Anand Ranganathan
Research Statement

1. Introduction

The main focus of my research is to try to reduce the complexity of large-scale pervasive or ubiquitous computing environments for developers, administrators and end-users. The complexity of developing, managing and using these environments has been one of the main bottlenecks in the widespread deployment of pervasive computing. In the past, I have approached this problem by using techniques from various fields like autonomic computing, location-aware and context-aware computing, the Semantic Web and ontologies, machine learning, AI planning, middleware, programming frameworks and workflows. In the future, I would like to explore other mechanisms to improve the programmability, manageability and usability of pervasive computing environments, in specific, and large distributed systems, in general.

Pervasive Computing advocates the enhancement of physical spaces with computing and communication resources that help users perform various kinds of tasks. Over the past few years, different pervasive computing scenarios and applications have been proposed and various prototype systems supporting these scenarios have been built and evaluated. However, so far, we have not seen widespread deployment of pervasive computing environments. One reason for this is their complexity. These environments feature large distributed systems containing a variety of inter-dependent services and devices. They are highly dynamic and fault-prone as well. Besides, different environments can vary widely in their architectures and the resources they provide. Hence, developers find it difficult to program new applications and services in these environments; administrators find it difficult to manage and configure these complex, device-rich systems; and end-users find it difficult to use these environments to perform tasks.

Solving the complexity problem requires a broad set of solutions. Firstly, pervasive computing environments need to be self-managing and self-repairing so as to reduce the burden on administrators and end-users. They must offer developers facilities to simplify the development of new applications and services through the use of middleware and programming frameworks. They must be smarter in their interaction with humans, by being aware of the context and location of users, and by providing mechanisms that help users perform various tasks more efficiently. In my research, I have explored various techniques that achieve these goals. In the future, I would like to continue exploring more solutions that will help propel the deployment of pervasive computing environments.

2. Past and Current Research

For the past four years, I have been involved in the Gaia project[8] at UIUC that aimed at building an infrastructure for Active Spaces, which are physical spaces that have been enriched with devices and services to become interactive, pervasive computing environments. My thesis has focused on building a framework for autonomic pervasive computing, that enables these environments to be self-managing and adaptive and to require minimal user intervention. The framework allows developers and administrators to program and configure these Active Spaces in terms of high-level tasks. A task is a set of actions performed collaboratively by humans and the pervasive system to achieve a goal. It is modeled as a flowchart involving one or more user interaction, communication and computation steps. In a smart conference room, for example, common tasks could include displaying slideshows, collaborating with others on one or more documents, and migrating applications from one's personal device to public devices.

Each task is associated with many parameters that influence how it is performed. These parameters are either behavioral parameters, which describe which algorithm or strategy is to be used for performing the task, or resource parameters, which describe which resources are to be used. For example, even the relatively simple task of displaying a slideshow has a number of parameters like the devices and applications to use for displaying the slides, the devices and applications to use for navigating the slides, the name of the file, etc. These parameters may be set manually by the end-user, pre-specified by the developers or the administrator, or automatically deduced by the framework based on the current state of the environment, context-sensitive policies, and end-user preferences. User preferences may be either explicitly specified by the user or learned by the framework based on past user behavior. Hence, a task can be executed entirely automatically (where the environment decides all the parameter values), entirely manually (where the end-user chooses all parameters), or anywhere in the space in-between. The framework thus enables autonomic behavior by enabling self-configuration – it frees developers, administrators and end-users from the burden of choosing myriad parameter values for different circumstances for performing their task, although it does allow them to override system choices and manually configure how the task is to be performed.

The framework also uses AI planning to compose actions and tasks to achieve high-level goals [1]. It uses a combination of machine-learning and policies specified in Prolog, along with a multi-dimensional utility function, to deduce the best values of various parameters. It can also recover from failures of one or more actions by re-trying equivalent actions or by re-planning.

A key part of the pervasive autonomic framework is Olympus [6], a high-level programming model for developing pervasive applications and services. The model allows developers to write programs that describe Active Space entities (which include services, applications, devices, physical objects, locations, users and Active Spaces) at an abstract, high level. It resolves these high-level descriptions of Active Space entities into actual entity instances based on constraints specified by the developer, the resources available in the current space, space-level policies and the current context of the space. Developers can also specify common Active Space operations at a high level. Examples of operators include starting, stopping and moving components. Thus, developers do not have to worry about how these operations are performed in the space in which their program is to be deployed. These details are taken care of by the model and the developer is free to focus on the actual logic of the program.

Our experiments and user studies have shown that this framework does make it easier to program, configure and use Active Spaces. The framework has been used by various members of my research group to develop new tasks for different prototype Active Spaces in our CS Building.

The framework uses ontologies to express domain knowledge and enable knowledge sharing and reuse. I have developed ontologies in OWL that describe different aspects of Active Spaces, including applications, services, devices, context information, users, data formats, etc.[6]. These ontologies allow developers and administrators to be aware of the different concepts used in Active Spaces and the relationships between these concepts. I have also used these ontologies to perform semantic discovery and matchmaking [7], which is more powerful than conventional syntactic discovery.

The framework is supported by a context infrastructure that allows modeling context, sensing and inferring different types of contexts, reasoning about uncertain contexts [2,3]. The infrastructure simplifies the development of various context-aware applications and context sensors [4,5]. We also used context to enhance various security processes in Gaia, including authentication and access control [9]. An important piece of context is the location of users, devices and other physical objects in the environment. I worked on developing a middleware that simplified the development of location-aware services and applications. This middleware, called MiddleWhere [11] used probabilistic reasoning to fuse sensor data from different location sensing technologies and to infer spatial relationships between mobile objects and the environment. MiddleWhere has been used extensively within our group for developing a variety of location-aware applications.

Another area of my research is the use of workflows to model and manage a user's interaction with the environment [10]. The prototype used the popular business workflow language (BPEL) to model various processes in pervasive environments as workflows. This approach improved the usability of these environments, while also increasing flexibility in changing the model of interaction without having to touch individual services and applications.

3. Future Directions

In the future, I would like to continue working on ways to make pervasive computing environments easier to develop, manage and use. In particular, I would like to explore pervasive computing from the viewpoint of multi-agent systems. Real-life pervasive computing environments in public-spaces like airports and malls will contain different users and groups with possibly conflicting goals. In order to ensure the smooth functioning of these environments, mechanisms that allow negotiation and conflict-resolution between different agents have to exist. Most current environments assume the existence of a single user or of a single group of users, in which all members cooperate to achieve a single goal. I would like to explore how competing users can use common resources in a pervasive computing environment. I plan to incorporate ideas from game theory and other economic mechanisms like auctions to tackle these problems.

In addition, I would also like to expand the scope of my research to other large distributed systems such as grids, sensor networks, internet-based systems and P2P systems. All these domains suffer from similar problems related to the complexity of developing, managing and using them. Hence, these systems must also be made self-configuring, self-managing and self-healing. I am interested in studying how these systems can be made more autonomic with the help of programming frameworks, middlewares and user-interfaces that allow developers, administrators and end-users to interact with these systems at a high-level, while letting the system automatically take care of low-level details.

Another area I would like to explore is inter-operability between different autonomous systems, and more specifically, between different pervasive computing environments. Different systems may use different terms to describe the same concept, may have different architectures and may follow different algorithms and strategies to perform the same task. Hence, applications and services written for one system often cannot run in other systems and cannot inter-operate with applications and services in other systems. One solution is standardization, though it is often difficult to achieve. Another decentralized solution is to develop and discover mappings and relationships between different systems, either manually or automatically. The use of ontologies to describe domain knowledge helps in enhancing inter-operability. However, automatically discovering mappings between different ontologies is still a hard problem. Besides, there has not been much work in using discovered mappings between ontologies to allow systems to inter-operate seamlessly and in resolving conflicts between policies defined for different systems. Hence, new solutions that allow semantic interoperability between different systems are required.

Finally, critical areas of pervasive computing that have not been addressed so far are metrics and measurement. We need ways of evaluating different aspects of pervasive computing such as architectures, programming frameworks, usability, security and context-awareness. I would like to undertake efforts to develop and standardize metrics and benchmarks that can be used to assess disparate pervasive systems.

In summary, I believe that as systems become larger and more powerful, they also become more complex. Hence, we need techniques that alleviate the complexity of systems for developers, administrators and end-users. I intend to combine solutions from various fields, including computer systems, artificial intelligence, human-computer interfaces and software engineering, to develop techniques that enable large systems to be autonomic.

References

1. Anand Ranganathan, Roy H. Campbell, “Pervasive Autonomic Computing Based on Planning”, In *IEEE International Conference on Autonomic Computing (ICAC 2004)*, New York, NY, May 17-18, 2004
2. Anand Ranganathan, Roy H. Campbell, “A Middleware for Context-Aware Agents in Ubiquitous Computing Environments” *ACM/IFIP/USENIX International Middleware Conference*, Rio de Janeiro, Brazil, Jun 16-20, 2003.
3. Anand Ranganathan, Jalal Al-Muhtadi, Roy H. Campbell, “Reasoning about Uncertain Contexts in Pervasive Computing Environments”, In *IEEE Pervasive Computing,* pp 62-70, Apr-June 2004
4. Anand Ranganathan, Roy H. Campbell, Arathi Ravi, Anupama Mahajan, “ConChat: A Context-Aware Chat Program”, In *IEEE Pervasive Computing*, pp. 52-58, July-Sept 2002.
5. Hui Lei, Anand Ranganathan, “Context-Aware Unified Communication”, In *IEEE International Conference on Mobile Data Management (MDM 2004)*, Berkeley, California, Jan 19-22, 2004
6. Anand Ranganathan, Shiva Chetan, Jalal Al-Muhtadi, Roy H. Campbell, M. Dennis Mickunas, “Olympus: A High-Level Programming Model for Pervasive Computing Environments”, To appear in *IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)*, Kauai Island, Hawaii, Mar 8-12, 2005.
7. Anand Ranganathan, Shiva Chetan, Roy Campbell, “Mobile Polymorphic Applications in Ubiquitous Computing Environments” In *Mobiquitous 2004* : The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston, Massachusetts, Aug 22-25, 2004
8. Manuel Román, Christopher Hess, Renato Cerqueira, Anand Ranganathan, Roy Campbell,Klara Nahrstedt, “Gaia: A Middleware Infrastructure to Enable Active Spaces”, *IEEE Pervasive Computing*, pp 74-83, Oct-Dec 2002.
9. Jalal Al-Muhtadi, Anand Ranganathan, Roy H. Campbell, Dennis Mickunas, “Cerberus: A Context-Aware Security Scheme for Smart Spaces*”,* In *IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, Dallas-Fort Worth, Texas, Mar 23-26 2003.
10. Anand Ranganathan, Scott McFaddin, “Using Workflows to Coordinate Web Services in Pervasive Computing Environments”, In *2nd International Conference on Web Services (ICWS 2004),* San Diego, CA Jul 6-9, 2004
11. Anand Ranganathan, Jalal Al-Muhtadi, Shiva Chetan, Roy Campbell, M. Dennis Mickunas, “MiddleWhere: A Middleware for Location Awareness in Ubiquitous Computing Applications” In *ACM/IFIP/USENIX 5'th International Middleware Conference,* Toronto, Canada, Oct 18-22, 2004