress on intelligent systems and information technologies "IS&IT'11". Scientific publication in 4 volumes*, volume 4, Moscow, 2011, pp. 71-76.

[8] Y. Raimond and S. Abdallah. (Oct. 2007). The event ontology, [Online]. Available: http://motools.sourceforge.net/event/event.html

[9] Y. Raimond, S. Abdallah. (Oct. 2007). The Timeline Ontology, [Online]. Available: http://motools.sourceforge.net/timeline/timeline.html

[10] D. Korzun, I. Galov, A. Kashevnik, N. Shilov, K. Krinkin, Y. Korolev, "Integration of Smart-M3 applications: Blogging in smart conference", *Smart Spaces and Next Generation Wired/Wireless Networking*, June 2011, pp. 51-62.

[11] J. Zhao, S. Drucker, D. Fisher, and D. Brinkman, Timeslice: interactive faceted browsing of timeline data, in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ACM, 2012, pp. 433436. DOI: 10.1145/2254556.2254639.

[12] P. André, M. Wilson, A. Russell, D. Smith, A. Owens, et al., Continuum: designing timelines for hierarchies, relationships and scale, in *Proceedings of the 20th annual ACM symposium on User interface software and technology*, ACM, 2007, pp. 101110. DOI: 10.1145/1294211.1294229.