What Makes a Good Pervasive Middleware?

Fahim Kawsar, Kaori Fujinami, Tatsuo Nakajima
Department of Computer Science, Waseda University, Japan
{fahim,fujinami,tatsuo}@dcl.info.waseda.ac.jp

ABSTRACT
We present 3D Evaluation Model for pervasive middlewares with dimensions of Pervasive Environment specific, Quantitative and Qualitative metrics. We expect our approach is viable to evaluate pervasive middleware since it attempts to cover diverse aspects of ubiquitous computing. In this short paper, we have briefly introduced our 3D Evaluation Model and put forth this issue for open discussion in the poster session.

Keywords
Pervasive, Middleware, Evaluation, Context Awareness.

1. INTRODUCTION
Perhaps one of the difficult tasks of the pervasive middleware developers is to evaluate the middleware. This is because of two reasons: First, the standard computing system evaluation methodology is not applicable to pervasive middlewares considering the diverse feature attributes of ubicomp inherited from different domains with different evaluation schemes. Second, there is no standard guideline for evaluation. Several attempts have been made so far by the research community to guide the evaluation phase of ubicomp systems. But unfortunately most of those guidelines are targeted to the ubiquitous applications. So, it is not clear what approaches are important to evaluate the system that provides the base support for the development of these applications. In this paper, we have targeted this particular aspect and present a 3D Evaluation Model for pervasive middleware with dimensions of Pervasive Environment specific, Quantitative and Qualitative metrics. Each of these metrics attempts to evaluate specific aspects of ubiquitous middleware. In the next section, a snapshot of the efforts done on the evaluation schemes for proactive applications is presented. Then we argue why these approaches fail to guide the middleware developers. Finally we present our 3D Evaluation Model with a peek into its immediate implication.

2. EVALUATION AND UBICOMP
One and a half decades ago Mark Weiser emphasized that real prototyping is one of the better alternatives for the evaluation of the ubiquitous systems [9]. Later Abowd and Mynatt [1] rephrased that real prototyping, as "Living Laboratory" is the appropriate evaluation approach for ubiquitous systems. Bellotti and her colleagues suggested five interaction challenges that the ubiquitous system designer should follow [2]. These are: **A**ddress, **A**ttention, **A**ction, **A**lignment and **A**ccident. Mainly she suggested these points to assist interaction designers to develop and evaluate systems that are not GUI based. Jameson proposed five usability challenges for adaptive systems: predictability and transparency, controllability, unobtrusiveness, privacy and breadth of experience [4]. From social aspect, Friedman et. al. suggested 12 key human values with ethical importance [3]: human welfare, ownership and property, universal usability, trust, autonomy, informed consent, accountability, identity, calmness and environmental sustainability. These key points basically cover all social aspects of human computer interaction and can be equally applied to evaluate ubiquitous systems from social aspect point of view. The outcome of the workshop "Evaluation Methodologies for Ubiquitous Computing" at ubicomp 2001 conference formulated a 4 axes framework: Universality, Utility, Usability and Ubiquity [8]. Another interesting proposition is Ubiquitous Evaluation Areas (UEA) proposed by Jean Scholtz et. al. [7]. Under UEA umbrella, they include the following areas for ubiquitous system evaluation: **A**ttention, **A**doption, **T**rust, **C**onceptual **M**odel, **I**nteraction, **I**nvisibility, **I**mpact and **S**ide **E**ffects, **A**ppeal and **A**pplication **R**obustness.

Considering all these propositions, it is understandable that key focus of ubiquitous system evaluation is on the user end rather than on the systems. This is in contrast to other computer system research where we explicitly focus on system performance by some benchmark tools. However, middleware is not the application instead it is a kind of tool that assists the application development. So setting up a stage story for real life prototyping is not strictly applicable to evaluate the performance of the middleware. It is true that all these key points in UEA and others propositions are recurring in nature in applications, but these are mostly end application designers responsibility where they have to consider the scenario in hand and how these requirements can be satisfied while providing the actual service. Strictly some of these requirements are not middleware specific, like: universality, ubiquity, attention or usability. So, middlewares' responsibility could be assisting application developers to provide all system related support so that developers can focus on the application level challenges considering user level metrics for pervasive applications.

3. 3D EVALUATION MODEL
In typical system middlewares, there are specific measures for evaluation; commonly we term that "Quality of Service (QoS)". Usually several quantitative and qualitative metrics are considered to evaluate the performance of the system. In ubiquitous systems, in addition to these measures we have to satisfy some human factor issues that are equally important as quantitative and qualitative metrics. Theses issues are recurring in nature, so support provision of these facilities is a requirement of ubiquitous middleware. Taking into account these factors, we propose a 3D Evaluation Model for evaluating ubiquitous middleware as shown in figure 1.

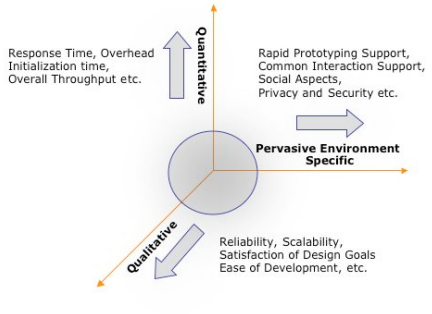


Figure 1: 3D Evaluation Model for Pervasive Middleware

3.1 Pervasive Environment Evaluation
This dimension is important because the goal of pervasive middleware is to support proactive application development. In proactive applications, many features are related to environment and interaction and are not common to typical computing system. The performance of these features contributes to the overall acceptability of the applications. So, supporting these recurring features is an inherent requirement for the quality of pervasive middlewares. Considering this, we have included following metrics in this evaluation dimension:

1. **Interaction:** The middleware should have support for recurring interaction mechanism support in a way not perceivable by the end users. For example, sensors are used extensively in pervasive applications, so middleware should provide a common base for supporting sensor based user interaction while minimizing application developers' tasks. This interaction

mechanism is also important to provide the calmness of the applications as introduced in [6].

2. **Support for Social Aspects:** Social aspect is a very broad domain and it is not possible to confine it by one or two metrics. But considering the evaluation propositions presented earlier, there are few social points that are system related: Environment Sustainability, Mental Model and Preference Management. Middleware should have proper support for facilitating these features in the applications easily without extensive development at application level.
3. **Privacy and Security:** Middleware should have minimal support for the application developer that allows them to incorporate privacy and security into their applications. At least middleware should have the responsibility for protecting the device end security.
4. **Support for Rapid Prototyping:** Real world application prototype must be developed on top of the middleware rapidly and should run in real environment to identify the performance of the middleware.

3.2 Quantitative Evaluation

Quantitative evaluation includes the traditional criteria that we use for analyzing computer systems. This includes but not limited to:

1. **Response Time**: How responsive the components of the middleware are?
2. **Initialization Time:** How much time and system resources do the middleware require to initialize?
3. **Communication Overhead:** The number of messages exchanged and latency between component communications.
4. **Overall Throughput:** Overall system performance considering the task underneath.
5. **Component Specific Unit Metrics:** Specific component may have specific metrics, for example if a middleware component use Bluetooth for communication, then Bluetooth specific quantitative metric contributes to this scheme.

3.3 Qualitative Evaluation

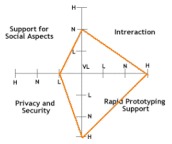
Following metrics are included for quantifying the quality of the middleware in the 3D Model:

1. **Reliability, Robustness and Fault Tolerance:** Every ubiquitous middleware should have self healing mechanism to adopt with the error conditions and should provide the flexibility to recover from failures without hampering the application running on top of it. There are unit measures for robustness like Mean Time Between Failure (MTBF) that can be considered here as a quality.
2. **Scalability:** It is important to consider, how scalable the middleware is in terms of applications that it can support simultaneously.
3. **Conforming Design Principles**: Pervasive Middlewares has some strict design requirements like: Abstraction, Separation of Implementation, Transparency, Dynamism in Discovery and Event Notification, Extensibility, Independence etc. It is important that the middleware explicitly satisfies each of these requirements.
4. **Ease of Use:** The difference between just good and widely successful middleware is how easily it lets the application developers exploit its various capabilities. It is important to consider that how fast developers can grasp the middleware concept. Also line of code for interfacing with middleware is a good indicator of easiness. So feedback from the application developers is very important for the evaluation of the middleware.

4. IMPLICATION

We have followed this model for evaluating the pervasive middleware called "Prottoy" [5] developed in our lab for supporting development of proactive application and observed some interesting results; we are enumerating here some of those:

1. **Pervasive Environment Specific:** Prottoy provides high-level abstraction to facilitate the sensor driven interactions. In addition it has component that provide automatic GUI generation and voice based interaction facilities for preference management. Furthermore, it supports device end security and allows rapid prototyping. If we impose the 4 metrics of this dimension into 4 axes with very low, low, normal and high values, we have anticipated the results as indicated in the right figure. This is because, Prottoy's useful support for interaction and rapid prototyping and minimal support for security and average support for social aspects.



2. **Quantitative Metrics:** Typical system related benchmark test have been performed. Acquired performance and comparison with related middlewares showed Prottoy's performance in this dimension is satisfactory.
3. **Qualitative Metrics**: Prottoy has some self-healing capability, and it strictly follows the design goals. Considering the feedback we have received from the application developers, Prottoy is found to be successful in minimizing developers' tasks. However, quantifying these metrics for conclusion is a long-term task. But from the evaluation aspect, we found that the clear distinction of quality metrics clarifies the performance issues that we have to monitor for further improvements if required.

For space constrain we cannot report the detail here. Please check [5] for elaborative results. We strongly believe our model is in initial stage and requires lot of feedback from the community for further improvement and amendment that we expect to receive from the participants of the Ubicomp 2006.

REFERENCES

[1] Abowd, G. D. et. al. , "Charting Past, Present and Future Research in Ubiquitous Computing". ACM Transactions 7(1): 29-58, 2000.

[2] Bellotti, V, et. al. "Making Sense of Sensing Systems: Five Questions for Designers and Researchers". The Conference on Human Factors and Computing Systems (2002)

[3] Friedman, B. et. al. "Human Values, Ethics, and Design," The Human- Computer Interaction Handbook, (2003)

[4] Jameson, A. "Adaptive Interfaces and Agents". The Human-Computer Interaction Handbook, New Jersey (2003)

[5] Kawsar F.; "An Architecture for Sentient Artefact based Computing"; M.Sc. Thesis, Department of Computer Science, Waseda University, Japan, February 2006.

[6] Riekki, J et. al. Evaluating the Calmness of Ubiquitous Applications", International Conference on Product Focused Software Process Improvement, 2004

[7] Scholtz, J. et al., "Towards a Discipline for Evaluating Ubiquitous Computing Applications," IRS-TR-04-004. 2004.

[8] Scholtz, J. et. al. Report from Ubicomp 2001 Workshop: Evaluation Methodologies for Ubiquitous Computing. SIGCHI Bulletin (Abowd, G., ed.).

[9] Weiser, M., "The Computer for the 21st Century", Scientific American, 265(3): 94-104, September 1991.