Context Addressing using Context-Aware Flooding

Koen Victor
KULeuven
Dept of Computer Science
Celestijnenlaan 200A
Leuven, Belgium
Koen.Victor@cs.kuleuven.be

Julien Pauty
KULeuven
Dept of Computer Science
Celestijnenlaan 200A
Leuven, Belgium
Julien.Pauty@cs.kuleuven.be

Yolande Berbers
KULeuven
Dept of Computer Science
Celestijnenlaan 200A
Leuven, Belgium
Yolande.Berbers@cs.kuleuven.be

ABSTRACT

Due to the proliferation of small networked mobile devices, the number of (indirectly) interconnected services in pervasive computing environments may grow without bound. The network contains a potentially enormous amount of context aware services that sense, gather and distribute context information. Without a central context repository, or a central server that locates the context information, it is a challenge to address parts of the environment that contain relevant context information. In this paper, we propose a model for addressing context and an algorithm for context gathering and distribution that imposes a virtual structure on the network, that aligns with the actual context information within a pervasive computing environment. Distribution of context uses an adapted form of flooding, that is context aware. Our evaluation shows that the algorithm performs substantially better than bounded flooding, if address accuracy is at least 30%.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

context-aware computing, pervasive computing

1. INTRODUCTION

In a pervasive computing environment, numerous services make context information available to the environment. This context information is used by services to enhance their operation by adapting their behavior. Context information may guide service adaptation and service composition, help service configuration or trigger service events.

Context gathering and distribution in such an environment is a difficult task. The network is a combination of structured and unstructured networks that have both fixed and mobile services. If there is no overlay network or central query service available, there is no global addressing scheme to access services or their content. Additionally, devices may enter or leave the network without notice and do not have any knowledge of the environment in advance.

To tackle these problems, we structure the network using the context information itself, and do not use the addressing scheme of the underlying network. We rely on the type of the context data that is distributed in the network. Context data is encapsulated in a Context Item [2] that adheres to a type system. To distribute or retrieve a Context Item, we use flooding, but only nodes for which the Context Item is relevant store and distribute it further. To determine the relevance of a Context Item, it is associated with Distribution Requirements [2] that describe its destination.

If the context information arrives in a part of the network where it is not relevant following the Distribution Requirements, the distribution stops. We also allow the relevance of a Context Item type to decrease gradually, to make context data travel within bounds through parts of the network where the Distribution Requirements are not fulfilled. To query for a Context Item, we express its address using a description of its context.

2. RELATED WORK

Cohen et al. [1] focus on the architecture for a context distribution system. A producer-consumer approach is used to disseminate information. A simulation is performed to compare different distribution algorithms, however none of them takes the approach of addressing the relevant part of the network using Context Items. Sygkouna et al. [4] study efficient decentralized usage-aware search mechanisms. The approach taken in this paper is using context usage patterns to restructure the context in the network. T. Gu et al. [5] describe the use of multiple overlays to cluster peers in a P2P-network based on predefined ontologies to provide an efficient decentralized search engine for context data. Methods to address context usually are publish/subscribe systems [3]. Also, in several domains, tuple spaces that support keyword pattern matching are used to acquire context information [6].

3. ALGORITHM AND EVALUATION

Context-Aware Flooding is based on regular bounded flooding. On every node that receives a request for context, the algorithm determines the relevance of the request on that node, by comparing the distribution requirements against the context information on that node. This involves a local search for context. Distributed relation checking is achieved

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil

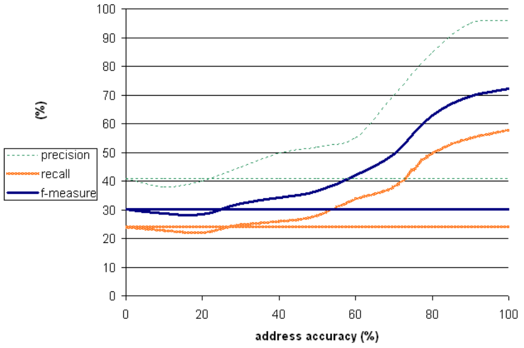

Figure 1: Effectiveness of Addressing

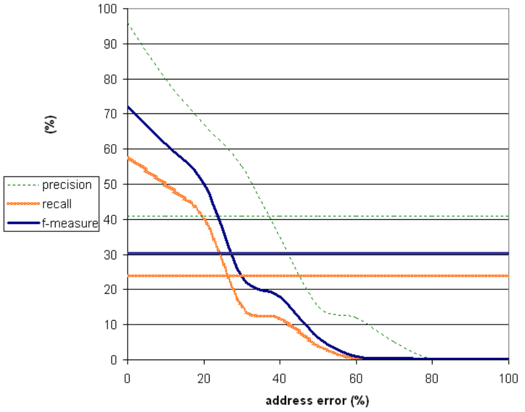

Figure 2: Error sensitivity of Addressing

through distributing requests that are not relevant for a limited amount of hops. In that way, requests traveling in a non relevant environment are in time removed from the network. Relevant requests are able to travel until a valid answer to the context request is found.

We evaluate the following features of the context addressing and the context-aware flooding algorithm: address effectiveness, address error sensitivity and performance of the algorithm. We use two ratios for the address effectiveness and the address error sensitivity that are well known in the domain of information retrieval: recall and precision.The f-measure is the weighted harmonic mean of precision and recall. Figure 1 and 2 show the results of the experiments on effectiveness and error sensitivity. The straight lines are the results measured with regular bounded flooding. All tests are performed on an unstructured network implemented in Omnet++.

Figure 1 shows a substantial improvement in recall and precision in comparison with bounded flooding. The improvement of recall shows that the algorithm is able to guide the request to the relevant part of the network. The improvement of precision shows that the algorithm is able to capture context relations across different hosts. We see that when address accuracy is low, the algorithm performs equal or slightly worse than bounded flooding. Especially precision suffers from inaccurate address specification. As expected, an address accuracy of 0 % provides similar results as bounded flooding. Figure 2 shows that when an error is introduced, precision drops quickly from 80% to below the performance of the bounded flooding algorithm. This is due to the fact that in the algorithm, there can only be a match to a request if the specification is exactly fit, or at least four parts of the specification match. As a result, the algorithm rejects many satisfactory Context Items, and tends to favor areas of the network that do not match the specification.

We conclude that the the algorithm is prone to address errors, and less to incomplete addresses. Therefore, if an address is not well known, one should rather leave out bogus specifications instead of providing a ‘complete’ address that contains errors. Starting from an address accuracy of 30 %, specifying an address and using the context aware flooding algorithm generates more accurate and precise results.

For each of the above tests, we performed measurements on the number of packets. For measurements with low f-measure results, the number of packets is substantially lower (8 to 15%) than the flooding algorithm. As such, when a correctly specified address has no corresponding Context Item in the network, less packets will be ‘wasted’. However, measurements with high f-measure results only benefit slightly from this algorithm.

4. CONCLUSIONS

We present a way to address context in non structured networks using the context features of the environment. The algorithm shows a substantial improvement in recall and precision in comparison with bounded flooding. The algorithm is prone to address errors, but resistant to incomplete addresses. From an address accuracy of 30 %, specifying an address and using the context aware flooding algorithm generates more accurate and precise results than bounded flooding.

5. REFERENCES

[1] R. Cohen and D. Raz. An open and modular approach for a context distribution system. *Network Operations and Management Symposium, 2004. NOMS 2004*, 19-23 April 2004.

[2] Y. B. K. Victor, J. Pauty. Context distribution using context aware flooding. *1st ERCIM Workshop on eMobility*, 2007.

[3] A. P. Seng Wai Loke and A. Zaslavsky. Context-based addressing: The concept and an implementation for large-scale mobile agent systems using publish-subscribe event notification. In *Distributed Applications and Interoperable Systems Volume*, volume 2893/2003 of *Lecture Notes in Computer Science (LNCS)*. Springer-Verlag, 2004.

[4] M. S. E. Sygkouna, I.; Anagnostou. Efficient search mechanisms in a context distribution system. *Advanced Information Networking and Applications, 2006. AINA 2006*, 18-20 April 2006.

[5] E. T. T. Gu, H. K. Pung, and D. Q. Zhang. Contextpeers: Scalable peer-to-peer search for context information. *1st International Workshop on Innovations In Web Infrastructure*, 2005.

[6] Y. Z. Xinhuai Tang and J. You. Rcacm: Role-based context-awareness coordination model for mobile agent applications. In *Lecture Notes in Computer Science*, pages 702–705. Springer Berlin / Heidelberg, 2004.