

USING SITUATION LATTICES TO MODEL AND REASON ON SITUATIONS ¹

Juan Ye²

Abstract

Situations, abstracted from context, are more decisive than individual pieces of context in driving a system's adaptive behaviours. Current research places emphasis on formally specifying individual situations in logical predicates. However, they do not provide an approach on how to maintain the consistency of a set of situations and how to efficiently and accurately identify their occurrences when faced with a large number (e.g., hundreds) of possible situations. To solve these problems, we propose a data structure – the situation lattice – to organise situations. We will apply a Bayesian Network technique and lattice operators to efficiently and accurately predict situations.

1. Introduction and Motivation

Pervasive computing attempts to make human lives simpler in environments filled with various computing devices that are sensitive, adaptive, and responsive to human needs. Pervasive computing systems operate under dynamic and ever-changing environments, which require systems to react to dynamic changes in a seamless and unobtrusive manner. Context-aware computing is an enabling technology for pervasive computing. It exploits information from physical and computing environments without involving users' direct perception. A context-aware computing system exhibits appropriate and customised behaviours adapting to users' contexts.

Generally, *context* can be any information acquired from a pervasive computing environment. There are multiple ways to acquire context: directly sensed from physical sensors, input by users, derived from application- or meta-information existing in an environment, or merged from multiple and possibly disagreeing sensors. Sensors producing context may suffer from sensor failure, delay (or infrequent update), disconnection from the sensor network, or environmental noise. All these factors make context imprecise, conflicting, inaccurate, or unstable, which complicates context-awareness. A system may not necessarily be expected to respond to each and every change of context. One of the main questions in the area of context-awareness is how to promptly distill a large number of error-prone contexts into the important and appropriate contexts that have an effect on determining a system's behaviour. *Situation-awareness* has been suggested as an alternative to context-awareness: situations as a middle layer between context and behaviour. Context is abstracted into situations and behaviours will be defined on top of situations rather than context.

Each situation is associated with a specification that represents an invariant constraint on contexts and their combination [9]. A situation is identified (that is, an entity is considered in this situation) when

¹This work is supported by Science Foundation Ireland under grant number 05/RFP/CMS0062.

²System Research Group, UCD, Dublin, Ireland, email: juan.ye@ucd.ie, Supervisor: Dr.Simon Dobson

its specification is satisfied by the relevant contexts of the entity. A context represents a parameter of a snapshot of the reality, whose occurrence is objective and singular, e.g., a context records a person's location at a certain time. In contrast, a situation captures a particular state that is interesting to applications, whose occurrence is subjective and can be instantiated multiple times on different entities, e.g., a meeting situation can be identified on different persons in different places at different times.

A situation may encapsulate imperfect context and trivial contextual changes that do not affect the identification of situations. Therefore situations are considered more decisive than individual pieces of context in determining a system's behaviour. The question is how to efficiently and accurately determine situations. By efficiency we mean that when a situation occurs in the real world, a system should synchronously identify this situation. By accuracy we mean that an identified situation should be the same as the situation that is defined and is occurring in the real world. To take responsive and appropriate behaviours, a system must be able to efficiently and accurately predict situations.

Current research on situation-awareness puts emphasis on the representation of situations from the logical perspective [6] and the construction of situations from context [1, 9]. Situations are usually defined by developers, which can be subjective and error-prone, especially when more various contexts (e.g., thousands) are available and complicated situations are expected. One of our concerns is how to explore relations between these contexts and situations and how to maintain the consistency of a set of situations. To solve these problems, we propose an approach to automatically learn relations between contexts and situations in a hierarchy where situations are organised with respect to their internal relationships and consistency between them are maintained.

1.1. Goal Statement

The main goal of this thesis work is to propose a theoretical model to define and organise situations, within which consistency between situations is maintained through a set of rules. This model is extended with Bayesian techniques to resolve uncertainty such that situations can be predicted efficiently and accurately. To facilitate the use of this model, we will design a tool to encapsulate the procedures of specifying situations from a dataset of sensor data and inferring situations. The main contributions of this thesis will be (1) a theoretical model to organise situations where multiple situation relationships are represented; (2) a set of rules used to maintain the consistency between situations during the construction procedure of situations; and (3) a tool to automatically learn the relationship between situations and an amount of sensor data and to infer situations using Bayesian Networks.

To show the feasibility, we will implement this model. Our hypotheses are (1) that the situations organised in our model will be consistent, and (2) that the efficiency and accuracy of inferring situations within our model will be no worse than others model where more developer efforts are involved. To validate the hypotheses, we will evaluate the accuracy and efficiency by using the real datasets that include a ground truth. Using the ground truth it will be possible to validate our approach at identifying situations.

2. Approach and Methodology

We apply a *lattice* to organise a set of situations. A lattice is a mathematical data structure, which is defined as a partially ordered set where for any pair of elements there exist a greatest lower bound and a least upper bound. Lattices are useful when studying a structure with a partial order, and it has been

successfully applied in the areas of distributed computing [3] and formal concept analysis [2]. We propose a *situation lattice* to organise a set of situations with respect to their specialisation relationship. The semantics of the specialisation relationship is the implication relation of the specifications of situations. A situation is more *specific* than another situation, if the specification of the former implies that of the latter. This approach organises situations with respect to the internal relationship between situations, which makes it possible to maintain consistency based on this relationship, to improve the efficiency of forward chaining, and to deal with uncertainty when predicting situations.

Within a situation lattice, we will discuss how *consistency* is maintained between situations. We define *consistency* from two perspectives. *When a situation is identified, it is implied that all of its more general situations are identified.* The identification of a situation implies the identification of its more general situations. It is required that a specification of a situation should not be conflicting with the specification of any of its more general situations. The second consistency requirement is that *given non-conflicting context input, the situations identified should not be mutually exclusive.* Situations are mutually exclusive if they cannot occur simultaneously. It is required that any two mutually exclusive situations should include conflicting predicates in their specifications such that they can not be identified from the same context input. To maintain the consistency, both requirements will be checked during the construction procedure of a situation lattice.

Forward chaining [8] is one of the typical techniques used when inferring situations. Inference rules are applied to relevant context to arrive at a situation. A situation lattice can be helpful in forward chaining of inferring situations. A specification of each situation is created with the logical conjunction of the specifications from its more general situations and a specification that is an extension to this situation. When inferring situations using forward chaining, a system will evaluate specifications of each situation from the most general situation to more specific situations. A system will only evaluate the extended specification that is particular to each situation. This will reduce the computation load and improve the efficiency. This forward chaining algorithm has been described in [11].

Situations can be identified incorrectly since they suffer from imperfect contexts and the crudely-defined inference rules. A situation lattice cannot resolve the uncertainty alone, so we convert a situation lattice into a Bayesian network. Bayesian networks encode the strength of causal relationships with probabilities. They provide a clear and well understood method for incorporating how the likelihood of an event is conditioned on another event. This dependence relationship is implicit in a situation lattice, which makes it have a naturally similar structure to Bayesian networks. A situation lattice can be converted to a Bayesian network in a straightforward manner by annotating its nodes with prior and conditional probabilities as meta-information. This will make it possible to predict situations by propagating the probabilities from the most general situations to more specific situations.

To ease the procedure of constructing situation lattices, we will design a tool that encapsulate the following three processes: automatically generating a situation lattice from the initial input by developers; converting it to a Bayesian network; and training it with initial sample data to arrive at an applicable situation lattice.

To use this tool, developers should abstract characteristic contexts from sensor data that are available in an environment, which are interpreted immediately as basic situations. They should also define the constraints on the basic situations such as specialisation and mutual exclusiveness relations that form the initial structure of these situations. The tool will take these basic situations and generate a situation lattice automatically. The developers can label the human-understandable names on the

nodes of the generated lattice or define behaviours on them. The procedure is shown in Figure 1.

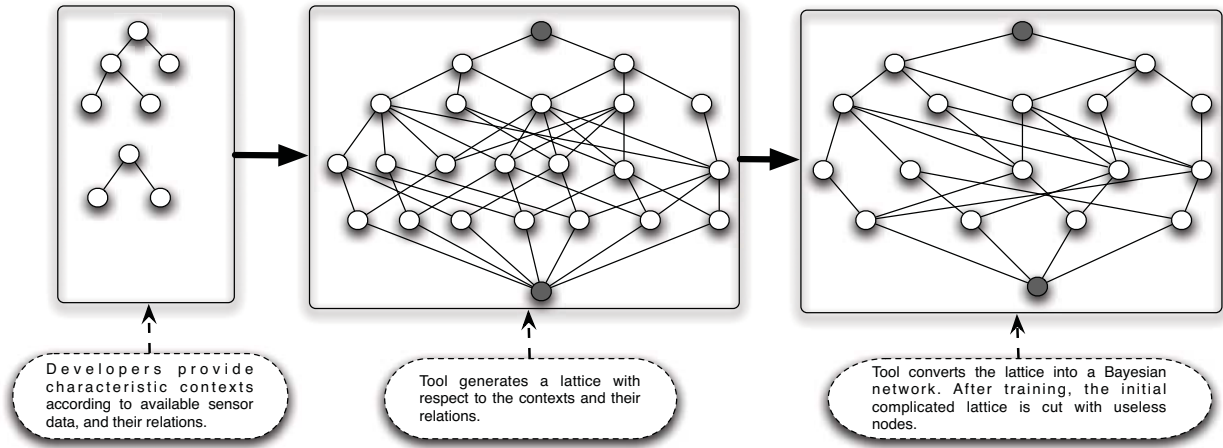


Figure 1. The procedure of constructing a situation lattice

The situation lattice is generated with the knowledge of developers through the training process of real data. Developers are not directly involved in defining and specifying each situation, which decreases the subjectivity and errors in defining situations.

3. Research Progress

We have built the theoretical model of situation lattices, and implemented an algorithm for checking consistency and an algorithm of inferring situations using forward chaining. To demonstrate the feasibility, we have constructed a simple situation lattice to describe the activities of a research student. We also have gathered a simple dataset² that captured the activities of a research student over a period of five working days. Using cross-validation, we have produced the preliminary evaluation results, like accuracy in predicting whether the subject is busy working at her desk (the “busy working” situation) correctly is 0.91 with the dataset on the fifth day (using the data from the first, second, third, and fourth days as training data) [11]. To test the general applicability of our approach, we will re-run our evaluations on a third-party dataset, e.g., the PlaceLab dataset that comprises of a set of common household activities over a period of time [5]. The ground truth of this dataset is determined from the annotated diary file.

Currently, we are working on developing a tool to encapsulate the process of creating situation lattices. This tool will automatically learn the relationships between sensor data and situations that are interesting to applications. In the process of automatically generating a situation lattice, we will evaluate the size of the lattice and check the consistency between situations. To make it easier for developers to use our tool, we will also examine how to analyse and abstract the available contexts, and what constraints should be initially defined on a lattice. In the process of merging a lattice with a Bayesian network, we will analyse a relation between the size of training data and the accuracy of the generated lattice. We assume that given more training data, the lattice will capture more accurate situations (i.e., closer to real situations) by more explicitly identifying noisy and trivial nodes. Another means to evaluate this tool is to compare the automatically generated result with the lattice that

²The sample data are published online here:

<http://kind.ucd.ie/~juanye/datasets/Ria2008Dataset.zip>.

is manually specified by developers. Since the PlaceLab dataset is public and well-known, we will also compare our accuracy with others like [5], and we expect that the accuracy of the automatically generated lattice should not be significantly worse than that of the manually defined lattice, which will be used to validate our hypothesis.

4. Related Work

Previous work in the area of situation-awareness defines situations by adding constraints on contexts expressed using a variant of logic predicate, which can be incrementally combined to form complex situations. Recently, the work has focused on formally defining situations and studying the relationships between situations [1, 6, 10].

Costa *et al* [1] and Yau *et al* [10] discuss the natural characteristics of situations, including the compositional relationships between context and situations and the internal relationship between situations. Cosat *et al* define ontological elements such as entities, intrinsic and relational contexts, which are constituent elements of a situation. Entities are users or objects that exist in the environment. Intrinsic contexts define a type of context that belongs to the essential nature of a single entity and does not depend on the relationships with other entities (e.g., `GeoLocation`). Relational context relates an entity to the collection of entities that play a role in the entity's context (e.g., `deviceAvailability`). Situations can be composed of these basic elements in multiple ways. This detailed analysis of semantics on situations provides a solid foundation of our work and inspires our idea of organising situations with respect to their internal relations.

Loke proposes to design situation programs using meta-programming within an extension of Prolog [6]. He exploits the relationships between situations: specialisation and compatibility, and applies them to reason on situations. He deals with an individual situation program, while we focus on a set of situations. He presents the logic definitions of completeness and soundness of a context-aware system. He points out that it is up to developers to verify soundness and completeness with respect to various systems. However, it is a hard task for developers, so we propose an approach to check these automatically during the procedure of defining situations.

When it comes to dealing with uncertainty in context-aware computing systems, Henricksen *et al* [4] give us a detailed analysis of uncertainty's causes in context-aware systems. Earlier attempts add quality metrics as meta-data of contexts, including resolution, accuracy, frequency, or confidence. This approach is good at expressing the uncertainty information of sensor data. However, it lacks a formal uniform technique of managing the uncertainty in a quantitative way. To solve this problem, recent research work applies fuzzy logic, Dempster-Shafer theory and Bayesian networks in resolving different uncertainty issues. Ranganathan *et al* [7] developed an uncertainty model based on a predicate representation of contexts. Each context predicate is attached with a confidence value between 0 and 1, indicating to what degree a context is true. The confidence value measures the probability (probabilistic approach) or the membership value (fuzzy logic). A Bayesian Network is built to capture the causal relationship between various context events. The high accuracy in their preliminary result shows that Bayesian networks are suitable in determining situations in context-aware systems. However, the openness and dynamism of context-aware systems make the cost of training Bayesian networks prohibitive, especially when learning the structure of nodes is required. To relieve this problem, we build our Bayesian networks from a situation lattice that pre-determines the situation structure.

5. Conclusion

The main contribution of our work is to help developers to design a consistent set of situations. To achieve this goal, we propose a theoretical model — the *situation lattice* — that organises situations with respect to the specialisation relationship. Using lattice theory, we exploit semantic relations between situations and define a suite of rules to maintain consistency of situations. This model improves the efficiency of forward chaining when inferring situations. It also increases the accuracy of inference by applying Bayesian techniques to resolve the uncertainty from imperfect context.

To make this model applicable, we will develop a tool to automatically generate situations. This approach eases the procedure of constructing situations when there are a number of varied contexts involved and complicated situations (i.e., whose specification involves complex predicates) that are required to define. Also, this tool will help to decrease the subjectivity and errors from developers.

References

- [1] P. Costa, G. Guizzardi, J. A. Almeida, L. Pires, and M. Sinderen. Situations in conceptual modeling of context. In *Proceedings of EDOCW'06*, pages 6–16, October 2006.
- [2] B. Ganter and R. Wille. Applied lattice theory: Formal concept analysis. Technical report, Technical University of Dresden, Germany, 1997.
- [3] Vijay K. Garg, Neeraj Mittal, and Alper Sen. Applications of lattice theory to distributed computing. In *ACM Special Interest Group on Algorithms and Computation Theory (SIGACT) News*, September 2003.
- [4] Karen Henriksen and Jadwiga Indulska. Modelling and using imperfect context information. In *Proceedings of PERCOM '04 Workshops*, pages 33 – 37, 2004.
- [5] Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Munguia Tapia, and Stephen S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *Proceedings of Ubicomp 2007*, pages 483–500, September 2007.
- [6] Seng W. Loke. Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective. *Knowledge Engineering Review*, 19(3):213–233, 2004.
- [7] Anand Ranganathan, Jalal Al-Muhtadi, and Roy H. Campbell. Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 03(2):62–70, 2004.
- [8] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. MIT Press, 2000.
- [9] N. Weisenberg, R. Gartmann, and A. Voisard. An ontology-based approach to personalized situation-aware mobile service supply. *Geoinformatica*, 10(1):55–90, 2006.
- [10] Stephen S. Yau, Dazhi Huang, Haishan Gong, and Yisheng Yao. Support for situation awareness in trustworthy ubiquitous computing application software. *Software: Practice and Experience*, 36(9):893–921, 2006.
- [11] Juan Ye, Lorcan Coyle, Simon Dobson, and Paddy Nixon. Representing and manipulating situation hierarchies using situation lattices. *Revue d'Intelligence Artificielle*, 2008. To appear.