

Data Literacy in the Smart University approach

Christian S. Sauer Antonio D. Kheirkhahzadeh
christian.sauer@uwl.ac.uk antonio.kheirkhahzadeh@uwl.ac.uk
Thomas Roth-Berghofer
thomas.roth-berghofer@uwl.ac.uk

School of Computing and Engineering
University of West London, United Kingdom

Abstract

Equipping classrooms with inexpensive sensors for data collection can provide students and teachers with the opportunity to interact with the classroom in a smart way. In this paper two approaches to acquiring contextual data from a classroom environment are presented. We further present our approach to analysing the collected room usage data on site, using low cost single board computer, such as a Raspberry Pi and Arduino units, performing a significant part of the data analysis on-site. We demonstrate how the usage data was used to model specific room usage situation as cases in a Case-based reasoning (CBR) system. The room usage data was then integrated in a room recommender system, reasoning on the formalised usage data, allowing for a convenient and intuitive end user experience based on the collected raw sensor data. Having implemented and tested our approaches we are currently investigating the possibility of using (XML)Schema-informed compression to enhance the security and efficiency of the transmission of a large number of sensor reports generated by interpreting the raw data on-site, to our central data sink. We are investigating this new approach to usage data transmission as we are aiming to integrate our on-going work into our vision of the Smart University to ensure and enhance the Smart University's data literacy.

1 Introduction

The students and lecturers at our institution, the University of West London (UWL), currently work in a static environment with regard to the allocation of rooms for teaching facilities. The allocation of rooms is currently based on a centrally managed room scheduling system that only takes into account technical necessities such as number of people in a room and for example available computers in a room. Complaints, for example, from students about their studying experience to their lecturers or lecturers' complaints about their teaching environment are currently not very well supported by evidence. Additionally the current system is too slow in case the need appears to change room allocation 'on the spot' for example if a laboratory has a technical problem. Furthermore the system is currently not able to 'advertise' rooms that became available by unforeseen events such as the cancellation of a lecture.

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: A. Editor, B. Coeditor (eds.): Proceedings of the XYZ Workshop, Location, Country, DD-MMM-YYYY, published at <http://ceur-ws.org>

The approach described in this paper aims to improve the room allocation system at the UWL by the use of contextualised sensor data within an online room recommender system. The proposed system itself is a prototype component of our vision of the Smart University approach.

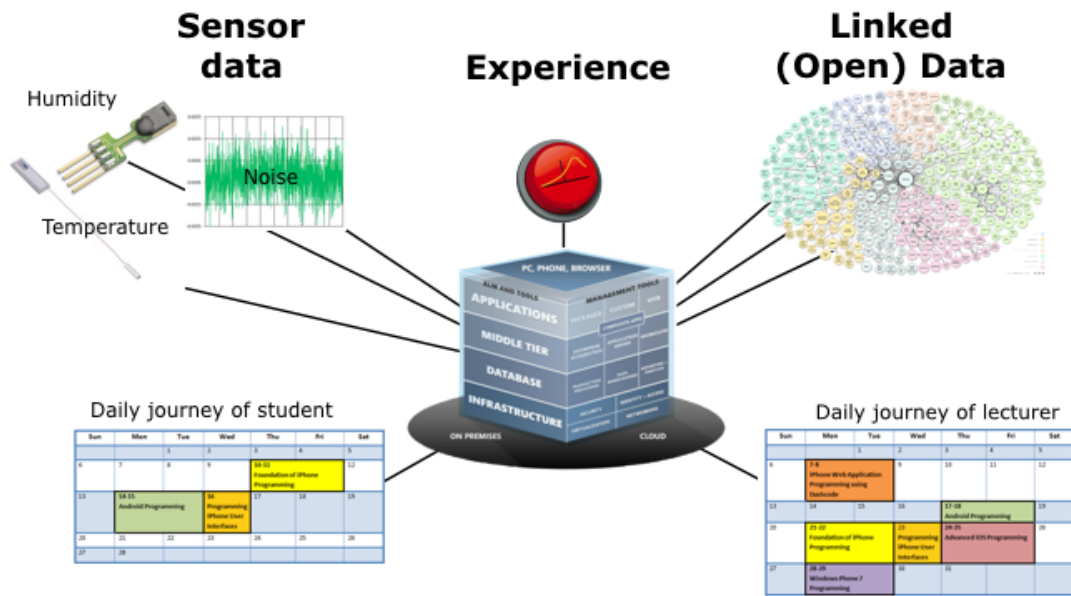


Figure 1: Smart University vision [Rot16]

Within our vision of the Smart University the university is seen as a platform that acquires and delivers foundational data to drive the analysis and improvement of the teaching and learning environment (see Figure 1). Sensor-data is joined with linked (open) data (LOD), and formalised teaching knowledge form the source data for the platform. From laboratories and lecture theatres equipped with low cost sensors the platform gathers room usage data and additionally employs single board computers in situ to process the context of the room usage data, such as retrieving the kind of teaching activity taking place during the sensor recording. The platform employs case based Reasoning (CBR) [Aam94] to formalise the captured room usage data in context and re-use it for room recommendations and room allocation planning. Furthermore the Smart University platform allows linking the daily journey of lecturers and students by accessing students and lecturer’s online calendars to include the data in personalised room recommendation and allocation [Rot16].

This paper shows two implemented approaches with which we are investigating the possible problems and solutions with regard to the Smart University’s data literacy. We investigated the sub problems of data collection, data analysis, data communication and finally use of the analysed data in a room recommendation process. Both prototypes were successful in covering the whole spectrum of data literacy from the collection of raw sensor data to the use of results from on-site data analysis in a central room recommendation system that allowed for easy interaction of the users with the collected and processed data. Both prototypes also demonstrate that we were able to integrate the rather different data sets of sensor event data with, for example, time tabling information provided by a University.

Our research focused firstly on the realisation and evaluation of the sensor units to address the first step in data literacy, data collection. Secondly we introduce the use of on-site low cost computers to address the task of data analysis in the form of generating reports of room usage in XML. We then investigate the aspect of data communication by the use of XML Schema based compression for the XML-based room usage reports to secure and compress them before they are send to a central data sink. As an implementation of the central data sink, a CBR system was implemented that can retrieve room usage data on specific room situations, allowing for an intuitive use of the analysed data by the systems end users, namely students and lecturers. The purpose of the CBR system is to act as a room recommender system that recommends the most suitable rooms for students and teachers based on previous recorded data acquired from the sensor units. The room recommender system also incorporates feedback on the suitability of rooms for specific teaching activities, gathered from surveys filled out by students, regarding the most comfortable environment for specific teaching activities such as seminar,

lectures or quiet study. The room recommender thus acts as a context-aware system [Zha10], as the different situational contexts of different teaching activities are taken into account.

The rest of this paper is structured as follows: In section 2 we present the objectives of our research. Section 3 then details on the methodology we employed for our research. In section 4 we then present the technological implementation of the two room sensor network approaches and the room recommendation systems main reasoning approach CBR. We also detail on the approaches chosen for the formalisation of the sensor data and the on-site generation of room usage estimation reports. Section 5 then details on the implementation of two approaches to provide data literacy for the Smart University platform. The results from experiments with these approaches are then presented in section 6 of the paper. In section 7 we introduce our new approach to employ XML Schema based compression to address the necessary data communication in our system when communicating the on-site generated reports to our central data sink. In the final section we summarise our work and detail on future work on the Smart University platform.

2 Objectives

The objectives of our research were to build a sensor unit that can acquire contextual data from a classroom environment. The intention of the sensor unit is to gather data, such as noise level, motion, temperature and humidity. The main goal pursued by these measurements is to establish the air quality and overall usefulness of a particular room at any given time, to be able to select rooms most suitable for teaching and thus increase the quality of the teaching and learning experience for both, the students and the lecturers. We therefore aimed to establish the desirable physical environment with regard to the dimensions of temperature, humidity, noise level and motion level, for specific teaching activities by surveying the students at UWL. The on-site measurements were made by an array of selected sensors that were controlled by low cost single board computer, being either a Raspberry PI or an Arduino. These simple, on-site, PC's gathered the raw sensor data and generated room usage estimation reports which were written in XML. These reports then were send to a central data sink, either via UHF radio or via the Internet. This objective required us to investigate different approaches to the formalisation of the gathered room usage data in the context of the respective teaching activity, including the feedback gathered from the students. To capture this feedback we created and performed surveys with the students on their room usage experiences which we then combined with the sensor data from the sensor units. Based on a suitable approach to the formal representation of the combined sensor and survey data we then aimed to build a CBR knowledge model for the room recommender system. For the purpose of building the CBR knowledge model we had to particularly investigate suitable similarity measures (please see section 3), to enable the knowledge model to compare room usage data. To evaluate our system we had to perform a substantial number of retrieval experiments in the form of getting room recommendations from the system and evaluate the accuracy and quality of these recommendations.

3 Methodology

Based on an initial empirical study [Wal05] of existing work in the utilisation of sensors to monitor room usage situations, context-aware computing, CBR, knowledge formalisation and the design of recommender systems, we designed two approaches to create our own sensor units and integrate them into the Smart University platform.

The implementation of the sensor units and their software followed the Action Research approach [Avi99]. The implementation of the necessary software for the sensor units followed the iterative development approach, starting off with basic prototype software and adding desired features in consecutive implementation cycles.

The design and implementation of the CBR knowledge model and its similarity measures followed the rapid prototyping approach. We employed our own CBR knowledge modelling software myCBR¹ to prototype the knowledge model for the room recommender system.

The evaluation of the built sensor units was performed by quantitative analysis of the sensors data gathered. The accuracy and performance of the room recommender system was evaluated by conducting recommendation experiments and performing qualitative and quantitative analysis on the data gathered from these experiments.

The qualitative feedback data we incorporated in our knowledge model was gathered by surveying students taking part in teaching activities that were monitored by our sensor units and combining the sensor data with the qualitative feedback on the study experience from the students.

¹<http://www.mycbr-project.net/>

4 Technology used

As mentioned the recommendation of rooms to students or lecturers in our scenario is based on re-using captured and on-site evaluated room usage data, combined with qualitative feedback from the users of a specific room/teaching activity. The basic idea is to recommend suitable rooms for a specific teaching activities with regard to necessities, such as available computers, number of peoples the room is need to accommodate and also captured feedback how ‘pleasant’ the experience of former students were in a room in the context of a particular teaching activity.

4.1 Case-based room recommendation

The key technology we employed to recommend rooms for specific teaching activities is to ‘compare’ the sensor data and user feedback from former uses of rooms with the current query for a room to recommend the best suited room. To be able to compare recorded room usage experiences with a current query we employ a specific Artificial Intelligence (AI) technique called case-based reasoning (CBR) [Wat99].

A central concept in CBR is the case, where a case consists of a problem description and a solution description. Therefore a case represents recorded knowledge of an encountered problem and the solution that solved said problem. In the context of our work a case’s problem description is describing a specific teaching activity and the necessities of a room, like number of peoples to accommodate and, for example, computers being available. The solution to this problem description is a room that is known to have been suitable and pleasant for past, similar, teaching activities. Additionally the solution description also contains advice on how to improve the room condition for a specific query, for example by advising to ventilate the room if the air quality is poorly matched. Therefore our system tries to match a query for a room witch recorded, past, queries and re-use the most suitable solution, in our case a room recommendation, to solve the query.

The basic process of the matching of a current problem case to recorded cases to retrieve a best matching case and re-use its solution is described by the CBR cycle. The four steps in the CBR cycle are:

- 1. Retrieve:** In this step, one or more cases similar to the current room query are retrieved from the case-base, storing all recorded room usage cases. that are similar to the current problem description from the case base. The matching of the query to the recorded room usage cases is performed by calculating the similarity of the problem descriptions, using similarity measures.

- 2. Reuse:** This step re-uses the solution, or room recommendation, from the retrieved case that is most similar to the current room query. If required, the proposed solution is adapted to fit the current problem [Aam94].

- 3. Revise:** In this step, the proposed solution is tested by evaluating the room recommendation and revised if necessary.

- 4. Retain:** In the final step of the cycle the applied, successful, room recommendation, together with the room query, is kept as a new case. Therefore the new query and the (possibly adapted) successful room recommendation is retained to be reused to solve a similar problem in the future [Ric13].

4.2 Gathering usage data

We developed two approaches to the gathering of sensor data. The first approach only gathered motion data, using an infrared (IR) motion detection sensor connected to a low cost single board computer. A second approach we implemented was more sophisticated and used sensors to capture data on temperature, motion, humidity and noise level inside a room. Again these sensors were controlled by a low cost single board computer. Both approaches communicated the sensor data to a central data sink. Both approaches employed low cost single board computers to control the sensors, with regard to sensing intervals and to pre-process the raw sensor data captured. This pre-processing basically aimed to derive a contextualise usage situation description for the room, combining the sensor data with other data such as date, time, location and additional data, for example from the online lecture schedule. The first approach employing XML to generate and communicate estimates of the room usage situation that were then send to the central data sink. Therefore the first approach employed UHF radio communication mesh networks to communicate usage data estimate reports, while the second approach worked upon existing Wi-Fi networks. The advantage of the radio based approach was the applicability of the sensor networks in areas where no Wi-Fi infrastructure is present. However both approaches are also in need of encrypting and compressing the room reports before sending them by radio or internet. This is necessary for data protection reasons as well as for performance and efficiency reasons.

Qualitative feedback on particular room usage situations was gathered from student volunteers. The students gave qualitative feedback on their learning experience in selected room/teaching activity constellations. This

feedback was then used as a quality measure in the room recommendations to enable the system to recommend rooms that are not only, technically, suitable for a specific teaching activity but also offer a ‘pleasant’ experience.

4.3 Knowledge formalisation

To be able to recommend rooms using a CBR-based recommender system we had to represent the gathered room usage data and the quality feedback from the students in a suitable knowledge model. We employed our own CBR development software, myCBR 3.0 to create the initial knowledge model and refine it, based on retrieval (room recommendation) tests. The basic aspects of the knowledge model we defined were the domain vocabulary, such as the concept room and its attributes and the value ranges of these attributes and the similarity measures needed to compare values of these attributes to calculate their similarity [Len98]. The calculation of these similarities were distance-based and resulted in a specific value from the normalised interval [0, 1] where 0 stand for totally dissimilar and 1 for identical values. So for example a similarity value of 0.8 still represents a fairly similar pair of values, whilst a similarity of 0.15 describes already quite dissimilar values for an attribute. We additionally modelled a global similarity measure to compute the overall similarity of a query’s problem description to the problem descriptions in all stored room usage cases. We employed mathematical distance functions, comparative tables and taxonomies as similarity measures.

5 Developments

This section describes the design of both systems built to serve as the room recommendation facility within the Smart University approach. The first approach employed Raspberry Pi (R-Pi) computers to handle up to three motion sensor units. The sensor data was pre-processed by the R-Pi computers in that way that based on quantitative analysis an estimate of a room’s usage situation was derived. This estimate was than enriched with date, time and location information and formatted in a report in XML that was sent to a general data sink via UHF radio signals.

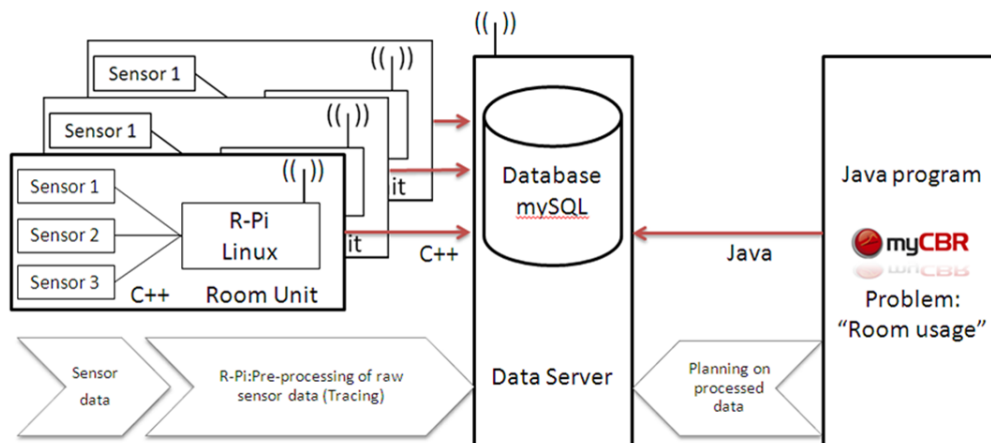


Figure 2: Our first approach to sensor data based room recommendation

The central data sink was a MySQL database that could be accessed by a myCBR system embedded in a Java-based recommendation software (Figure 2). This approach was developed up until the stage involving the storage of processed room usage estimates in the central data sink. The actual use of the data for room recommendation was not implemented as we decided to follow a more sophisticated approach, our second approach, before we would start to model the CBR knowledge model. However the sensor units, their estimation of room situations and the communication of these estimates in XML format send of UHF radio worked in a satisfactory way.

Based on insights gained from our first, simplified approach, we developed a more sophisticated approach to the problem of sensor data based room recommendation (Figure 3). This second approach was based on Arduino single board computers [Ban11] that controlled an array of different sensors. The sensor units of our second approach were able to sense a room’s temperature, humidity, noise level and motion events in the room. Also different to our first approach our second approach employed Wi-Fi rather than UHF radio to communicate the room usage estimates. The estimates themselves were encoded into Comma-separated-value (CSV) files. This

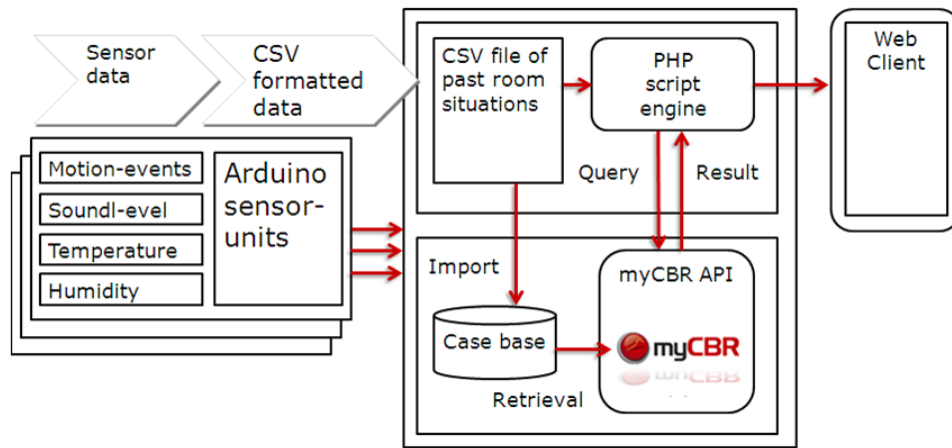


Figure 3: Our second approach to sensor data based room recommendation

encoding in CSV format allowed for the direct import of the room usage estimates as cases into the case-base of our CBR system. As we were satisfied with the richness of the data we gathered from the sensor units we proceeded to formalise the data into our CBR knowledge model and set up a website to access the room usage data in real time as well as to incorporate our CBR-based room recommendation system.

It is worth to mention that both our approaches were prototype implementations based on available hardware for experimentation with sensors, such as the Raspberry Pi computers and Arduino boards. In an industrial scale application of our conceptual work the sensor units would, of course, be replaced custom built hardware. Assuming the necessary funding for a pilot system in a higher education institution, the only additional need for our system to be scaled to production level would be a normal, single, database server. Depending on the time needed to create these tailored hardware units we estimate that, as the underlying technologies is already implemented and tested, our approach could reach the market within 8 to 12 months.

6 Results from the prototype evaluations

To evaluate our approaches we performed experiments with both of them. Within our first approach we performed experiments by placing sensor units in 3 classrooms and offices. The sensor units performed satisfactory and the situation estimates for each room were communicated to the central data sink via radio without interference. However we noticed that in steel frame buildings the radio signal of the sensor units was diminishing quickly and thus the range of the sensor units communication was reduced significantly. We assume that this problem can be solved by the use of the ability of the sensor units to create mesh networks and communicate the room situation estimates of more remote sensor units through this network.

With regard to the data gathered from the sensor units we performed a quantitative analysis on the gathered data. Based on this simple analysis we were able to establish that already one motion sensor with a measuring interval of 1 minute gathered enough data to allow for a reasonably accurate estimate of a rooms usage. We based this estimate on the count of motion events triggered by the sensor per minute. See Figure 4 for an example of the data from one sensor unit.

To evaluate our second approach we performed retrieval (recommendation) experiments and analysed the sensor data. An initial important point that we noticed was the data volume. As we stored al contextual data acquired by different sensors in CSV format, so that it can be directly imported into our CBR system. The format allows rapid generation of cases from its records. However, this file format is not the best one to use when data volume becomes massive. If a large number of sensor units are deployed in different locations, a massive number of records are destined to be stored. For efficient handling of this data in a later stage of our system we will therefore employ Hadoop.

With regard to the actual retrieval (recommendation) of rooms we performed retrieval experiments upon a case base that consisted of 2,452 instances of classroom environments. It is worthwhile to mention that this number is not the number of classrooms monitored but rather resulted from the monitoring of several rooms over time. The results from these retrieval experiments were analysed in a qualitative way, mainly in the form of user feedback, were a student or lecturer posted a room query to the system and was asked of his/her estimate of the

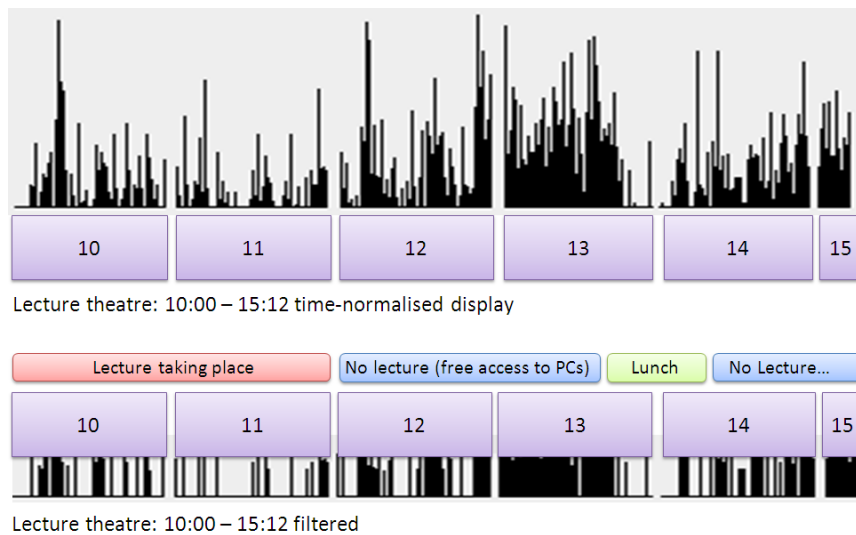


Figure 4: Initial quantitative analysis of motion events recorded by our first implementation

recommendations accuracy and quality, e.g. if the person agreed that the system recommended a suitable room. A convenient majority of the feedback indicated a satisfactory quality of our systems room recommendations. Furthermore, to determine the most desirable physical environment in a classroom in terms of temperature, humidity and air quality, surveys were conducted at different locations at different times in the University.

7 Refining the use of XML to facilitate data analysis from the sensor units

Having implemented and tested the prototypes described we are currently investigating a new approach to address the need to encrypt and compress the room usage reports that are generated from the sensor data on-site before sending the reports to the central data sink. The necessity for this arises from the need for data protections as well as from the aim to have the on-site low cost computers work more efficient and reduce the network traffic within the sensor-unit, data sink network. Especially in the first prototype, which used UHF radio to transmit the usage reports to the data sink, it was noticed that, when scaling the number of rooms (room units), the network traffic increased so much that it became desirable to have the XML files (the room usage report, see Figure 5) compressed before they are transmitted to the central data sink. To address this need for encryption and compression we are now investigating to employ a new approach, the use of Schema-informed compression to encrypt and compress the XML files generated by the on-site low cost PCs before their transmissions.

```

<report from unit 21>
<kindofreport>Minute</kindofreport>
<reportID>1182</reportID>
<time>10:29:12
</time>
<rawData>
<#MotionEvents>0</#MotionEvents>
</rawData>
<estimate>
<occupied>0</occupied>
<estimateofsituation>Room is vaccant</estimateofsituation>
</estimate>
</report from unit 21>

```

Figure 5: A snippet from an XML room usage report generated by the sensor units by our first prototype

7.1 XML compression

XML is a standard for data storage and exchange over the internet. It has the ability to represent structured data in a human and computer readable format and provide support for Unicode [W3C94]. One of the main features of XML is the ability to create user defined structures to wrap information using markup language. As the popularity of XML continues to grow, the demand for processing and storing has increased leading to a number of interesting results [Ari07, Bon06, Che06]. Software capable of handling XML data efficiently led to many advantages:

1. minimise the size of the data to reduce disk space and network bandwidth required to exchange XML
2. reduce the memory consumption required to process XML using efficient parser
3. efficiently querying both compressed and uncompressed XML data

XML compression refers to techniques and algorithms that take advantage of the verbose and redundant structure of XML to improve the compressed size. Because of the nature of XML, most compression techniques focus on loss-less compression algorithms. However, lossy and near-loss-less techniques have also produced interesting results [Can01]. Traditional compression techniques can be successfully used to compress XML data. However, an XML-*conscious* technique is able to compress beyond traditional text compressors. In addition, knowledge of the XML structure allows techniques based on prediction and statistical algorithms to apply better compression.

7.2 Using Schema-informed compression

To be able to describe the data that exists in a file containing markup, we also need some meta-data which we will call a schema. If we need to communicate our data between parties A and B, in this case between the on-site PC and the central data sink, party A, the on-site PC, can compress the data using the schema and send it to party B, the central data sink, who can decompress it, providing it has the same schema. This scenario works well for compressing network communication as the schema defines the contents of the network protocol and can be used to define complex protocols. However, if we try and use this approach to compress a document we need to store the schema along with the compressed markup, which, however in our scenario wasn't a problem. For network communication we can define a schema ahead of time and it can be written by a human. However, for document compression we must automatically generate a suitable schema at runtime for the data we are trying to compress. This is not a problem as the on-site PCs are able to do so. As the necessity for using Schema-informed compressions are met by the capacities of our on-site units and the central data sink we are currently implementing the use of Schema-informed compression and preliminary results are encouraging with regards to bot, the rate of compression achieved as well as the security of the encryption delivered by the use of schema to compress test data.

8 Summary

In this paper we proposed two approaches to the use of sensor data to improve the classroom experience of students in a higher education institution. We did so to demonstrate two ways to provide data literacy to our Smart University approach. We introduced the conceptual approaches to address the sub problems of data literacy, ranging from data collection to data analysis and data re-use. We detailed on the technical implementation of our approaches addressing these sub problems. We then described the evaluation we performed on both approaches we implemented to evaluate the data literacy of the subsequent overall room recommendation system. We further described our current investigation of using Schema-informed compression to compress and encrypt the room usage reports, generated on-site, before transmitting them to the central data sink. We did so as we noticed in the evaluation of the two approaches that the sub problem of data communication needed to be addressed before we can scale the systems further.

As we have stated in the paper the technologies necessary for ensuring the data literacy of our approaches are implemented and tested with good results, therefore we assume our approaches to be of significant interest for industrial application mainly in the higher education area but not limited to this market as major corporations with larger buildings may also be interested in our approach. However, during the evaluation it was also noted that with scaling our approaches we also needed to address the sub problem of data communication, mainly the

problem of increased network traffic and secure communication. To address these we investigated the application of Schema-informed compression with, although preliminary, very encouraging results.

For our immediate future work we focus on the data communication sub problem by integrating the schema-informed compression into the room-units. We want to focus on re-implementing the UHF radio communication from our first approach to enable the Schema-informed compression to reduce the radio traffic. Furthermore we want enhance our solution to the data collection sub problem by adding more sensors (e.g., CO2 detection sensor, light detection sensor) to the systems. Finally we are investigating the use of the X10 protocol to control devices (e.g., heating) in a classroom, based on the room usage estimates established by our prototype system, therefore integrating further data sets into our system.

References

- [Aam94] A. Aamodt. E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches *Artificial Intelligence Communications*, 7:39–59, 1994.
- [Ari07] A. Arion. XQueC: A query-conscious compressed XML database *ACM Transactions on Internet Technology (TOIT)*, 7:2:10, 2007.
- [Avi99] D. Avison. et al. Action research *Communications of the ACM*, 42:1: 94–97, 1999.
- [Ban11] M. Banzi. Getting started with Arduino *O’Reilly Media, Inc.*, 2011.
- [Bon06] P. Boncz. et al. MonetDB/XQuery: a fast XQuery processor powered by a relational engine *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, 479–490, 2006.
- [Can01] M. Cannataro. et al. Semantic Lossy Compression of XML Data. *KRDB*, 2001.
- [Che06] J. Cheney. Tradeoffs in XML compression *Proceedings of the 2006 IEEE Data Compression Conference (DCC 2006)*, 392–401, 2006.
- [Len98] M. Lenz. et al. Case-Based Reasoning Technology: From Foundations to Applications *Lecture Notes in Artificial Intelligence*, 1400, 1998.
- [Ric13] M.M. Richter. R.O. Weber. Case-based reasoning: a textbook *Springer Berlin*, 2013.
- [Rot16] T. Roth-Berghofer. C.S. Sauer <https://smartuniversity.uwl.ac.uk/blog/>. Accessed: 2016-22-01
- [W3C94] W3C Extensible Markup Language (XML) <http://www.w3.org/XML>, 1996.
- [Wal05] N. Walliman. Your research project: a step-by-step guide for the first-time researcher *Sage*, 2005.
- [Wat99] I. Watson. Case-based reasoning is a methodology not a technology *Knowledge-Based Systems*, 12:5:303–308, 1999.
- [Zha10] D. Thang. et al. Context reasoning using extended evidence theory in pervasive computing environments *Future Generation Computer Systems*, 26:2:207–216, 2010.