Exploring The Design of a Memory Model for Smart Objects

Daniel Fitton, Fahim Kawsar and Gerd Kortuem
Computing Department, Infolab21, Lancaster University, Lancaster, UK.

Abstract. This paper presents an exploration of the design of a memory model to support the management of persistent historical memories recorded by a smart work object. Analysis of a range of potential application categories and scenarios involving a smart work object is used to highlight the requirements and different characteristics of digital memories. The analysis is then used to identify a range of pertinent issues and trade-offs which are used to inform the design of a generic parameterized memory model. A case study involving a smart object prototype in a workplace application scenario is then presented. The case study then analyzes how the proposed memory model can be applied to memories collected by the prototype.

Keywords. Smart Object, Memory Model, Pay-Per-Use

Introduction

The storage and management of the digital memories of humans (images, videos, e-mails, documents and so forth) is an ongoing research topic [1][2] and has been recognized as a 'Grand Challenge' for computer science [3]. This paper is concerned with the emerging area of digital memories produced by smart objects. Smart objects are everyday physical objects augmented with embedded technology that include sensing, processing, communication and persistent storage. The memories generated by a smart object are often related to sensing/context awareness and, through their analysis, enable a wide range of new and novel applications. For example, existing applications have involved health and safety monitoring [4] and the support of new business models [5] within the same domain.

The focus of this paper is the design of a memory model to support persistent historical memories recorded by smart objects in the workplace. A memory model may have to manage the recording of a multitude of different memories to support different applications. For example, memories of its usage, its service history, its location etc. These memories may vary in level of detail, importance and granularity. A memory model must also take action when storage limits are reached. While some objects may remain indoors with plentiful network connectivity and access to backend infrastructure others, such as the smart tool discussed in this work, may spend long periods outdoors with no network connectivity or access to backend infrastructure. Therefore the memory model must take action when storage limits are reached and three main approaches, each with unique problems and trade-offs, are discussed in this work. While this work focuses on a memory model for smart work objects we also consider how our finding can be applied in a more generic smart object memory model.

The following section discusses application scenarios that involve smart objects with persistent memories. Next requirements highlighted by the scenarios and characteristics of object memories are considered. The design of the memory model is then presented followed by a case study involving a prototype smart object together with analysis of the memories recorded in the context of the memory model. Finally related work and concluding remarks are presented.

Smart Objects Applications

A wide range of potential applications exist for the smart objects which maintain a persistent history of their memories. Memories include an object's *experiences* (events or activities involving the object) and other important pieces of information. The focus in this work is smart work objects (specifically tools and associated equipment) and figure 1 shows four different categories of memories and associated applications.

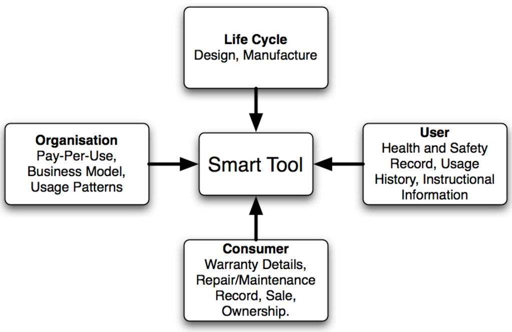

Figure 1. Categories of Experiences Recorded by a Smart Object.

The category of 'Life Cycle' is related to the physical lifecycle of the object from design and manufacture through to disposal and recycling. Storing these memories enables possibilities such as identification of materials used in manufacture in order to inform recycling or re-use in the disposal process. Memories of an object's origins may also be used to validate its authenticity. In the category of 'User' these memories are related to instructional information or legislation (user manuals, health and safety policy etc.) and memories of use of the object. For example, usage memories can be used to check associated risks to a user's health [4]. The 'Consumer' category relates to issues affecting the value or desirability of the object such as a record of repair, maintenance or past owners. The 'Organization' category relates to memories such as object movement to enable analysis of business processes (and whether they need to be revised or redesigned etc). Additionally, new business models are enabled through the use of smart objects such as pay-per-use equipment rental [5].

From this discussion two findings can be summarized:

- Smart object memories are likely to emerge from a range of different sources.
- Historical object memories enable a diverse range of novel applications.

Smart Object Memory Model

In this section the key properties of memories stored by smart objects are identified, followed by a discussion of the design of a smart object memory model.

Smart Object Memories

In order to effectively manage memories it is first necessary to understand their origin and characteristics in order to make appropriate decisions how they should be treated. On differentiation is between external sources where memories are *imprinted* and internal sources where memories are dynamically *generated* through interpretation of embedded sensor data (using predefined algorithms). For example:

- Imprinted: Memories regarding design and manufacture.
- Dynamically Generated: Memories regarding instances of use and misuse.

It is probable that imprinted memories cannot be altered and must either be stored or deleted. However, in the case of dynamically generated memories it may be possible to change the parameters of the algorithm to produce fine-grained memories (high sampling interval, high precision etc) or course-grained memories (low sampling interval, low resolution etc). These extremes represent a trade-offs in terms of detail in data recorded vs. storage space utilization. The actual requirements placed on memories are, of course, application dependent. While a memory model may be required to support multiple applications it is likely the underlying requirements must be known in advance in order to define, then potentially adapt, these algorithms.

The temporal importance of memories may be *short-term* or *long-term* relative to the lifetime of the object. For example, imprinted memories about object manufacture have long-term significance while memories of individual periods of use may only be required for calculating hire cost based on pay-per-use billing in the short term. This is an additional consideration in the design of a memory model.

Managing Memories

When recording memories to support multiple applications there is potential for commonalities and replication in the memories stored, this must be considered in order to make efficient use of storage. For example, when requiring memories of each instance of use and the use total, the summation of the former provides the latter. In this example the memory model could potentially optimize use of storage by only recording memories of use. This is a trade-off between pre-processing (processing required during recording of memories) and post-processing (processing required during retrieval of memories) within the memory model. In more detail, the saving in terms of storage is at the cost of additional processing required to retrieve memories at a later stage. Application requirements would have to be known prior to implementation of the memory model to enable this example.

In an ideal situation memories would be transferred to a backend infrastructure for periodic archival. However, objects such as tools may be in use outdoors for long periods where the deployment of infrastructure is challenging. When no free storage is available and new memories need to be stored three possibilities exist:

- Forgetting - Discarding new memories or overwriting existing ones.
- Abstracting - Combing multiple memories or reducing level of detail.
- Compressing – Compressing memories unlikely to be removed or changed.

Each of these three possibilities offers a trade-off in terms of processing overhead vs. loss of data. Forgetting requires the least overhead and greatest loss and while compression does not incur loss there is significant processing and power overhead. This trade-off is particularly important in a resource constrained embedded system. While the application scenario controls the actual implementation of the model, it is likely that long-lived memories would not be forgotten or abstracted, but may be compressed as a matter of course. The selection of a compression algorithm is likely to involve another trade-off in terms of complexity (and associated processing overhead) vs. size reduction. While the application requirements and memory characteristics dictate the applicability of these three techniques in a given scenario, we would argue that the notions of forgetting, abstracting and compressing provide the foundation of a more generic memory model. The applicability of each techniques could potentially be represented as a set of three parameters for each type of memory.

Figure 2. Smart Drill Prototype.

Case Study: Smart Drill Prototype

This work considers memories in the context of a heavy duty battery powered drill, the prototype implementation of which is shown in figure 2. The prototype contains an ARM7 processor, microSD card slot, 802.15.4 radio, Bluetooth radio and 2-axis accelerometer. The hardware and battery is contained in a small case (see Figure 2, insert). The prototype primarily records memories of its use in order to support a pay-per-use equipment hire model [5]. Each memory requires 20 bytes of storage and includes an identifier of the user (8 bytes), a timestamp (8 bytes) and a use duration (4 bytes). These are short-term memories and will be removed when the hire period ends and the drill is returned to the hire company.

For the application of a pay-per-use model the most important memories are those that containing total usage time. Several solutions for abstracting these memories, such as replacing the individual records with a single use total or removing either the user identifier or timestamp (each freeing 40% of the currently occupied space). Other possibilities include changing the granularity of the memories from single instances of usage to encompass more information. For example, the individual memories could be replaced with single usage total for each user for a specified time period such as a day. Conversely, the memories could be combined in terms of the times of day they occurred, giving a usage total for each time period. The savings gained by latter two possibilities are dependent on the data and the exact algorithms used but potentially could be as high as 80%. Another solution would be to actually execute the pay-per-use billing model on the object, storing the total cost and removing the usage memories instantly. However, in this case it becomes challenging verify that the billing model has been executed correctly or change the model.

Related Work

Most of the existing works on attaching structured digital information to physical objects are limited to tag based approaches. For example, auto-ID[6] technologies like barcode [7], RFID [8][9], QRCode [10], etc. are successfully used in logistics, supply chain management, and healthcare applications. However, these tags are primarily used for object identification and unable to store more fine-grained dynamic contents due to their architectural limitations. Some researchers have investigated a more holistic approach of associating digital information by applying the notion of "Digital Object Memory". Schneider and his colleagues designed memory models revolving around a variety of application scenarios (kitchen, shopping, etc.) and reasoned about the dimensions that influence the model e.g., on-board or off-board storage, software or hardware implementation [11]. On an application level, there are several works that have looked at capturing and sharing everyday experiences using a multitude of personal devices [1][2]. In fact, "Technology for Life Long Memories" has been identified as one of the grand challenges of computer science [3]. However, most of these works have taken an ad-hoc approach in terms of the memory organization. This work is concerned with systematizing this organization using a range of customizable parameters.

Discussion

This paper has categorized memories in several different ways based on their characteristics (imprinted, generated, long-lived, short-lived) in order to consider how each should be treated within a memory model. It is also possible to base categorizations within the memory model upon physical or logical areas of storage to which they are assigned. As storage limits were reached memories could then be moved between these areas forgetting, abstracting and compressing as appropriate.

A key aspect of the memory model is the generation (interpretation) of memories from sensor data. Our goal is for a generic memory model that is device and application independent to allow, for example, the same pay-per-use billing model to be used on different devices in different scenarios. As sensing possibilities and capabilities change between different hardware implementations and scenarios 'pluggable' sensor interpretation algorithms are required. With a generic model in place such a pay-per-use billing model would be primarily based on memories of usage duration but is flexible enough to include detailed memories of usage parameters (intensity, time of day, qualifications of user etc) if available.

The possibility of utilizing compression on smart object memories is an area for future work and requires careful consideration in a resource-constrained embedded scenario. However, existing work has shown that compressing data has the potential to save energy in terms of reducing the overhead of transmitting that data [12] and this reduced overhead may also apply to storage. Additional areas for future work include support of external communication in the memory model to enable access to and imprinting of memories (addressing issues such as privacy, security and authentication). A key area is the mechanism for moving memories from a smart object onto a backend infrastructure. This includes issues such as verification that external memories have been transferred successfully (before removing them from the smart object) and how to maintain some form of 'link' to them.

Concluding Remarks

This paper has considered the design of a generic memory model for smart objects, everyday objects augmented with embedded technology, with a focus on supporting smart work objects in the construction domain. The model is intended to support memories with a range of different characteristics and requirements in order to cater for multiple diverse application scenarios involving a single object. Smart objects typically operate independently and potentially may spend large periods without network connectivity and backend infrastructure (especially in the work object scenario considered in this work). A key aspect in the design of the memory model is the action to take when storage space is running out. Three main techniques exist in this situation each with their own advantages, drawbacks and trade-offs. Additionally, the selection of an appropriate technique is related to individual memory characteristics and application scenarios. This can potentially be represented through a sets of parameters understood by the memory model. The goal in this paper was the design of a generic smart object memory model, rather than provide specific implementation details, and instigate discussion at the workshop.

References

[1] M. Czerwinski, D. Gage, J. Gemmel, C. Marshall, M. Pérez-Quinones, M. Skeels, and T. Catarci. "Digital memories in an era of ubiquitous computing and abundant storage", Communication of the ACM, 49, 1, 44-50, 2006.

[2] K. Mase,Y. Sumi, and S. Fels. " Memory and Sharing of Experiences" Personal and Ubiquitous Computing, 11, 4, 2007.

[3] A. Fitzgibbon, and E. Reiter,. Memories for life: Managing information over a human lifetime. In T. Hoare & R. Milner (Eds.), Grand Challenges in Computing Research, (13-16). Swindon, 2004.

[4] G. Kortuem, L. Ball, J. Busby, N. Davies, C. Efstratiou, M. Iszatt-White, J. Finney, and K. Kinder, Sensor Networks or Smart Artifacts? An Exploration of Organizational Issues of An Industrial Health and Safety Monitoring System. Proceedings of Ubicomp 2007, Inssbruck, Austria, 2007.

[5] D. Fitton, V. Sundramoorthy, G. Kortuem, J. Brown, C. Efstratiou, J. Finney, and N. Davies, Exploring the Design of Pay-Per-Use Objects in the Construction Domain. European Conference on Smart Sensing and Context, Zurich, Switzerland, pp 192-205, 2008.

[6] Auto-ID Labs http://www.autoidlabs.org/

[7] P. Ljungstrand, J. Redstrom, and L. E. Holmquist, Web-stickers: using physical tokens to access, manage and share bookmarks to the web. Designing augmented reality environments, pp 23-31., 2000

[8] S. Konomi, and G. Roussos, G. Ubiquitous computing in the real world: Lessons learnt from large scale rfid deployments. Personal and Ubiquitous Computing, 11(7), 507-521, 2007

[9] R. Want, K.O. Fishkin, A. Gujar and B. Harrison. Bridging physical and virtual worlds with electronic tags. In ACM Conference on Human Factors in Computing Systems, pp370 – 377, 1999.

[10] J. Rekimoto, and Y. Ayatsuka, Y. Cybercode: Designing augmented reality environment with visual tags. In Designing Augmented Reality Environment, pp 1-10, 2000

[11] M. Schneider, M. Towards a General Object Memory, 1st International Workshop on Design and Integration Principles for Smart Objects, 2007

[12] S. Rein, F. Fitzek, "Compression of Short Text on Embedded Systems", Journal of Computers, 2006.