Using fibrations for situation identification

Simon Dobson and Juan Ye

Systems Research Group, School of Computer Science and Informatics, UCD Dublin
IE
{simon.dobson,juan.ye}@ucd.ie

Abstract. Pervasive and autonomic systems rely on notions of *context* and *situation* to control their adaptation to changing user and system circumstances. Ensuring that a system exhibits the correct adaptation to on-going circumstances remains problematic, however, especially in the presence of dynamic user and device populations, changing requirements and complex interacting adaptations. We discuss the use of fibres and fibrations between context and situation graphs as a basis for reasoning about context-sensitive behaviour, and suggest that these concepts may provide both a semantic and implementational basis for highly adaptive systems.

1 Introduction

Context-aware systems adapt their behaviour to changes in their environment or use. The same constraints and techniques apply in large measure to user-facing systems reacting to real-world environments (pervasive computing), to self-managing infrastructure (autonomic computing), and other domains.

The core common problems are to collect and represent large volumes of disparate and richly-interlinked information, and to distill this into a smaller set of situations which drive behaviour [1]. A situation represents the semantic interpretation of context, and is generally derived by fusing several pieces of contextual information in some way, with potentially many different contexts being indicative of the same situation. The existence of noise and uncertainty mean that situation identification is needed to clean up raw context information before selecting behaviours.

A systems approach to context-aware systems therefore requires that we can specify situations and the contexts that they represent compositionally rather than *en bloc*. This immediately leads to two questions: how can we ensure that the “right” situation and behaviour are selected in a given context? [2], and how can situations be specified compositionally? These problems might be regarded as together giving a semantics to context-aware computing systems which would allow meaningful analysis and design at a component, rather than full-system, level.

Both context and situations are typically modelled as graphs, so the issue becomes one of capturing relationships between (collections of) graphs in such a way that the situations selected by particular contexts change in a well-defined and externally-correct way. In this paper we consider using a particular relationship, *fibrations*, between context graphs to model the semantics of adaptations. These fibrations provide a strong connection between context and situation, and hence behaviour. We present some basic observations on the mathematical structures involved and provide some pointers for further investigation. Our aim is to stimulate a discussion on how best to structure, represent and analyse complex adaptive behaviours.

2 Modeling context, situation and behaviour

Both contexts and situations can be represented as graphs, for which we will use the standard model:

Definition 1. *A* ***graph*** $G = (N_G, E_G)$ *consists of a set of nodes* N_G*, a set of edges* E_G*, and a pair of functions* $s_G, t_G : E_G \rightarrow N_G$ *(referred to as the* ***incidence maps****) mapping each edge to its source and target node respectively. We use the notation* $G(a, b)$ *to refer to the set of edges with source* a *and target* b *in* G*, that is to say* $G(a, b) = \{e \in E_G | s(e) = a \wedge t(e) = b\}$*. The edges in a given* $G(a, b)$ *are termed* ***parallel edges****. A* ***sub-graph*** $G = (N_{G'}, E_{G'})$ *of a graph* G *(written* $G' \leq G$*) has* $N_{G'} \subseteq N_G$ *and* $E_{G'} \subseteq E_G$ *with the domains of* $s_{G'}$ *and* $t_{G'}$ *being contained in* $N_{G'}$*.*

Context is typically modelled using graphs whose nodes are object identifiers or values, and whose edges are predicates asserting a relationship between an object and either another object or a value. The Resource Description Framework (RDF) [3] is becoming standard within the community, as it provides a straightforward and well-founded method for context modelling that can leverage many of the tools being developed as part of the semantic web initiative.

A typical application of RDF to modelling context may be seen in location-based services. The location l of an object a under some model of location might be represented by an edge (a, p, l) where p is a predicate (represented by a URI) that identifies uniquely the intended interpretation of the relationship between a and l. As the object a moves, the model evolves to change the value mapped to by p.

Situations are the first-level external interpretation of context. A situation might be "in a meeting" or "travelling in her car from home to the office" – an externally-meaningful interpretation of the observed data. Context is typically more variable than situation: using the travelling example above, a number of location observations for the individual might be mapped to the same situation. This is the process of **situation identification**, which is the core semantic process in context-aware adaptive system.

Given a set of situations, we may define a behaviour which will be exhibited by the system when it finds itself in a given situation. There may be structure on the order of situations, in that (for example) one must travel to a meeting before being at that same meeting: this induces a graph structure on the situations whose edges represent possible transitions between situations. In essence this is a simple form of workflow which captures the expected external semantic behaviour of the users and environment of the system, and is used to structure the observed behaviour.

The situation is complicated by the errors and uncertainties inherent in much contextual information. Sensors have only limited precision, and may exhibit glitches or other spurious outputs which, if reacted to directly, could cause the system's view of its current situation to become unstable. A real-world context-aware system must therefore adopt measures to stabilise mapping from context to situation [4].

As a first model, however, the objects, sensors and information sources we have available give rise to a set of all possible context graphs – all possible states of the world as observed by these sources. We may further induce a graph structure onto this set by treating each possible context graph as a node in the larger graph [1] (which we will term the **context evolution graph**) and, given two context graphs c_1 and c_2, introducing an edge between them if they differ by a single observation (either having an extra edge, or mapping an edge to a different value, and so forth). Clearly there will be at most one edge between any pair of nodes.

We therefore have two graphs, one capturing the possible changes in the observed state of the world, and one capturing the situations and their legal transitions. The relationship between these two graphs is what defines the overall behaviour of the system.

Definition 2. *Let A and B be graphs. A* ***graph homomorphism*** $f = (n_f, e_f)$ *is a pair of functions $n_f : N_A \to N_B$ and $e_f : E_A \to E_B$ that commute with the incidence maps, so $s_B \circ e_f = n_f \circ s_A$ and $t_B \circ e_f = n_f \circ t_A$.*

Given a context evolution graph E and a situation graph S, the semantics of the system is given by a graph homomorphism $i : E \to S$ that maps contexts to situations and context transitions to situation transitions. This homomorphism provides a complete description of the system's identification of situations, and therefore a complete description of its adaptation to context.

2.1 Fibres and fibrations

We may identify some useful structures within this homomorphism:

Definition 3. *Let $f = (n_f, e_f)$ be a graph homomorphism. If b is a node in B, then the* ***fibre of*** f ***above*** b *is the sub-graph G' of G consisting of the nodes of $a \in N_G$ for which $n_f(a) = b$ and the set of edges $e \in E_G$ such that $s_B \circ e_f(e) \in N_{G'}$ and $t_B \circ e_f(e) \in N_{G'}$ A fibre is said to be* ***proper*** *if it contains more than one node. The set of nodes in the fibre above b is denoted $n_f^{-1}(b)$, the inverse image of n_f.*

---

[1] This structure – a graph whose nodes are graphs – is sometimes referred to as a *condensed* graph.

One may view a fibre as an equivalence class of nodes in the source graph: equivalent nodes map to the same node in the target graph. The fibre above a situation is those contexts which will be identified as placing the system "in" the given situation.

However, there is a somewhat stronger relationship between fibres:

Definition 4 (Boldi and Vigna [5]). *Let A and B be graphs, and let $f = (n_f, e_f)$ be a homomorphism between them. f is a* ***fibration*** *if, for each edge $e \in E_B$ and each node $a \in N_A$ such that $n_f(a) = t_B(e)$, there is a unique edge $e^a \in E_A$ (called the* ***lifting of*** *e* ***at*** *a) such that $e_f(e^a) = e$ and $t_A(e^a) = a$. The graph A is referred to as the* ***total graph*** *of the fibration, with the graph B being referred o as the* ***base graph****.*

What the definition implies is that, for every edge e between fibres in the base graph, there is at most one edge whose target is in the fibre $n_f^{-1} \circ t_B(e)$ and whose source is in the fibre $n_f^{-1} \circ s_B(e)$. Essentially the fibration collapses the total graph onto the skeleton graph in such a way that there is only one way of moving from one fibre to a particular node in another.

3 Applications to contextual reasoning

How does such a simple model help in constructing context-aware systems? We believe there are three answers to this question. Firstly, constructing a predictable adaptive system involves ensuring that the transitions between situations (the base graph of the fibration) correspond correctly to the externally-desired behaviour of the system. Secondly, fibrations in many cases compose cleanly, allowing different aspects of behaviour to be specified independently – and the cases where it is *not* possible to describe a system as a fibration may point to important design decisions that need to be resolved. Thirdly, certain aspects of sensor uncertainty may be explored within the fibration.

3.1 A simple location-based service

For the rest of this paper we will assume the existence of context graphs having two users a and b being mapped to one of locations $l_1, l_2, \cdots, l_n$ by a predicate p. As a is seen to move, the graph will change: if a starts in location l_1 and then moves to l_2 and then l_3 while b remains in l_1, the model will evolve through a succession of three graphs (figure 1): at each step, a single edge changes, giving rise to a context evolution graph *Loc*.

Suppose we now want a system to adapt to changes in this context. We might do this by identifying a number of situations, such as (for example) exhibiting one behaviour when a and b are together in the same location and a different behaviour otherwise. If we model the behaviours as a two-node situation graph *Comp* with *separate* and *meet* actions and"identity" loops, we may model this situation as a fibration f shown in figure 2.

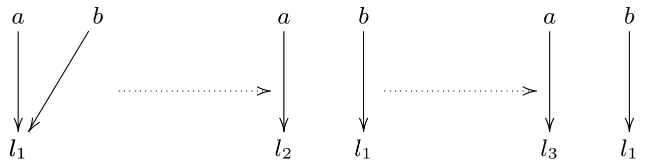

Fig. 1. Evolution of a location model

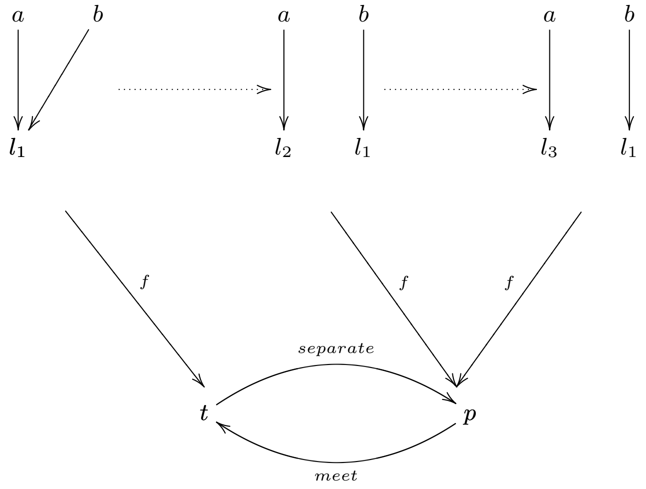

Fig. 2. Location-sensitive behaviour as a fibration

What is this structure telling us? Certain configurations of the context model are identified as different situations, in this case when a and b are together in the same location (situation t) or apart (situation p). The transitions between these situations define the possible state changes. The fibration both identifies the situation for each model *and* matches transitions between models with transitions in the workflow. (The definition of the evolution graph means that it has no parallel edges, so uniqueness of the lifted edge is guaranteed.)

The models in each fibre are "equivalent", in the sense that they denote the same situation. Since we will typically use a situation as a selector for behaviour – we exhibit certain behaviours when in certain situations – this formulation allows us to define which contexts will lead to which behaviours.

3.2 Composing fibrations

The purpose of a contextual system is to collect richly-interconnected information, and this immediately raises the problem of specifying the response that a system makes to particular contexts.

Suppose we have another graph *Act* which maps the individuals a and b to actions $a_1, a_2, \cdots, a_n$ they are observed to be performing, such as reading e-mail or accessing presentation software. We may fibre this graph over another situation graph representing (for example) the actions performed in a meeting, while travelling and so forth, leading to a fibration g. We may further construct the cross-product of the two graphs *Loc* and *Act* by adding the edges and target nodes from *Act* to the subjects in *Loc*, leading to a graph which classifies a's location and action. We may similarly construct the cross-product of the two situation graphs and the cross-product of the two fibrations, classifying the larger context into its underlying situations (together in a meeting, for example).

This construction allows separate situations and contexts to be combined very simply. The more complex case – where the two fibrations specify overlapping situations, or map the same context to different situations, requires more careful handling, in that one of the fibrations must be designated as "overriding" the other. This is simple to do algebraically but relies on an understanding of the external semantics of the situations, and so requires human intervention. What we *can* do automatically, however, is identify such systems as they appear through being unable to construct a consistent fibration from the two components.

3.3 Dealing with uncertainty

All sensed or inferred information is uncertain: it is this uncertainty that precludes viewing a context-aware system as being driven by individual sensed events [4].

How does this uncertainty manifest itself in the fibration? Suppose we have a location sensor that classifies user a's location as l_1, but which unexpectedly encounters a glitch and issues an observation that a is in l_2. This gives rise to a transition along a single edge in the context evolution graph, let us say from c_1 to c_2. Although these contexts are different, they will only affect the system's behaviour if they lie over different situations, that is if $n_i(c_1) \neq n_i(c_2)$ in fibration i. If they *do* lie in the same fibre, then the error will be tolerated with no external impact; if they *do not*, then the system would make an "inappropriate" transition between situations. This could have a number of consequences, including exhibiting radically incorrect behaviour and becoming "stuck" in an inappropriate part of the workflow.

There are a number of ways to address these concerns. One way is to provide smoothing of the context so that "raw" observations are filtered to remove likely extraneous observations. The smoothed context can then be used in situation identification. Such filtering can occur purely within the context evolution graph, and so need not complicate the definition of the fibration.

An alternative strategy is to enrich the situation graph to capture uncertain position reports explicitly. This complicates the fibration but allows the system to respond (should it wish to) to uncertainty in a controlled way.

4 Conclusions and further work

We have proposed a model of context-aware adaptive behaviour based on graph homomorphisms. Such a description induces a fibration structure that highlights a number of important features within the system being described. The fibre structure constrains a system to exhibit a well-defined response to contextual change.

We have considered the case where the context evolution and situation graphs, and the fibration between them, are defined *a priori*. However, one might also consider how the situation graph and/or fibration might be constructed by observation. One might, for example, observe repetitive patterns in the transitions observed in the context evolution graph, and use this to "learn" a new situation which may then be described as a fibration and composed. This sort of learning underlies Driver and Clarke's work on "trails" [6]. Realistically such observations would need to be made by projecting sub-graphs out of each node in the context evolution graph to focus only on repetitions of "relevant" context – and this projection again has a fibre structure.

Fibrations are more commonly encountered within the framework of category theory, and while we have talked exclusively about graphs in this paper the structures we have discussed can be constructed using the category of graphs and appropriate endofunctors. In previous work [7] we suggested the use of fibres to provide a categorical semantics for context-aware behaviour: the work described here uses similar ideas on a slightly larger scale, fibring complete context graphs (rather than individual contextual parameters) over situation workflows. There are however clear links between these two scales which remain to be explored.

References

1. Coutaz, J., Rey, G.: Foundations for a theory of contextors. In Kolski, C., Vanderdonckt, J., eds.: Computer-aided design of user interfaces. Volume 3. Kluwer (2002) 13–34
2. Coutaz, J., Crowley, J., Dobson, S., Garlan, D.: Context is key. Communications of the ACM **48** (2005)
3. Lassila, O., Swick, R.: Resource Description Framework model and syntax specification. Technical report, World Wide Web Consortium (1999)
4. Dobson, S., Coyle, L., Nixon, P.: Hybridising events and knowledge as a basis for building autonomic systems. Journal of Trusted and Autonomic Computing (2006) To appear.
5. Boldi, P., Vigna, S.: Fibrations of graphs. Discrete Mathematics **243** (2002) 21–66
6. Driver, C., Clarke, S.: Context-aware trails. IEEE Computer **37** (2004) 97–99
7. Dobson, S., Nixon, P.: More principled design of pervasive computing systems. In Bastide, R., Roth, J., eds.: Human computer interaction and interactive systems. Volume 3425 of Lecture Notes in Computer Science., Springer Verlag (2004)