Immune Inspired Context Memory

Philipp H. Mohr
Computing Laboratory
University of Kent
Canterbury, UK
phm4@kent.ac.uk

Jon Timmis
Computing Laboratory
University of Kent
Canterbury, UK
jt6@kent.ac.uk

Nick Ryan
Computing Laboratory
University of Kent
Canterbury, UK
nsr@kent.ac.uk

ABSTRACT

This paper presents a context processing system, which stores context in an appropriate data structure and can provide a selective context history to a range of applications. Artificial Immune System algorithms are used to achieve data reduction, continuous online learning and forgetting of obsolete context.

Author Keywords

context history, context memory, artificial immune systems

INTRODUCTION

In order to set the scene for the content of the paper we first clarify our working dependencies:

> *Context* is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [1].
>
> *Context history* is the total collection of recorded past context.
>
> *Context memory* is a mechanism for retaining and recalling interesting and relevant past experience.

The objective of our research is to build a context processing system which only stores *relevant* context in a context memory. The stored context is used to provide context-aware applications with a selective context history, or context memory; we believe such a system should fulfil the following requirements:

- *Minimal amount of storage:* Storing all context is intractable [7, 2], therefore the amount of stored context data needs to be kept to a minimum; by removing duplicate context data and 'forgetting' potentially obsolete context data we construct a context memory (i.e., a *selective* context history).
- *Layered design:* The system should be designed with a layered approach (see Figure 1) which provides services and information to a number of applications.
- *Episodic memory:* Relationships between consecutive events need to be highlighted [6].
- *Context data should be smoothed:* When the users' behaviour changes, the systems' perception of the users' common behaviour should change gradually, as a sudden change is not desirable [11]. The system should also not be affected by noisy context data.
- *Ubiquitous environment:* The system needs to be made available on small, portable, resource-constrained devices and it needs to work in a range of networking environments with the real possibility that it must spend a proportion of time working with no connectivity.
- *Every day environments:* In order for the system to diffuse into every day environments, the user should not be required to be an active part of the system.

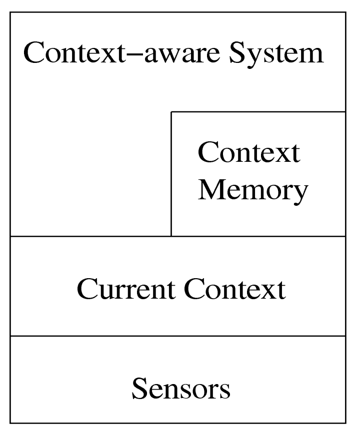

Figure 1. Layers.

Our proposal focuses on context data of an individual person. In order to create such a context memory we capitalise on techniques developed in the field of Artificial Immune Systems, as we believe they fulfil the above requirements. A widely accepted definition of an Artificial Immune System (AIS) is:

> Artificial immune systems are adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving [3].

For a system to qualify as an AIS it is required to have a minimum level of immunological inspiration incorporated, such as a model to perform pattern matching. AISs are often associated with virus detection, but their strengths are further reaching: they can perform pattern recognition, data compression, supervised and unsupervised learning, and be used to construct specialised memory structures. Calling something immunological does not make it an AIS. For a detailed introduction to AIS see [3] and [4].

In our context processing system we make use of AIS memory mechanisms, in particular Artificial Recognition Balls (ARBs) [10] which enable us to perform data reduction. A detailed explanation of ARBs is given in Section ; other AIS techniques relevant to our work are discussed in a previous paper [9].

Section describes our context processing system and Section presents our conclusions.

CONTEXT AWARE IMMUNE SYSTEM

In order to explain our Context Aware Immune System (CAIS) we define the *representation* of the components, the *affinity measures* which are used to quantify the interactions of the elements, and the *algorithm* which governs the behaviour of CAIS.

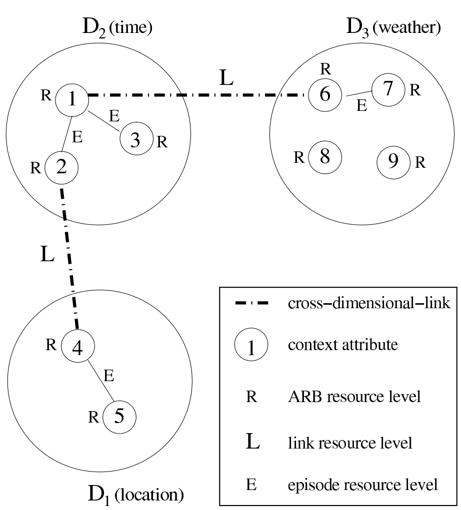

Figure 2. Data structure.

Representation:

The input to and output from CAIS is context represented by an attribute vector, $\langle a_1, a_2, ..., a_n \rangle$, which contains attributes (which can appear in an arbitrary order) along with their attribute identifier; for example:

```
⟨ Location.Building = Library,
  Wlan.MacAddress = 0A:40:C3:8D:00:32,
  Time.Hour = 18:30,
  Activity = Meeting ⟩
```

Where `Location.Building`, `Wlan.MacAddress`, `Time.Hour`, and `Activity` are attribute identifiers, and `Library`, `0A:40:C3:8D:00:32`, `18:30`, and `Meeting` are their respective values.

In order to store the attributes CAIS makes use of ARBs, which enable CAIS to perform data reduction since an ARB represents all elements in a region (in our case a 'region', and hence an ARB, is a hyper-sphere) eliminating the need for repetition. For example, a place of interest can be represented by an ARB instead of all individual GPS co-ordinates which fall within its radius; we associate an ARB with a resource level R, which increases when data points fall within its radius, and decreases by a constant decay. The radius r is a function of the resource level: $f(R) = r$.

The memory is an n-dimensional hierarchical network structure, where each dimension represents a different class of attributes. The attribute identifiers in our above example indicate to which dimension an attribute belongs, for example `Time.Hour` indicates that `18:30` belongs to the dimension 'time'. Figure 2 shows an example of the prototype data structure. The example consists of three dimensions and nine ARBs (drawn as small numbered circles). ARBs from different dimensions which appear in the same context are connected by cross-dimensional-links — this mechanism exploits the information present in the relations between context attributes. Furthermore, these links have a resource level L associated with them which reflects the likelihood that these two attributes occur in the same context. In our example ARB_4 and ARB_2 link dimensions D_1 and D_2, and ARB_1 and ARB_6 link dimensions D_2 and D_3. Furthermore, every dimension itself contains a network structure, for example dimension D_1 contains ARB_4 and ARB_5. In this structure, ARBs are connected with each other to highlight the relationship between consecutive events, which enables us to create an episodic memory. These links are associated with a resource level E to reflect the likelihood of certain events happening after each other.

Affinity Measures:

Every dimension has an affinity measure associated with it, which is used to determine the similarity between elements belonging to this dimension — for example, we use Euclidean distance to determine the simlilarity between GPS co-ordinates.

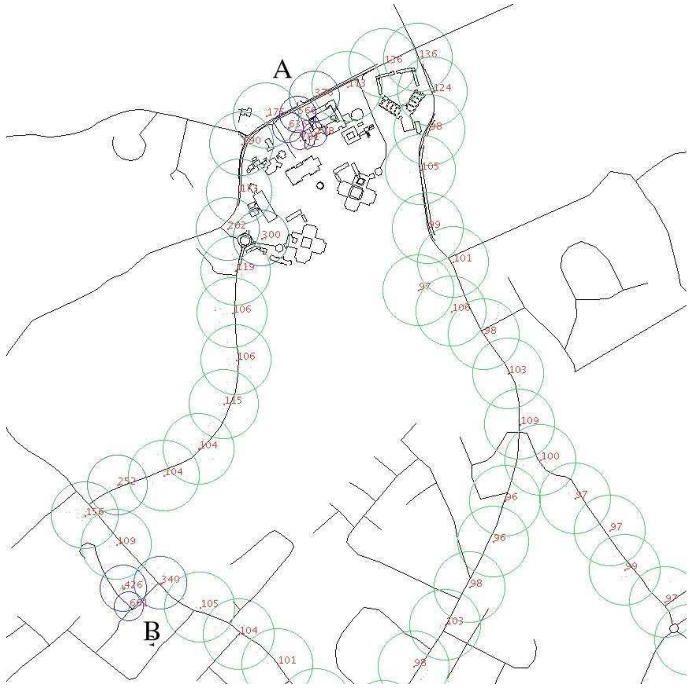

Figure 3. A snapshot of context memory, where the 65 circles represent ARBs. The smaller circles close to A and B show that these areas are stored with a high granularity.

Algorithm:
The algorithm used by CAIS is based on the principles of unsupervised and reinforcement learning and uses a combination of the representation and affinity measures described above; a suitable definition of unsupervised learning is:

> In unsupervised learning or clustering there is no explicit teacher, and the system forms clusters or "natural groupings" of the input patterns. "Natural" is always defined explicitly or implicitly in the clustering system itself [5].

Reinforcement learning can be defined as:

> Reinforcement learning addresses the question of how an autonomous agent that senses and acts in its environment can learn to choose optimal actions to achieve its goals. Each time the agent performs an action in its environment, a trainer may provide a reward or penalty to indicate the desirability of the resulting state [8].

Unsupervised learning allows us to construct a system which can cluster input data without any prior knowledge of the classes of the data. Reinforcement learning requires feedback from a trainer. However, an explicit trainer is not present in most context-aware systems, therefore an ARB in our system receives positive feedback when context attributes fall within its radius, and negative feedback is introduced by the notion of 'forgetting', which gradually decays all resource levels. For example, locations a user visits often have their rating reduced, but every visit increases the rating, which enables these locations to remain in the system. This also enables the system to reduce obsolete data, making the amount of data stored more manageable. Smoothing of context data is achieved by stimulation and decay of context attributes. The minimum amount of time an attribute remains in the system is controlled by the decay mechanism.

Having explained the individual components of CAIS, we now explain how they fit together. When CAIS receives an attribute vector the context processing algorithm selects the first attribute a and searches for the dimension it belongs to. In Figure 2 this might be attribute 1, which belongs to dimension D_2. Having found the dimension, it searches it for an ARB— using the appropriate affinity measure— which already covers the area of a; if such an ARB is found it is stimulated, which results in an increase in its resource level, R. If no such ARB exists, a new ARB is centered at a and initialised with a default resource level. The system then checks if the stimulated or newly created ARB matches the criteria for being part of an episode (e.g. if it is encountered shortly after the previously stimulated one); if it does, an existing link to the previous ARB is stimulated (e.g. the link between 4 and 5) or, if one does not exist, a new link is created with a default resource level, E. This process is repeated with all attributes in the vector. Next the system searches for existing links between attributes from this vector (e.g. cross-dimensional-link between 2 and 5). If links exist they are stimulated, otherwise new ones are created with a default resource level, L. All ARBs and all links between them are subject to a decay mechanism to control the population.

If an outlier is chosen as the center of an ARB, this ARB will not receive enough stimulation and will die out, as the loss of resources due to the decay mechanism is higher than the resource gain. If the centre of an ARB is a sub-optimal center for the area which is covered by this ARB, the centre needs to be shifted. This is achieved by the stimulation and decay mechanisms. For example, if a person spends most of his time in his office, but the initial ARB covering the area of the building is created using the entrance as the centre of the ARB, the ARB should be shifted to a centre close to the persons' office. This shift happens when the initial ARB is stimulated to such a degree that it does not cover the area of the office any more, as the next location recorded outside this region, which is most likely to be close to the office, will form the centre of a new ARB. The new ARB will quickly gain resources as it is stimulated by the high activity in the area of the office, and the initial ARB will grow again as the activity in the area of the office now stimulates the new ARB. This process happens gradually over time and highlights the dynamic nature of CAIS.

A context-aware application may need access to current context, but also to context memory in order to identify recurring situations. As shown in Figure 1 a context-aware application may either query the underlying current context or the context memory [12]. If it requires current location data, for example to predict a route, it would obtain matching previous context data from the context memory.

Figure 3 presents preliminary results with a small subset of GPS co-ordinates. The data set consists of 3000 GPS data points which where collected by a single person over a period of 46 days. In the experiment we simulate this period by iterating through the data set. The 3000 data points are reduced by the algorithm to 65 (not all are visible in the figure), showing its data reduction capabilities. A large proportion of the data points where collected around position A and a slightly smaller proportion around position B, this is reflected in Figure 3 by small circles — representing a high data granularity. Less frequently visited areas and commonly traced routes have a lower data granularity and are reflected by larger circles.

CONCLUSION

Context memories have a great potential to improve context-aware systems, but constructing context memory structures or selective context histories is not straightforward. Such systems should be usable by multiple applications, therefore they need to be designed in such a way that they are generic enough to serve different applications. In an ideal world we want context processing systems to be able to generalise, be adaptable, and be able to compress or reduce the amount of data stored; immune inspired algorithms are good candidates, as they offer us mechanisms to fulfil all the requirements.

In this paper we propose an immune inspired context memory, which can adapt to a wide variety of user behaviour and environmental inputs. This is achieved by using immunological metaphors and immune inspired algorithms combining unsupervised and reinforcement learning.

REFERENCES

1. Abowd, G.D. and Dey, A.K. Towards a better understanding of context and context-awareness. http://www.cc.gatech.edu/fce/contexttoolkit/chiws/Dey.pdf
2. Aizawa, K., Hori, T., Kawasaki, S. and Ishikawa, T. Capture and efficient retrieval of life log. In *Proc. Pervasive Computing 2004 workshop on memory and sharing of experiences*, 15–20.
3. de Castro, L. and Timmis, J. Artifcial Immune Systems: A New Computational Approach. *Springer-Verlag, London. UK.* (2002).
4. de Castro, L. and Timmis, J. Artifcial Immune Systems as a Novel Soft Computing Paradigm. In *Soft Computing 7* (2003), 526–544.
5. Duda, R.O., Hart, P.E., Stork, D.G. Pattern Classifcation. *John Wiley & Sons Inc* (2000).
6. Hart, E., Ross, P., Webb, A. and Lawson, A. A Role for Immunology in "Next Generation" Robot Controllers In *Proc. Second International Conference on Artifcial Immune Systems*, LNCS Springer (2003), 46–56.
7. Harvel, L.D., Liu, L., Abowd, G.D., Lim, Y., Scheibe, C. and Chatham, C. Context cube: Flexible and effective manipulation of sensed context data. In *Proc. Pervasive Computing 2004*, LNCS Springer (2004), 51–68.
8. Mitchell, T.M. Machine Learning. *McGraw-Hill, New York* (1997).
9. Mohr, P.H., Ryan, N. and Timmis, J. Exploiting Immunological Properties for Ubiquitous Computing Systems. In *Proc. Third International Conference on Artifcial Immune Systems*, LNCS Springer (2004), 277–289.
10. Perelson, A.S. Immune network theory. In *Imm. Rev.* (1989), 5–36.
11. Petzold, J., Bagci, F., Trumler, W. and Ungerer, T. Global and local state context prediction. In *Artifcial Intelligence in Mobile System* (2003).
12. Ryan, N. Smart Environments for Cultural Heritage. In *Reading Historical Spatial Information from around the World, Proc. 24th International Research Symposium* (2005), 71–80.

Philipp H. Mohr received his BSc Honours in Computer Science from the University of Kent in 2003. He is currently a PhD student in the Pervasive Computing and Applied and Interdisciplinary Informatics Group at the University of Kent under his supervisors, Nick Ryan and Jon Timmis.