

Context Evolution in Conceptual Spaces

Marwa F. Khater and Ahmed Y. Tawfik

School of Computer Science
University of Windsor
Windsor Ontario, Canada
<khaterlatawfik>@uwindsor.ca

Abstract—One of the main challenges in context-aware systems is representing and reasoning about changes in context. In this paper, conceptual spaces provide basis for representing contexts and motion classes, defined for spatial reasoning, provide the tools necessary to reason about context evolution in conceptual spaces. The work also explores the consequences of integrating motion classes in conceptual spaces to represent and reason about context evolution

Keywords—context-awareness, conceptual spaces, region connection calculus (RCC), spatio-temporal reasoning.

I. INTRODUCTION

Context-awareness requires systems to adapt their operations to the current context without explicit user intervention. Recently, there have been calls to deal with context as a process instead of considering it a state [3]. For example, a context-aware printing application that chooses the most convenient printer for a user should not print each page of a file on a different printer as the user moves. Instead it should print the document at a time and location convenient for the user's intent and needs.

Activity-based computing [1] views context as the correct adaptation at a meta-level that takes into consideration the goals, the intentions, and the various tradeoffs that may be involved rather than simply adapting to some form of dynamic state. Motion awareness [12] would allow a context-aware city guide to provide different information about the same physical location like the entrance of a museum depending on whether the user is arriving to the museum, leaving it, or just seeking shelter from the rain at its entrance.

One of the main challenges facing context-aware systems is reasoning about change in context and its evolution in order to take the appropriate action.. We consider context as a point in a conceptual space, where the evolution of context can be represented as transition between points in that space.

Conceptual spaces [5] impose a structure on a multidimensional feature spaces by defining various quality dimensions. Quality dimensions are considered integral if an object cannot be assigned a value in one dimension without a corresponding value assignment in the others. Quality dimensions that are not integral are separable. A number of constraints can be defined for a dimension or on the relationships between dimensions.

Concepts form continuous and convex regions in Conceptual Spaces. Qualitative notions emerge as

abstractions of physical or perceived dimension spaces. A property represents a region in a domain (subspace) while a concept combines several separable subspaces. For example, “yellow” is a region in the hue-chromaticity-brightness domain, and a yellow ball is a concept. The region defining a property is a convex region to ensure continuity of the property such that a line connecting two arbitrary points within the region lies entirely inside the region.

These features make conceptual spaces a useful framework for knowledge representation and learning [7]. Conceptual spaces are also suitable for context awareness because they provide a natural way for relating abstract concepts to sensory data. This grounding of abstract symbols facilitates context specification, recognition and the formulation of context specific adaptations. A context in a conceptual space assigns weights representing the salience of quality dimensions [8]. These weights are allowed to change and quality dimensions can be added to the conceptual space or removed from it according to a set of rules. Effectively, changes in weights and dimensions alter distances in the conceptual space. Consequently, two points in the conceptual space can be similar in one context but different in another.

The goal of this work is to extend context in conceptual spaces to deal with contexts involving a pattern of change over time rather than a dynamic state. Section II reviews the foundations of context in conceptual spaces. Section III overviews reasoning in conceptual spaces using the region connection spatial calculus (RCC), followed by a set of RCC motion classes in Section IV. Section V presents context evolution as motion, followed by the conclusions in Section VI.

II. CONTEXT IN CONCEPTUAL SPACES

A conceptual space can be defined as an n -dimensional vector space. Formally, let the conceptual space be represented by $C^n = \{c_1, c_2, \dots, c_n\} | c_i \in C\}$ where c_i denotes a quality dimension. A domain is set of integral dimensions (or a separable subspace) that defines one of the quality dimensions Q_j and that can in turn be defined as a set of vectors in some set of quality dimensions $c_j = D^m = \{(d_1, d_2, \dots, d_m) | d_k \in D\}$ [11]. The process of defining quality dimensions in terms of more primitive dimensions supports the eventual mapping of quality dimensions to sensed quantities thorough abstractions.

Connecting sensed quantities to abstract notions is a desirable feature in context-aware systems as it supports the

automated context recognition based on the readings of sensors embedded in the system. Context recognition may then be performed by computing the distances between the present situation, represented as a point in the conceptual vector space, and some reference or prototypical contexts defined as points in the conceptual vector space as well.

As the relative importance of various quality dimensions vary with context, a set of weights representing the salience of various dimensions in a prototypical context are defined along with the prototypical context. Moreover, it is also necessary to specify how to measure distances in each domain and how to combine the distances along the various quality dimensions. Therefore, the distance between a point u and a context prototype v in the conceptual space C_v^n which is the conceptual space C^n tuned with the set of weights to recognize context v . Subsequently, the distance of situation u from prototypical context v is

$$dist_{uv} = f_c(w_{1v}, g_1(u_1, v_1), w_{2v}, g_2(u_2, v_2), \dots, w_{nv}, g_n(u_n, v_n))$$

where f_c is a combination function that combines the distances along quality dimensions given $g_i(u_i, v_i)$ weighted as appropriate for context v by the weight factors w_{iv} . Context v is activated if the present situation u is close enough to prototypical context v , or $dist_{uv} < Threshold_v$. Thus, each context is characterized as a concept that occupies a region in the conceptual space.

III. A SPATIAL CALCULUS FOR CONCEPTUAL SPACES

The relationships between regions in conceptual spaces can also be expressed qualitatively like other spatial regions. The qualitative abstraction of these relationships supports making inference and abstract reasoning about conceptual spaces. A popular region-based spatial representation known as the region connection calculus serves as a foundation for reasoning about categories in conceptual spaces [6], [9].

A. The Region Connection Calculus

The region connection calculus (RCC) is a qualitative spatial representation which exploits the connectedness of spatial regions to provide a formal calculus. The topological representation of RCC defines a set of jointly exhaustive and pair-wise disjoint relations that may hold between any two spatial regions. The RCC8 is boundary-sensitive and defines eight relations [10]. Two non overlapping regions A and B can be disconnected (written as DC(A,B)), or externally connected (EC(A,B)). Two regions A and B can partially overlap (PO(A,B)) or be equal (EQ(A,B)) RCC8 distinguishes between a proper part that meets the region at some boundary point, or tangential proper part (TPP(A,B)), and non-tangential proper part (NTPP(A,B)). The last two relations have two inverse relations TPPI and NTPPI.

The boundary-insensitive set of relationships can neither distinguish between disconnected and externally connected regions, nor between tangential and non-tangential proper parts [2]. Therefore, the five relations in RCC5 are DR(A,B) to indicate that regions A and B are distinct, PP(A,B) to indicate that A is a proper part of B, EQ(A,B) to indicate that regions A and B are equal, and

PO(A,B) to indicate that region A partially overlaps B. Figure 1 illustrates the RCC5 relationships.

A continuous motion of a spatial region with respect to another leads to the transition of relations from one relation to another following a motion path described in a conceptual neighborhood. For example, two discrete regions gradually moving towards each other will eventually partially overlap and subsequently become either equal or proper part or its inverse. The conceptual neighborhood enforces some kind of constraints on the transition between relations in the RCC.

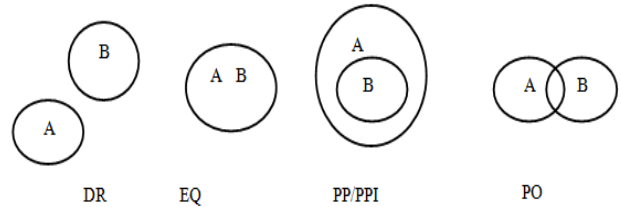


Figure 1: The RCC5 Spatial Configuration

B. Spatial Reasoning in Conceptual Spaces

Various psychological studies indicate that humans classify objects based on similarities to prototypes. Similarly, in conceptual spaces category membership is determined based on similarity to a prototype as measured by a conceptual distance threshold. The prototypes and the underlying similarity relation can be used to tessellate a conceptual space into regions representing categories [6].

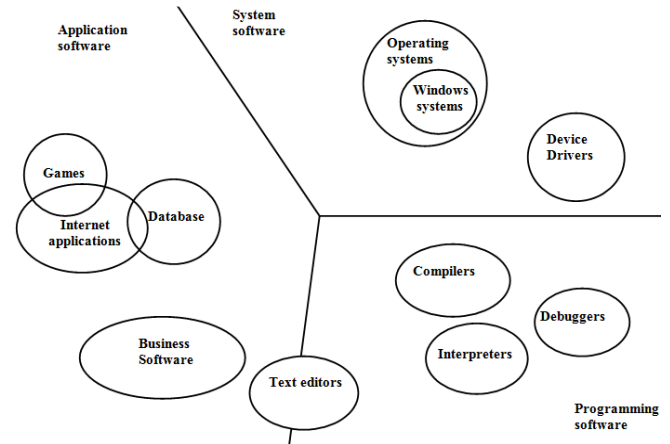


Figure 2: prototype regions in Software space, & the power categorization of application software, programming software and system software.

Voronoi tessellation can be used to compute the smallest region containing a number of prototype points. RCC machinery can be used to reason about categories and to describe other aspects of concept management. Reasoning about categories using the RCC enhances the conceptual space model.

For example let us consider the conceptual space of computer software. This representation can assist in building activity based systems that help and guide

computer users after identifying their activity. In Figure 2 and the RCC5 relationships can express relations such as: PP(operating systems, device drivers), DR(compilers, debuggers), and PO(internet applications, games).

IV. A QUALITATIVE SET OF MOTION CLASSES

Topology-based motion ontologies [12] distinguish between primitive and compound motion classes. The motion classes are formed from a topological spatial base (RCC8) to construct a set of primitive and compound motion classes. The nine primitive motion classes identified are LEAVE, REACH, HIT, BOUNCE, PERIPHERAL, EXPAND, SHRINK, INTERNAL and EXTERNAL. These classes represent in each case the motion pattern that holds between two moving regions during a specific time interval. For example, consider a vehicle that LEAVES the household in a specific location and REACHes an industrial zone CROSSing (CROSS is a compound motion class) a shopping district. The vehicle’s motion is INTERNAL with respect to the city and EXTERNAL to another city.

The set of motion classes has the property of being complete and continuous. When mapping this set of motion classes into boundary-insensitive patterns of motion in RCC5, we end up with LEAVE, REACH, INTERNAL, EXTERNAL, SHRINK and EXPAND. Table 1 gives a complete characterization of the RCC5 motion classes. In Table 1, the rows correspond to the RCC5 relation holding at the start of the motion. The columns correspond to the RCC5 holding between the two regions as the motion ends.

Table 1: The RCC5 set of motion classes

		ENDS				
		DR	PO	PP	EQ	PPI
STARTS	DR	EXTERNAL				
	PO		LEAVE or REACH		REACH	
	PP			INTERNAL	EXPAND	
	EQ				INTERNAL	
	PPI		LEAVE		SHRINK	INTERNAL

V. CONTEXT EVOLUTION

An evolution through a series of contexts is an important feature in many context-aware applications. In such applications the transitions from a context, identified by a region in a conceptual space, to another define some form of meta-context. We can use the RCC5 motion classes to represent and reason about change of context in conceptual spaces. The motion classes along with the spatial base RCC, describes all possible forms of motion between two regions.

A. Motion Classes describe concept evolution

A point in a conceptual space can be seen as a vector of coordinates, one for each dimension in the space. In that case objects can be viewed as narrow concepts with

domains reduced to points. Consider a particular object; say an apple, and how it changes over time. The conceptual vector space of an apple consists of quality dimensions such as size, color, taste and shape. It starts with a value which represents small in size, green in color bitter in taste, and round in shape. As time passes, the small immature apple, changes. The apple is later represented with values corresponding to big in size (relative to apple sizes), red or yellow in color, sweet in taste, and round in shape. There is a trajectory that could be defined describing the changes that happen to apples. This evolution can be captured by motion in conceptual spaces. Through its development, the apple LEAVES the concept of an immature apple, and eventually REACHes the concept of a fully developed apple. Moreover, we can reason about intermediate concepts. In the example, an apple definitely CROSSes the concept of semi mature apple.

The different concepts representing the apple are all moving along a trajectory that is INTERNAL to the category of fruits and EXTERNAL to the concept of red small ball. What makes these two concepts that have a lot in common but at the same time EXTERNAL is the salient weights assigned to each quality dimension. In the ball concept, the weight given to the dimension of taste would set it apart from all apples.

The size of a particular concept is remarkably important. The size of the concept “duck” is bigger than the size of the concept “ostrich” [5] because “duck” covers much more variety of birds than “ostrich”. To clarify the shrinkage and expansion motion classes in conceptual spaces consider the concept of “Internet applications”, as it EXPANDs over time. On the other hand, some concepts tend to SHRINK as time passes by. One example of a shrinking concept would be the “gramophone” which used to be the only recorded sound playing machine. The process of extinction of an animal species can also be modeled using SHRINK motion class in conceptual spaces.

B. Constrains on trajectories of evolution

From the previous sub-section, the trajectory along which the object apple is moving, as it matures, is one directional. However, some concepts may follow reversible trajectories that can be described as bi- directional evolutions. For example, consider the phase transition of a material. The dimensions would be temperature and pressure. The state of the material changes due to change in context. In a heating context this change can be from solid to liquid and then gaseous state. Yet, in a cooling context, the evolution will go in the opposite direction from a gas to a solid. The states of material can be viewed as concepts in conceptual spaces. Therefore a heating context can be described in terms of motion classes in conceptual spaces. For example, water is an object in the liquid concept, and ice is a solid. Ice cannot LEAVE from solid state and REACH the gaseous state (in this case called vapor) without CROSSing by the liquid state on the way.

More constraints can be added regarding trajectories besides the direction. In each domain there are possible trajectories and impossible ones. There is a trajectory representing the change of an object from the concept “immature apple” to the concept “fully developed apple” but there is no trajectory describing the change on an object from the concept “apple” to the concept “ball”. These constraints about trajectories govern the state of evolution and transition. It sets some rules and limits transitions in conceptual spaces to justifiable transitions.

C. Motion in Conceptual Spaces and Context

Context models vary from graphical models, object oriented models, ontology- based models and more. In our view, contexts can be represented by points or trajectories in conceptual spaces. Change-awareness has motivated the representation of context evolution as motion in conceptual spaces. This model can be integrated in location aware systems, where the change in location can be identified as motion evolution. However, there is more to this model than simply modeling motion.

The main challenge in Activity Based Computing is to identify the activity of the user. For example, consider classifying user activities in the domain of computer applications, we can classify some of these activities as browsing, programming, word processing, using spreadsheets, or watching multimedia. Taking the activity of programming as an example, we can divide it into phases like coding, debugging, and testing. These are the different states of developing a program which can be represented by the transition between three different contexts. Making use of the trajectory defined along the evolution of these contexts, the programmer LEAVES the context coding and REACHes the context testing.

The efficient capturing of change in context is a challenging task in context-aware systems. Due to this dynamic nature of context, a powerful tool for triggering a specific procedure after a change in context is essential. Knowing the trajectories of motion classes reduces the search space after triggering a change of context. Depending on the domain of application, many of them have a known and finite set of possible trajectories that specify the possible changes of context over time. These limited trajectories can limit the search space. Knowing the possible trajectories and eliminating the impossible ones eases the process of reasoning after triggering the change in context.

Identifying the motion classes and studying their trajectories support the prediction of upcoming contexts. Constraints enforced on trajectories limit the set of different contexts that could evolve. Similarly, knowing possible past contexts could be achieved based on the study of trajectories. Moreover, an intermediate unknown context, which is represented as a concept, can be known after indentifying the trajectory of the evolving context.

Suppose a design of a context-aware system will include all possible trajectories of different context evolutions. In

the activity based of computer users we could have a trajectory representing the evolution of activity of writing a program, say activity A, and another activity representing writing a research paper, using a text editor, say activity B. If the user started a new activity C, which is using a new tool to write a research paper instead of a text editor, the system should automatically classify this new activity as being more similar to the trajectory of writing a paper on text editor than the trajectory of developing a program. Even if the system fails to indentify the new context at first, on-line learning could come as a next step and classify the new activity C as being closer to the activity B than to A.

VI. CONCLUSIONS AND FUTURE WORK

The work introduces motion in conceptual spaces as a tool to represent and reason about evolving contexts in context-aware applications. The proposed approach generalizes the use of motion classes to conceptual spaces. We intend to implement a middleware that supports context-aware application development based on the framework proposed here.

REFERENCES

- [1] H. B. Christensen, J. Bardram, “Supporting Human Activities- Exploring Activity-Centered Computing” in Proc. of Ubicomp 2002 Ubiquitous Computing, Gothenborg, Sweden, 2002, pp 107-116.
- [2] A. G. Cohn, B. Bennett, J. Gooday, and N.M.Gotts “Qualitative Spatial Representation and Reasoning with the Region Connection Calculus”, *Geoinformatica*, 1997, 1(3), pp. 1-42.
- [3] J. Coutaz et al., “Context is key” *Communications of the ACM*, 48(3), pp. 49-53
- [4] S. Dietze, A. Gugliotta, and J. Domingue, “Towards Context-aware Semantic Web Service Siscovery through Conceptual Situation Spaces”, in Proc. of the 2008 WWW 2008 International Workshop on Context Enabled Source and Service Selection, Integration and Adaptation, ACM New York, NY, USA, 2008.
- [5] P. Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. The MIT Press, Cambridge, 2000.
- [6] P. Gärdenfors and M.A. Williams “Reasoning about Categories in Conceptual Spaces”, in Proc. of the 17th International Joint Conference on Artificial Intelligence, 2001, pp. 385–392.
- [7] L. Ickja and B. Portier. “An Empirical Study of Knowledge Representation and Learning within Conceptual Spaces for Intelligent Agents”, Proc. of the 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007), pp.463-468
- [8] C. Keßler and M. Raubal, “Towards a Comprehensive Understanding of Context in Conceptual Spaces.”, Workshop on Spatial Language in Context – Computational and Theoretical Approaches to Situation Specific Meaning. Workshop at Spatial Cognition 2008.
- [9] I. Lee, “Fast Qualitative Reasoning about Categories in Conceptual Spaces”, HIS 2004, IOS Press, Melbourne, Australia, pp. 341-350
- [10] D. Randel, Z. Cui, and A. Cohn, “A Spatial Logic Based on Regions and Connection”, In Proc. of the International Conference on Knowledge Representation and Reasoning, 1992, pp. 165-176.
- [11] M. Raubal, “Formalizing Conceptual Spaces”, Formal Ontology in Information Systems, In Proc. of the 3rd International Conf. Frontiers in Artificial Intelligence and Applications 114, 2004, pp. 153-164.
- [12] A.Y. Tawfik, “On Middleware Support for Motion Awareness”, In Proc. of the 3rd IET Conference on Intelligent Environments (IE’07), 2007, pp. 89-95.