Extensible Data Management Architecture for Smart Campus Applications
A Crowdsourcing based Solution

Attila Adamkó and Lajos Kollár
Department of Information Technology, Faculty of Informatics, University of Debrecen, Debrecen, Hungary

Keywords: Smart Campus, Crowdsourcing, Participatory Sensing, Data Integration.

Abstract: The technological advancements that have occurred during the past decade in various domains, including sensors, wireless communications, location positioning technologies and the web, allow the collection of a wide range of data. Possible sources of that data include intelligent devices (smartphones, tablets, etc.) containing various sensors, Web pages, and social networking sites. Collected data are subject to analysis (using data mining or pattern recognition approaches, for instance) and after processing new content might be inferred. This is a value-added service that can itself be used as a data source. In this paper, we use our University Campus as an example for establishing a data management architecture that integrates into a more general, extensible publish/subscribe based model of crowdsourced applications.

1 INTRODUCTION

The recent rise in ubiquitous computing (ubicomp) and supporting devices resulted in lots of new applications that are able to exploit the advantages of this approach. According to (Luo, 2012), "ubicomp is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities". The collected data from the built-in sensors of smartphones, tablets, phablets, and embedded devices let the applications based on participatory sensing become more and more popular and useful. Moreover, in a real-life environment we need additional (user- or application-generated) data sources in order to provide valuable services.

A Smart Campus is a good prospect of applying participatory sensing combined with other data and event sources and due to the number of people studying, working or even living at a Campus and considering their commitment of using novel technologies and applications, it is an ideal choice for developing value-added services based on crowdsensing.

This paper is organized as follows. Section 2 describes the context of our research and explains the reasons why we selected Smart Campus for our focus. Key challenges and our research goals are also defined here. Section 3 briefly describes the related work. The main contribution of the paper, namely, the supporting architecture is discussed in Section 4 while Section 5 contains some concluding remarks.

2 SMART CAMPUS AS A PART OF A SMART CITY

The context of this research is a project entitled Future Internet Research, Services and Technology (FIRST[1]) has been started in 2012 by University of Debrecen and its consortial partners. The research objectives of the project include to face the challenges posed by the Internet, to carry on investigations covering basic theoretical questions, network modelling examinations, the reassessing of network architecture, content management, and the creation of an intelligent application platform.

The aims of one of its subprojects, Future Internet for Smart City Applications, are threefold:

1. to establish an open, mobile crowdsensing application platform for smart city applications,
2. to provide data management and knowledge discovery solutions for them, and
3. to develop prototype applications based on the

[1] http://first.tamop422.unideb.hu/About-the-project.html?language=en

Adamkó A. and Kollár L..
Extensible Data Management Architecture for Smart Campus Applications - A Crowdsourcing based Solution.
DOI: 10.5220/0004964802260232
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 226-232
ISBN: 978-989-758-023-9

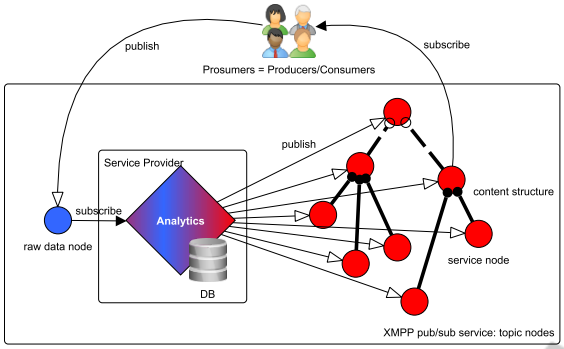

Figure 1: XMPP-based publish/subscribe architecture (Szabó and Farkas, 2013a).

common platform combining the data management solutions with real-time analytics.

The open platform has already been established (Szabó and Farkas, 2013a; Szabó and Farkas, 2013b; Szabó et al., 2013). Figure 1 shows how the model can directly be mapped onto the Extensible Messaging and Presence Protocol's (XMPP) publish/subscribe (pubsub) service (Saint-Andre, 2011).

During our work to design data management infrastructure, we started to investigate the field of Smart Campuses instead of Smart Cities (fortunately, this also fitted into the overall project). The reason for it was that there are lots of similarities between the challenges of developing Smart Campus and Smart City applications (the corresponding terms used by Figure 2 and (Szabó et al., 2013) are given inside parentheses):

Lots of Potential Users (Consumers)
University of Debrecen has as much students and employees as the number of inhabitants of a medium-sized Hungarian city. These users are consuming the provided services.

High Variety of Data Sources (Producers)
Information in various formats coming from multiple sources should be integrated in both cases (e.g., timetable, academic calendar, information on consultation dates and times in case of Smart Campus, or energy consumption, traffic info, etc. for Smart Cities). Social media sites and geolocation sensors are examples of sources that are common in both domains.

Need for Value-added Services (Service Providers)
Service Providers give added value to the crowd-collected raw data. Basically we can say that this is why someone would like to use the applications (e.g., for being notified about a room change or finding a jam-free route between two points).

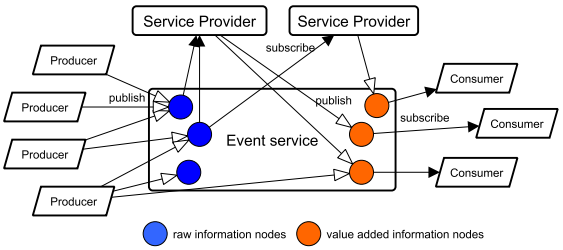

Figure 2: Crowdsourcing model based on a pubsub scheme (Szabó et al., 2013).

Examples of such value-added services include but not limited to:

- get notified when one of my favourite bands (based on my listening history from last.fm or Google Play Music) will play in a concert venue of the Campus,
- get notified when an event's (class, concert, etc.) date and time is modified,
- based on the location data provided by some of my friends I can realize that they are going to have a lunch so if I am in a hurry I can join them,
- when preparing for a consultation with an instructor, get notified if too many students plan to visit the instructor's office hour at the same time, and assist of reorganizing my schedule based on my task list.

We have a much longer list describing various use cases for Smart Campus applications but since the focus of this paper is not on the applications but the underlying data management architecture, we omit the details of use cases.

An advantage of building a prototype application on the Smart Campus field over a Smart City is that young people's (especially students') willingness to use the applications and provide valuable feedback on them is much bigger than elders' therefore it seemed to be a good idea to test and evaluate the underlying data management architecture. On the other hand, participatory sensing can successfully be applied as students actively use handheld devices and social media (presumably they are much more active than city inhabitants are). Our aim is to build services that can be used to infer some patterns regarding the operation of the community, and the derived information can be utilized as a part of a new service offered for the community after applying appropriate analytics.

Smart Campus applications should fit into this framework so they need to follow an XMPP-based pubsub approach with crowdsourcing. However, in order to achieve platform independence, we decided to encapsulate services into Web services which allow

to develop consumer apps on any of the most popular mobile platforms (Android, iOS, Windows Phone) along with traditional Web applications.

2.1 Challenges

A Smart Campus environment has lots of various (and most importantly, heterogeneous) data sources including the following:

- an Education Administration System called Neptun that contains information on course enrollments, timetable information of courses, exam dates and times, etc.,
- faculty members offering office hours, consultations, etc.,
- Education Offices of the various faculties offering office hours,
- Student Governments organizing events for students,
- the menus of the canteens located at the Campus,
- geolocation (e.g., GPS), WiFi or some other sensor data collected by smartphones or similar devices,
- data gathered by environmental and building sensors (temperature, humidity, air pressure, air pollution, etc.),
- a Library Information System that is able to tell whether a given book is available or not,
- social media sites (like Facebook or Google+) containing information on friends and ranges of interests of a person,
- professional sites (like LinkedIn) holding data on work experience and professional achievements (however, this is not necessarily the most important data source from Smart Campus perspective),
- bibliographic databases (like Google Scholar, DBLP or Scopus) that provide information of published journal articles or conference papers of researchers,
- event hosts of actually any events (like public lectures, concerts, exhibitions or whatever users might be interested in), and, which is essential,
- the crowd itself with the added value of the capability of generating content that is interesting for a set of people (or, to be more precise, consumers).

Of course, this is not an exhaustive list, these have only been listed in order to demonstrate how diversified sources of data can be used in a complex application. Valuable applications typically require integration of some data originating from several of these sources. Therefore, our main challenge is to create an architecture which allows the access of information from existing sources while also easing the addition of new sources in a seamless manner.

Some of those data can be gathered in an automated way (like sensor data), some of them may require manual interaction (e.g., canteens' menus or instructors' office hours); some of those data sources offer Application Programming Interfaces (APIs) to provide access to data (social media sites, for instance) while others do not have APIs so some web spiders are required to parse the data; some of the data sources provide built-in notification mechanisms (e.g., an event feed of a social network site) while others do not (for example, adding new office hours or changing the daily menu).

Considering the amount of data originating from those sources, we can state that a big data management solution is required. The frequency of arriving data elements can significantly differ, some of them (e.g., inserting a new article into a bibliographic database or offer office hours for the forthcoming semester) might be quite rare while others (especially when collecting location data) can be very frequent.

This will be further discussed in Section 4.

2.2 Our Goal

An architecture for data management and knowledge discovery is needed that fits into the framework shown in Figure 1. This architecture should address the challenges described in Section 2.1 and be as generic as it can be in order to integrate a high variety of data sources regardless of how the data is processed. Extensibility is a key concern: the design should allow to add new sources and define their processing in an easy way.

Besides setting our goals, it might be useful to discuss what are the non-goals. On one hand, discussions on the use of XMPP will not be provided as the overall framework was given. On the other hand, the development of value-added services may (and will) require data mining activities (including understanding the semantics of data in order to eliminate duplications), as well, however, they are completely out of scope of this paper as we here focus only on the data management platform that can (and should) later be complemented with on-line data analytics solutions which are currently under development (by some of our colleagues). Data security is an important issue that needs to be addressed in the technology stack, however, our opinion is that it must be introduced at a higher level than the data management level, therefore we omit the discussions regarding that field.

3 RELATED WORK

There have been innovative steps in Smart City development over the past few years. It became essential to develop a platform that will aggregate all the available (related) information and will orchestrate it in the best way as possible, towards meeting the defined goals.

3.1 Smart Campus

There is an even growing literature for Smart Campuses which are related to data integration and real-time processing of massive heterogeneous data sources. An application prototype using semantic technologies is discussed in (Boran et al., 2011). The idea behind this approach is to include semantic information using ontologies with OWL and SPARQL. Another approach (Valkanas and Gunopulos, 2013) deals with event detection from real-time data using heterogeneous sources.That experiment shows that information extracted form social networks and sensor data can be used for identifying events and they show how affected users (that are physically close to the event source) will be notified based on their actual geolocation.

The high majority of related literature approaches Smart Campuses from an Internet of Things (IoT) point of view, having a focus on the physical campus infrastructure including sensor networks built into buildings, physical entities and places (e.g., parking). This is promising from the energy efficiency point of view but basically provides the same challenges as Smart City projects do. As we were looking for a bit different kind of challenges, we decided to turn to the social factors more than infrastructural. Besides finding the need for integration of data coming from sensor networks very important we think that non-infrastructural aspects should play an equally important role in a life of a Campus. Therefore, people living, studying and working on the Campus should heavily be involved so we feel that exploitation of the crowd's power has a crucial effect on the success of any Smart Campus-related projects.

3.2 Crowdsourcing

The term crowdsourcing is not easy to define. It is a hybrid of "crowd" and "outsourcing", meaning that a given task is assigned to a number of people working on it explicitly. This approach is too restricting, as it is coined in (Doan et al., 2011), therefore they give the following definition: "a system is a crowdsourcing (CS) system if it enlists a crowd of humans to help solve a problem defined by the system owners, and if in doing so, it addresses the following four fundamental challenges: How to recruit and retain users? What contributions can users make? How to combine user contributions to solve the target problem? How to evaluate users and their contributions?"

Starting from these challenges, (Doan et al., 2011) also gives a classification of CS systems. Nine dimensions are identified but here we only concentrate on those that are relevant for us. Based on the nature of collaboration, CS systems can be either explicit or implicit. Both are important for us as explicit systems include evaluating (rate courses, meals, etc.), sharing (location info, files, etc.) and networking (adding new friends or classmates) while implicit systems might be either standalone (when event prediction is based on the history of users' activities) or piggyback (for example, based on the trajectories of a user's movement, predictions can be made). In the context of Smart Campus applications, for the first fundamental challenge (recruiting users) only voluntary participation is possible. For more classes of CS systems with detailed explanations, see (Doan et al., 2011).

3.3 Graph Databases

Graph databases are gaining popularity in the past few years as they provide adjacency structures for data elements without having any indices (i.e., data elements contain direct pointers showing their adjacent elements). The three basic building blocks of graph databases are *nodes* (representing entities), *properties* (pertinent information related to nodes) and *edges* (connecting nodes to nodes using directed arcs). According to the Property Graph Model (Robinson et al., 2013, p. 4), nodes contain properties that are key—value pairs, and edges (a.k.a. relationships) can also have properties. This organization of data allows them to be processed using the well-known graph algorithms.

3.3.1 Advantages of Graph Databases

According to (Robinson et al., 2013), graph databases offer three key advantages:

Performance. Graph databases are easy to scale since queries are localized to a portion of the graph only therefore query execution time is proportional only to the size of the affected sub-graph rather than the size of the overall graph.

Flexibility. Graphs are "naturally additive, meaning we can add new kinds of relationships, new nodes, and new subgraphs to an existing structure without disturbing existing queries and application

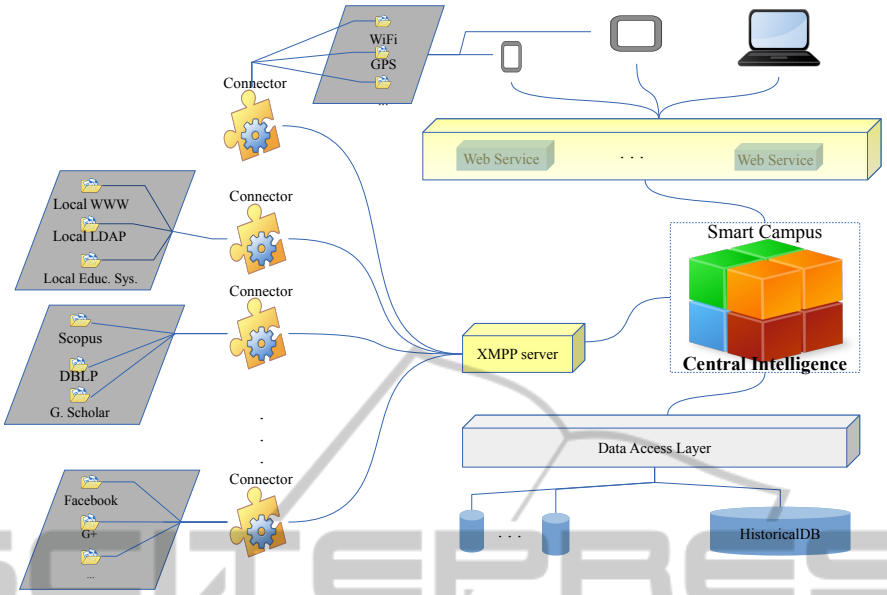

Figure 3: Our extensible architecture.

functionality. These things have generally positive implications for developer productivity and project risk. Because of the graph models flexibility, we do not have to model our domain in exhaustive detail ahead of timea practice that is all but foolhardy in the face of changing business requirements."

Agility. Since the graph data model has a schema-free nature, graph databases suit well for iterative incremental style software development practices. Even the evolution of the data model can be performed in small, agile steps which is a huge advantage when talking about data integration. One can easily add new properties caused by emerging data sources to the existing data model without the need of performing complex schema evolution steps (which are, in fact, probably the biggest challenge when using relational databases).

The graph data model fits well to the data structure of many of today's popular social networking sites inluding Facebook, Twitter, or LinkedIn. It is a definitive advantage over relational databases, key–value stores, and XML databases when our goal is to integrate data from such sources.

4 EXTENSIBLE DATA MANAGEMENT ARCHITECTURE

One of the primary objectives of our system is to capture continuous data streams arriving from different sources. As we discussed in Section 1, Extensible Messaging and Presence Protocol (XMPP) protocol, defining a standardized way of event-based messaging, has been selected as the underlying communication protocol (Szabó and Farkas, 2013b), due to its extensibility and publish/subscribe model. Based on the heterogeneity of the data we should prepare proper input channels for them.

Our proposed architecture (see Figure 3) introduces the term *Connector* which is a node responsible for collecting data from given sources and send updates to the Smart Campus. The notification—and data transfer—is achieved by sending a special XMPP message to the Smart Campus Collector Interface (as shown in Figure 4). Using this information it can be decided whether to approve or decline any updates based on the gathered information. The repository update is declined when a Connector sends a redundant piece of information, for instance, when the Research Connector integrating several sources like DBLP, Scopus, Google Scholar, etc. realizes that it collects the same piece of data that has already been found earlier.

The Connectors in our proposal are crowdsourced compound entities without any deep business logic. They containing the necessary parser or API implementation for the proper sources for a given field—like a parser for Google Scholar and an API for Scopus. It can be imagined as an open box which can be extended (for existing domains) or created and attached to the system as a brand new one—as a new and previously not existed field. This is the power of

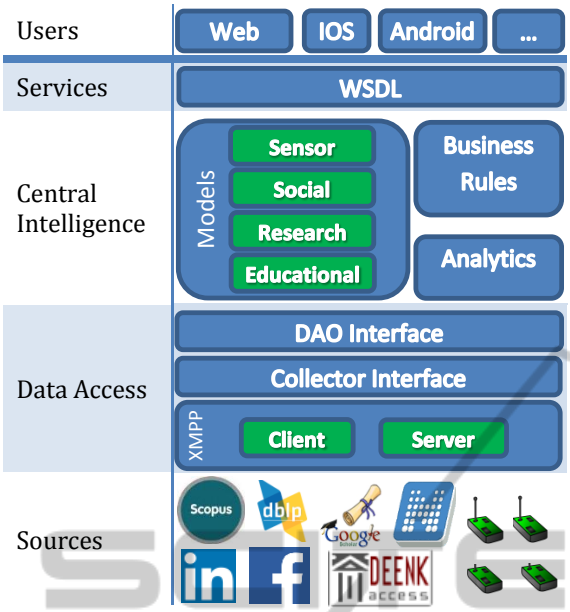

Figure 4: Smart Campus layers.

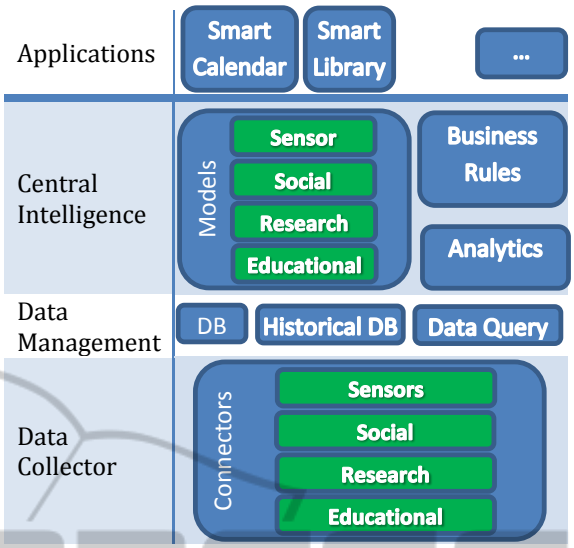

Figure 5: Smart Campus services.

the architecture because it can be done by the crowd. The only requirement is to prepare the Connector for the proper XMPP message structure. If the XMPP server has not got the required message type the extension mechanism of the XMPP comes into the picture. It can be created, inserted and published to the server.

Figure 4 shows a layered view of our Smart Campus architecture. The bottom layer shows the various (and incomplete) sources that are used for gathering data.

The Data Access Layer also plays an important role in our architecture. It provides quick access to the more-or-less statical personal data and to the "fresh" real-time data. It means, the frequently changing data (like sensor data, news feed) can be easily accessed only for a limited time, e.g., three or six months and after all it goes to the HistoricalDB part. This separation can speed up the queries targeted for "fresh" data because the data store cannot grow to a huge one slowing down all the searches. However, the other part, the historical one introduces new challenges as well. At one hand, efficient handling of big data. On the other hand, proper support for the Analytics module to find new and relevant information from that huge amount of data. Since our original goal was to design a data management platform, development of the Analytics module has been postponed until a subsequent iteration.

The Smart Campus Central Intelligence (SCCI) component in our architecture provides an interface between the information sources including both the incoming events (XMPP server) and the information stored in the database and the Web services layer (Figure 4). A service-oriented view of our architecture is shown in Figure 5. The Data Collector layer uses Connectors to connect sources for receiving data. The collected data is then used to populate the databases of the Data Management layer. Applications that are created based on Web services are at the top of the view.

The underlying data model for Smart Campus also highlights challenges. We need to face structured and unstructured data as well. Moreover, it needs to be an extensible model which leads us to a semi-structured model where we know it does not fit into a uniform, one-size-fits-all, rigid relational schema. Faced with the need to generate ever-greater insight and value-added services we turned to graph technologies to tackle the complexity. Besides increasing volume of data, another force which has to be contended with is that over time, data will become more and more unstructured. As data volumes grow, we trade insight for uniformity; the more data we gather about a group of entities, the more that data is likely to be semi-structured but this the way what we originally imagine, an extensible system. But insight and end-user value do not simply result from collecting large volume and variation from data. Insight depends on understanding the relationships between entities, so we need to map how the entities in our domain are connected. The key thing about such a model is that it makes relations first-class citizens of the data, rather than treating them as only metadata. With this approach we established a data store which can be extended in an easy way. When a new—and previously

not existed—source is attached to the system we only need to add the new nodes and proper edges to the database.

5 CONCLUSIONS

In this paper we briefly described a data management architecture for Smart Campus applications. This architecture fits well into the more general publish/subscribe based architecture of Smart City and Smart Campus applications. The developed architecture is extensible with new data sources (appropriate Connectors need to be developed when adding a new source) providing the capability of integration of heterogeneous data. Our future plans include to develop more applications based on that architecture along with the addition of the Analytics module.

ACKNOWLEDGEMENTS

The publication was supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

REFERENCES

Boran, A., Bedini, I., Matheus, C. J., Patel-Schneider, P. F., and Keeney, J. (2011). A smart campus prototype for demonstrating the semantic integration of heterogeneous data. In Rudolph, S. and Gutierrez, C., editors, *RR*, volume 6902 of *Lecture Notes in Computer Science*, pages 238–243. Springer.

Doan, A., Ramakrishnan, R., and Halevy, A. Y. (2011). Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96.

Luo, X. (2012). Social network-based media sharing in the ubiquitous environment: Technologies and applications. In Abraham, A. and Hassanien, A.-E., editors, *Computational Social Networks*, pages 349–366. Springer London.

Robinson, I., Webber, J., and Eifrem, E. (2013). *Graph Databases*. O'Reilly Media.

Saint-Andre, P. (2011). RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core.

Szabó, R., Farkas, K., Ispány, M., Benczúr, A. A., Bátfai, N., Jeszenszky, P., Laki, S., Vágner, A., Kollár, L., Sidló, C., Besenczi, R., Smajda, M., Kövér, G., Szincsák, T., Kádek, T., Kósa, M., Adamkó, A., Lendák, I., Wiandt, B., Tomás, T., Nagy, Á., and Fehér, G. (2013). Framework for smart city applications based on participatory sensing. In *Proceedings of the* 4^{th} *IEEE International Conference on Cognitive Infocommunications*. to appear.

Szabó, R. L. and Farkas, K. (2013a). A publish-subscribe scheme based open architecture for crowd-sourcing. In Bauschert, T., editor, *EUNICE*, volume 8115 of *Lecture Notes in Computer Science*, pages 287–291. Springer.

Szabó, R. L. and Farkas, K. (2013b). Publish/subscribe communication for crowd-sourcing based smart city applications. In *Proceedings in Conference of Informatics and Management Sciences*, pages 314–318.

Valkanas, G. and Gunopulos, D. (2013). Event detection from social media data. *IEEE Data Eng. Bull.*, 36(3):51–58.