Pervasive and Mobile Computing 6 (2010) 239–253

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Incremental awareness and compositionality: A design philosophy for context-aware pervasive systems

Seng W. Loke

Department of Computer Science and Computer Engineering, La Trobe University, Australia

ARTICLE INFO

Article history:
Received 12 March 2008
Received in revised form 8 October 2008
Accepted 19 March 2009
Available online 31 March 2009

Keywords:
Context-aware computing
Incremental awareness
Composition
Software engineering
Declarative programming languages

ABSTRACT

Context-aware pervasive systems are an important emerging category of software, increasingly pervading into daily life, play and work. These systems are characterized by capabilities for sensing the physical world and taking action, autonomously or in cooperation with users. This paper proposes an incremental approach to building context-aware pervasive systems, with a particular emphasis on systematically extending over time the contexts and situations a system can be aware of, and creating a formalism in which these systems can be composed. We present a formalism of operators for building context-aware pervasive systems incrementally and in a compositional manner (by combining multiple systems and subsystems), facilitating reuse in a formal way. The formalism can serve as: (i) a basis for a scripting language for programming composite systems, and (ii) a language for specifying these systems (whether existing or to be built) and then reasoning with specifications of these systems.

1. Introduction

There has been a growing excitement about context-aware pervasive systems, that is, systems capable of knowing and understanding the physical and virtual context of users and objects and responding intelligently to such knowledge. From context-aware services to artifacts, there has been diverse efforts to build context-awareness into systems [1,2]. Such systems are termed "pervasive" in the way that they can be ubiquitous and pervade in daily living environments.

This paper proposes an incremental approach to building context-aware pervasive systems, with a particular emphasis on extending over time what a system can be aware of and creating a formalism in which these systems can be composed (perhaps old with old or new with old, or new with new). The idea is that a system initially built might only be capable of recognizing particular contexts or particular situations[1] of entities, but later, can be extended (by developers or even more technically savvy users) to recognize more types of context and more situations. We do not deal with learning in this paper, i.e., applying machine learning techniques to recognize more context and situations, but our approach does not necessarily exclude applying machine learning later. A key underlying assumption in our approach is that knowledge about situations can be modularized or discretized, as in [7,4].

More specifically, the contributions of this paper are twofold:

1. We propose an abstract model of a situation recognition system, influenced by our discretized view of knowledge about situations. The model makes precise what it means for a system to be more powerful (in recognizing situations) than another system.

E-mail address: s.loke@latrobe.edu.au.

[1] We take situation as being at a higher level of abstraction than context as also done in other works [3–6].

1574-1192/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2009.03.004

2. We then consider a language of operators for combining a mathematical model of context-aware systems, and show how composite systems formed using these operators results in greater recognition ability than individual systems. From a software engineering perspective, this language can serve two purposes, programming and specification:
   - *a basis for scripting or programming composite context-aware pervasive systems*: given an existing set of context-aware systems, the operators can be used to compose any subset of them in a systematic way with well-defined semantics, and
   - *a language for specifying these systems (whether existing or to be built) and then composing specifications of these systems*: at design-time, a set of context-aware systems can be represented in abstract form (in a way we will demonstrate later). We can then reason with different possible compositions of these systems to explore the capabilities of these composites (or of their specifications). Moreover, the same system can be reused in different composites, and hence, the formalism can be used to reason with and represent modular designs.

Our approach has the following benefits:

- *tailored formalism:* we view this language as a first step towards a specialized formalism tailored for building extensible and composable context-aware pervasive systems, in contrast to general specification formalisms for distributed systems or ad hoc approaches.
- *extensible:* our approach supports the modular construction of context-aware pervasive systems, not only by encouraging such systems to be built separately and then composing them using operators (whether at run-time or design-time), but also, the set of operators to be changed (more operators can be added over time).
- *high level of abstraction:* our approach models systems either at the blackbox level, or a whitebox level; but does not require details of the underlying implementation of the system.

By "context", in this paper, we use Dey's definition [8], which is "any information that can be used to characterize the situation of an entity". Although we have in mind the typical sensors (from temperature to motion sensors), we also use a broad definition of a sensor, which is any device (software and hardware) that can be used to acquire context. By situation, we mean a state of affairs or from [9], "a structured part of reality that it (the agent) somehow manages to 'pick out'". And this done by directly perceiving the situation and some reasoning without requiring the agent to give a complete or exact description.

We use an analogy with expert systems, where particular knowledge is added to a generic reasoning engine in order to create a particular expert system, and such knowledge can be updated or extended, thereby updating or extending the expert system. Often, such knowledge might be structured into modules. The typical form of knowledge represented as rules facilitates this process, in that knowledge is in a sense discretized into collections of units of knowledge, where a unit of knowledge is a rule or a bunch of related rules. We suggest that what a system can be aware of can be similarly discretized, and call this *awareness discretization*; for example, a system capable of recognizing situations A, B and C can be extended to recognize situations (i.e., physical situations of the world as well as computational and networking states) A, B, C and D, or previously a system can only recognize three types of context (e.g., location, time and nearby objects) but is extended to recognize temperature as well (another type of context information). We have in mind a knowledge-based approach in this paper using a representation of situations as units of knowledge, e.g. *situation programs* in [2], but note that our approach is not restricted to only systems using situation programs.

The rest of this paper is organized as follows. Section 2 presents an abstract model of incremental context-awareness, introducing a notion of monotonic extension of context-aware pervasive systems. Section 3 dicusses a language of operators for composing context-aware pervasive systems. Section 4 describes examples of compositions in a smart home context. Section 5 outlines a scheme to compose systems running on different machines. Section 6 notes related work and Section 7 concludes with directions for future work.

2. An abstract model of incremental context-awareness

Following [1], we consider a typical context-aware pervasive system as having three subsystems in order to recognize situations: a subsystem comprising sensors, a subsystem mapping sensor readings to context or models of situations, and a subsystem taking actions. If we are extending a context-aware system, we want to be able to say, in some precise way, the sense in which one system extends another, in terms of the ability to recognize situations, and the following subsections aim towards this.

2.1. Recognition power

Suppose that, with respect to a system R, we use the operator $R(K) \vdash S$ to mean, the system with capability K is able to recognize situation S. By "capability" we mean knowledge as well as perhaps sensors and reasoning components to make recognition of situations possible.

We can define the *recognition power* $\mathcal{R}_K^R$ of the system with capability K as *the set of all situations that the system can correctly recognize*, and write this as follows:

$$\mathcal{R}_K^R = \{S \mid R(K) \vdash S\}.$$