REFLECTION: A SOLUTION FOR HIGHLY ADAPTIVE MOBILE SYSTEMS

Christos Efstratiou, Keith Cheverst

Distributed Multimedia Research Group,
Department of Computing, Lancaster University,
Bailrigg, Lancaster, LA1 4YR, U.K.

telephone: +44 (0)1524 594337
e-mail: most@comp.lancs.ac.uk

EXTENDED ABSTRACT

Mobile systems are characterised by the fact that they operate in an environment prone to rapid and drastic changes [4]. More specifically, attributes like network quality of service (QoS) or power availability can vary significantly compared to non-mobile systems. Mobile applications should be capable of being informed of such changes and should also be able to adapt their behaviour accordingly in order to deliver the best level of service to the end-user. Moreover, future mobile systems should be capable of adapting to general environmental and contextual triggers such as the system's physical location [2]. As presented in [3], current middleware support lacks the required level of control for co-ordinating highly adaptive systems triggered by multiple contextual attributes.

Reflection has been suggested as a solution for adaptive middleware [1]. The benefits of being able to open the implementation of the platform, inspect and adapt the behaviour of the system have already been recognised. In this paper, we argue that a reflective approach is appropriate for the development of middleware support for mobile systems and the development of mobile applications themselves. Indeed, we argue that it is necessary to allow system wide adaptation in all parts of the system (middleware and applications) in order to enable appropriate arbitration for resources and thus enable a better level of resource utilisation to be achieved.

Research on the development of reflective operating systems [5] offers solutions for the design of highly adaptive computer systems. However, these efforts do not support any specific means for controlling context-based adaptive behaviour triggered by the dynamic change of actual user requirements.

The benefits of a reflective approach can be summarised as:

i) the ability to inspect the implementation of a system by accessing the meta-space that defines the operation of the system and

ii) the ability to alter the implementation during run-time in order to achieve specific requirements.

In terms of mobile computing this approach can be used for altering the system/application components (that utilise system resources or access contextual information) as and when the values of those attributes change.

To make this last point clearer we shall consider the following example. A mobile system is equipped with a GPS device and also has access to a wireless cellular network. The system is capable of defining its current location via two different methods: either by polling the GPS device or by requesting a location server to define the actual cell in which it operates. Both solutions have different trade-offs as regards the operation of the system and its resources. In more detail, the first (GPS) solution allows a better level of accuracy but requires more power to operate. Alternatively, the use of a communications cell for positioning information may provides less accuracy but, in the case that the network link is already in use for some other operation, no extra power is required to specify the systems location. The requirement for a highly adaptive system is to be able to choose either of these approaches during runtime according to the user needs.

In this example the adaptive mechanism that we would require is to be able to alter the method of getting location information according to the present requirements of the system (e.g. less power consumption or more accuracy). Adopting a reflective approach in the development of the system would allow us to inspect the implementation of a location-aware application and modify the location mechanism according to the systems needs. More specifically, the applications provide a meta-space where the implementation is represented. By accessing the meta-space the system is able to inspect the behaviour of the application. This approach enables the components that retrieve location information to be identified. Furthermore, the meta-space protocol allows us to change the implementation of specific behaviours of the application. Thus the system can modify the implementation for acquiring location information during runtime. If the operational requirements of the system favour one of the mechanisms over the other (e.g. using the network rather than the GPS device) then the system can alter the implementation of all location components in the system in order to utilise the appropriate mechanism.

Alternatively, the previous example could be implemented without a reflective approach. In this case, a general location component for the whole system could be capable of switching between the two mechanisms transparently. However, highly adaptive systems depend on an arbitrary number of contextual triggers, and this general solution cannot be applied in a wide scale.

A second issue that arises from the previous example is the fact that highly adaptive mobile systems should be co-ordinated according to the dynamic change of user needs. As it was described in [3] the user should be allowed to specify system wide adaptation policies. These policies will provide the criteria for choosing the appropriate adaptation mechanism. Reflection and open implementation can provide the means to inspect the effects that every system component has on the system. Specifically the meta-object protocol can support inspection mechanisms for calculating the power consumption of the underlying objects, the network requirements, etc. This information can allow the adaptation mechanism to discover the way in which the system's implementation needs to be altered in order to achieve the criteria specified in the adaptation policies.

In the previous example the system would be able to inspect the requirements of all applications in terms of how accurate the location information should be. Furthermore, it would be possible to estimate the benefit in power consumption by switching off the GPS. Thus if the battery's power level drops the system could evaluate the current situation according to the adaptation policies defined by the user and decide whether it would be beneficial to adapt and switch from GPS to a network based location mechanism.

Summarising, in this paper we argue that adopting a reflective approach from system level to application level in mobile computing is a useful paradigm for the development of highly adaptive systems. Moreover, the adaptation mechanism can be based on the user requirements described through the definition of system wide adaptation policies.

References

1. Costa, F., G.S. Blair, G. Coulson, "Experiments with Reflective Middleware", Proceedings of the ECOOP'98 Workshop on Reflective Object-Oriented Programming and Systems, ECOOP'98 Workshop Reader, Springer-Verlag, 1998.
2. Davies N., Cheverst, K., Mitchell, K. and Friday, A. "Caches in the Air: Disseminating Information in the Guide System". Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99), 1999.
3. Efstratiou, C., Cheverst, K., Davies, N. and Friday, A. "Architectural Requirements for the Effective Support of Adaptive Mobile Applications". To be presented in Middleware2000, 2000.
4. Noble, B., Satyanarayanan, M., Narayanan, D., Tilton, J. E., Flinn, J. and Walker, K. "Agile Application-Aware Adaptation for Mobility", In Proceedings of the 16th ACM Symposium on Operating System Principles, 1997.
5. Yokote, Y., "The Apertos Reflective Operating System: The Concept and Its Implementation", In Proceedings of OOPSLA '92, vol. 28 of ACM SIGPLAN Notices, pp 414-434, ACM Press, 1992.