

How the Web of Things Challenges Requirements Engineering*

Pete Sawyer, Animesh Pathak, Nelly Bencomo, and Valérie Issarny

Inria Paris-Rocquencourt, France

p.sawyer@lancaster.ac.uk,

{animesh.pathak,nelly.bencomo,valerie.issarny}@inria.fr

Abstract. As a subset of the Internet of Things (IoT), the Web of Things (WoT) shares many characteristics with wireless sensor and actuator networks (WSANs) and ubiquitous computing systems (Ubicomp). Yet to a far greater degree than the IoT, WSANs or UbiComp, the WoT will integrate physical and information objects, necessitating a means to model and reason about a range of context types that have hitherto received little or no attention from the RE community. RE practice is only now developing the means to support WSANs and UbiComp system development, including faltering first steps in the representation of context. We argue that these techniques will need to be developed further, with a particular focus on rich context types, if RE is to support WoT application development.

Keywords: Web of Things, Context, Requirements Engineering.

1 Introduction

Requirements engineering (RE) [7] has evolved to discover, model, specify and manage the required and desired properties of software systems. However, conventional RE makes an assumption that the knowledge from which the requirements will be formulated exists *a-priori*, even though the knowledge may be fragmentary, distributed and tacit. Thus, although their discovery may take significant effort, the requirements *are* discoverable using the appropriate RE practices.

However, the last decade or so has seen the emergence of new types of systems where this assumption does not hold. Two prime examples of such systems are ambient, ubiquitous and pervasive (Ubicomp) systems, and wireless sensor and actuator networks (WSANs). Conventional RE is ill-equipped to discover, model, specify and manage these systems' requirements because incomplete knowledge of the context under which they must operate is available at design time. While some progress has been made, by (e.g.) maintaining requirements models that support reasoning over context at runtime, the IoT and, even more recently, the WoT have now emerged to compound the challenge for RE.

* This work was supported in part by the European Commission FP7 CHOReOS and NESSOS projects, and the ANR Murphy project.

It could be argued that the RE-significant characteristics of the IoT and WoT [3] are simply the union of those of Ubicomp (primarily heterogeneity among devices) and WSANs (primarily large scale). Certainly, context and design-time uncertainty need to be handled, implying that many IoT and WoT applications need to be requirements-aware [12]. However, the WoT adds something that is significantly new; the ambition to seamlessly fuse physical and information objects. The contribution of this position paper is to argue that this fusion of the physical and virtual requires reasoning about kinds of context that have hitherto been neglected by RE: *personal*, *social* and *information* context.

2 The Web of Things

The IoT [8] has been defined as [2] “*A global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities.*”

The IoT builds upon ideas already developed in WSANs and Ubicomp and the WoT can be considered to be that subset of the IoT that uses web standards to connect physical objects and information resources seamlessly. The rationale for the WoT is that the use of web standards should ease the integration of new smart objects as well as making the developer’s job easier. The ever-increasing abundance of existing web-enabled devices such as smart phones, and the expected web-enablement of everyday objects that is envisioned by Ubicomp (fridges, cars, etc.) is bringing the WoT closer to realization. We expect to see increased adoption of WoT concepts in the fields as diverse as personal health, inventory management, and domestic energy usage monitoring, which bring to the fore the dual challenges of scale and heterogeneity.

WoT application development is envisaged to span a range of project types from ad-hoc user-developed Mashups, through domain-expert application programming to conventional commercial development by software professionals. Our interest in this paper is towards the latter end of the range and we are particularly interested in how and whether RE may be usefully applied to WoT application development. If RE can be adapted to support the understanding of and reasoning about the WoT problem and solution domains in a way that helps domain experts and developers communicate the application requirements, then the same techniques may also prove useful to domain experts developing their own applications. In any case, evidence from experience with service-based systems suggests that integration standards don’t make the need for rigorous development processes and practices (whether these are conventional or agile) go away. This is particularly true where the problem includes multiple stakeholders, legacy systems and all the other scale-and complexity-related factors that are a routine feature of software development projects.

3 Illustrative Application

To illustrate the WoT, consider a *Liveliness Indicator* application. This is conceived as a cost-effective method of determining the relative *liveliness* of different city locations, such as city squares, parks, areas of a university campus, etc. Such an application might be useful for tourists during city festivals, for example to help tourists find street theatre during the Edinburgh Fringe.

The application is based on an assumption that ambient noise is an indication of human activity, in which the more noise (filtered to remove, e.g., traffic noise) there is, the livelier is the location. The application would display liveliness indicators for the city, and depending on the user's preferences, the application would recommend a location to visit. Thus if the user was a student wishing to study for an exam, a quiet park might be a suitable option. If they were looking for somewhere to celebrate after finishing their exam, a lively square might be chosen.

Figure 1 shows a very simple i^* [17] Strategic Dependency Model indicating as circles the primary agents; a *Noise data provider*, the user of the application *Liveliness user*, and the application itself *Liveliness Indicator*. The Liveliness user has a goal depicted by ellipses, to *Find suitably lively location* and satisfaction of this goal is dependent on the Liveliness Indicator application. This dependency is indicated by the direction of the 'D' characters on the arcs that connect the goal to the *dependee* and *dependum* agents. The Liveliness Indicator itself, is dependent on the Noise data provider to provide resources in the form of *Noise data* and *Location data* depicted as rectangles. Finally, the Noise data provider has a reciprocal dependency on the Liveliness Indicator to satisfy a QoS requirement, represented here as the *softgoal Preserve privacy*.

The application could be realized by exploiting the microphones that form part of modern smartphones, whose owners volunteer to provide the data anonymously. The application could aggregate noise data from collocated volunteers. Thus the data is crowdsourced, and the volunteers' smartphones are the smart objects - the *things* - in a WoT. Finally, the Liveliness Indicator might be used as a Facebook plug-in helping people organize social events.

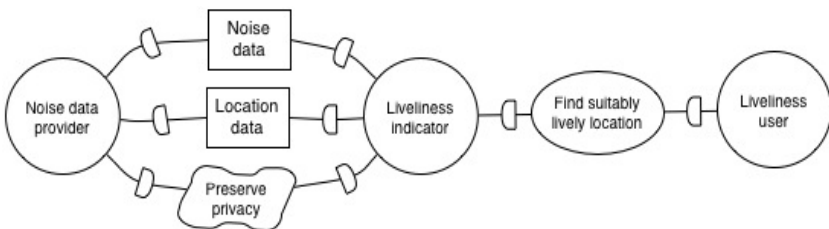


Fig. 1. i^* Strategic Dependency Model for Liveliness App

4 WoT Requirements

As a first step towards identifying how RE needs to change to accommodate the needs of WoTs, we draw on experience from Ubicomp and WSAN applications. With Ubicomp systems, a key challenge is that of modeling context [9] [5] [13], which can be succinctly defined as “... partial state of the world that is relevant to an actor’s goals” [1]. Krogstie decomposes context under six different headings:

- *spacio-temporal* context, including “location, ... place, the social arena”;
- *environment* context, including “things, noise, persons and networks”;
- *personal* context, including “mood, expertise, anger, stress”;
- *task* context;
- *social* context;
- *information* context “... the global and personal information space ...” [9].

All of these types of context are relevant to actors in the Liveness Indicator application, particularly location, noise, mood (somewhere quiet or somewhere lively?), task (somewhere to study? somewhere to celebrate?), social (somewhere to meet friends?), information (need to avoid disclosing smartphone owner’s identify). However, a fundamental feature of context is that while it should be possible to anticipate the kinds of context of relevance, not all types of context can be directly sensed at runtime or easily modeled. Further, there may be many different combinations of context, each of which may have a wide range of possible values. Thus enumeration of the system’s required behaviour under each context state may be impractical.

Finkelstein and Savigni [5] proposed requirements reflection as a means for dealing with context. An implication of this was that the system would need to maintain run-time requirements models that permitted systems to introspect over the current degree to which their requirements were being satisfied. This in turn implies some kind of requirements monitoring [4]. Recent work has seen progress towards realization of Finkelstein and Savignis’ vision, by employing goal models kept “live” for reasoning over and (to a lesser extent) updating at runtime. In [1] and [16], for example, context is sensed and when change is detected, an adaptation is triggered to optimize satisfaction of those requirements that are context-dependent.

Context is also important for WSANs, although it is typically less rich than that for Ubicomp systems. For example, the relative priorities of the non-functional requirements (NFRs) of the GridStix [6] flood prediction WSAN changed according to whether the river in which the sensing nodes were located was quiescent (energy conservation was prioritized) or in flood (network resilience was prioritized). GridStix further illustrated the problem of incompletely understood context and its impact on the system. Thus how the weather affected wireless signals and battery life proved to be hard to predict; an example of design-time uncertainty that could only be resolved at runtime. The key point is that, to date, the Ubicomp, WSAN and RE communities have focussed on a subset of context types, particularly spacio-temporal and environmental context.

For these types of context, significant progress has been made, particularly by the Ubicomp research community.

Integrating physical objects with information objects is a significant challenge to RE, and stems partly from the sheer variety of functions of the physical objects that may become part of the WoT, and partly from the range of information that they may hold. For example, a smart phone is a composite of phone, web client, GPS device, repository of text notes, music, photo and film library, and many more. It has a physical affordance for its user and sometimes a virtual one (e.g. as a router for other network devices, as an aggregator of noise data) and has relationships to several human, social and organizational entities; owners and service provider(s), music and app virtual marketplaces. Applications for which some subset of this information is important needs to deal with it as context. Domain ontologies and semantically-extended web standards [15] are likely to have a role to play, and current realizations of requirements-awareness will need to be augmented to exploit them.

To take information context as an example; it has been almost completely unexplored by RE. There is significant interest in modeling security and privacy requirements [11] and this is certainly important for the WoT, but is only one aspect of information context for WoT applications. For example, access to information context will enable new social networking possibilities. Were the Liveliness Indicator to be a Facebook plug-in, modeling of personal and social context would also become important. Sentiment analysis [10] allied to recent work on emotional requirements [14] might permit the application to infer the user's mood, but how to model and how to react to mood (does sad imply they will want to go somewhere quiet or somewhere lively?) is much less clear, and thus can't yet be sensibly specified.

5 Conclusions and Future Work

In this paper we draw on recent work on Ubicomp and WSANs to draw out some ways in which the WoT challenges existing RE practices. Our conclusion is that RE is currently poorly suited to the problems posed by WoT applications. In part this is nothing new; as new technology-driven classes of system emerge, the role of RE is limited until the technology matures and applications need to become requirements-driven to be useful or competitive. However, we believe that there are more fundamental inhibitors to the eventual utility of RE to developers of WoT applications.

Recent research on requirements-awareness [12], [1], [16] has shown that RE for many new classes of system spans design- and runtime, and reasoning with context for WoT applications will be no exception. We are far from ready to define a research roadmap for RE for WoT applications, but some waypoints might include research on modeling, sensing and reasoning over personal, social and information context [9].

References

1. Ali, R., Dalpiaz, F., Giorgini, P.: A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering* 15, 439–458 (2010)
2. Casagras, R.: RFID and the inclusive model for the Internet of Things report (2011)
3. Dillon, T.S., Talevski, A., Potdar, V., Chang, E.: Web of Things as a Framework for Ubiquitous Intelligence and Computing. In: Zhang, D., Portmann, M., Tan, A.-H., Indulska, J. (eds.) *UIC 2009*. LNCS, vol. 5585, pp. 2–13. Springer, Heidelberg (2009)
4. Fickas, S., Feather, M.: Requirements monitoring in dynamic environments. In: *Second IEEE International Symposium on Requirements Engineering, RE 1995* (1995)
5. Finkelstein, A., Savigni, A.: A framework for requirements engineering for context-aware services. In: *First International Workshop From Software Requirements to Architectures (STRAW 2001)*, pp. 2–7 (2001)
6. Hughes, D., Greenwood, P., Coulson, G., Blair, G.: Gridstix: Supporting flood prediction using embedded hardware and next generation grid middleware. In: *Proc. the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks (WOWMOM 2006)*, pp. 621–626 (2006)
7. Jackson, M.: Defining a discipline of description. *IEEE Software* 15(5), 14–17 (1998)
8. Kortuem, G., Kawsar, F., Fitton, D., Sundramoorthy, V.: Smart objects as building blocks for the internet of things. *IEEE Internet Computing* 14(1), 44–51 (2010)
9. Krogstie, J.: Requirements engineering for mobile information systems. In: *Proceedings of the Seventh International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2001)*, Interlaken, Switzerland (2001)
10. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* 2(1-2), 1–135 (2008)
11. Price, B., Adam, K., Nuseibeh, B.: Keeping ubiquitous computing to yourself: A practical model for user control of privacy. *International Journal of Human-Computer Studies* 63(1-2), 228–253 (2005)
12. Sawyer, P., Bencomo, N., Whittle, J., Letier, E., Finkelstein, A.: Requirements-aware systems: A research agenda for RE for self-adaptive systems. In: *18th IEEE International Requirements Engineering Conference (RE 2010)*, pp. 95–103 (2010)
13. Schmidt, A., Beigl, M., Gellersen, H.W.: There is more to context than location. *Computers and Graphics* 23(6), 893–901 (1999)
14. Sutcliffe, A., Thew, S.: Analysing “people” problems in requirements engineering. In: *ACM/IEEE 32nd International Conference on Software Engineering (ICSE 2010)*, vol. 2, pp. 469–470 (2010)
15. Toma, I., Simperl, E., Filipowska, A., Hench, G., Domingue, J.: Semantics-driven interoperability on the future internet. In: *IEEE International Conference on Semantic Computing (ICSC 2009)*, pp. 551–558 (2009)
16. Welsh, K., Sawyer, P., Bencomo, N.: Towards requirements aware systems: Runtime resolution of design-time assumptions. In: *26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, pp. 560–563 (2011)
17. Yu, E.: Towards modelling and reasoning support for early-phase requirements engineering. In: *Third IEEE International Symposium on Requirements Engineering (RE 1997)*, pp. 226–235 (1997)