A Reflective Goal-Based System for Context-Aware Adaptation

Dejian Meng and Stefan Poslad

School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, United Kindom
{dejian.meng,stefan.poslad}@elec.qmul.ac.uk

Abstract. Many context-aware applications exploit current world contexts to control information access and service adaptation. To support a user centric utility model for a context-aware system, user goals are supported by the system. However, traditional goal-based approaches such as planning are hard to achieve in a context-aware environment because environments and systems are dynamic. This can lead to inconsistencies and goal conflicts. In this research, a new goal-context based approach is introduced to guide service adaptation for context-aware applications. A reflection model is investigated and applied to the system to resolve ambiguities and inconsistencies between goal-based planning and actual service adaptation. The new system, with its improved flexibility and robustness, is demonstrated in the form of a mobile spatial routing application.

Keywords: Context-aware; Reflection; Adaptation; Goal-based; Spatial routing.

1 Introduction

Mobile applications [1] [2] [3] [4] can exploit spatial routing services to plan all kinds of traffic routes for mobile users. Mobile applications can configure or customize these routing services according to user needs. When the environment where the application is situated changes, the application will sometimes need to reconfigure these services to match different context conditions. Context-awareness, in this case is deployed to enable the mobile application to adapt its service configuration. In a user-centric pervasive environment, such as travel, the environment should include not only the physical environment such as current location, traffic flow, etc., but also the user environment. The user environment model can include a user's situation, tasks, preferences and goals.

To introduce support for user goals into the context-aware system, several problems need to be addressed: how user goals can be defined so that a continuous and variable user expectation or preferences to the application can be recognized; how a goal-based approach and dynamic planning can be used to introduce more flexibility into existing static context-service relationships; how goal-based planning can handle user situation, physical environment and system changes; How any ambiguities and inconsistencies in the planning process that occur can be handled. In this paper, a

C. Hesselman and C. Giannelli (Eds.): Mobilware 2009 Workshops, LNICST 12, pp. 105–110, 2009.

reflective user goal-based context-aware framework is proposed to address these design issues.

2 A Reflective User Goal-Based Context-Aware Framework

In this context-aware framework, context models especially the goal context model are defined. Two layers of adaptation, i.e., context adaptation and service adaptation are defined. The context adaptation layer enables planning to be deployed during context evolution (from current contexts to goal contexts), which provides a way to exploit goal contexts to reflect user expectations about the interaction. A service adaptation layer performs the actual configuration or customization of services. A reflective middleware between the two layers is used to resolve any inconsistencies.

2.1 Contexts

Context-aware systems are systems that are aware of the situation in their physical and virtual environment and can respond in some way to benefit from knowledge of that situation. A context, according to Dey and Abowd [5], can be defined as any information that can be used to characterize the situation of an entity, considered relevant to the interaction between a user and an application. Spatial routing applications require information such as current location, time, weather, traffic situation and so on to characterize route environment changes, so that it can be aware of their changes and adapt to them.

Context adaptation can be represented as a transformation from current user contexts to users' goal contexts. In the spatial routing application, users expect routes to be generated to satisfy all their constraints and preferences. The simple view is that the destination location is the goal context in the spatial routing example, because the application user is heading to the destination. However, in terms of the spatial routing application, a more complex view of the adaptation should also take into account goal constraints of the route as goal contexts, i.e., a route to the destination that is short, fast or scenic. The goal context might be changing by the time, as user expectations in such a dynamic environment will be different.

Two dimensions of a goal context are modeled: the intrinsic goal (path) context defines how the goal context is reached from a corresponding current context; while the extrinsic goal (constraints) context defines meta-dimension information about the goal dimension information, e.g., how the goal context is constrained. The difference relies on whether it is information about the goal (user expected) situation, or the goal (user expected) information about the current situation. The goal dimension configuration specifies rules to change the configuration of goal contexts, e.g., the weights of each goal context. Some concrete context instances for a spatial routing application are shown in Fig. 1. For example, if it is known that other users will arrive late at a meeting venue, a fast route becomes less important because the user is not in a hurry as his or her preferences for the goal constraints have changed.

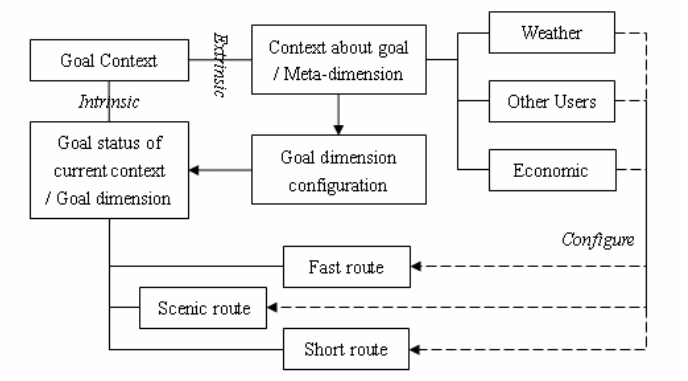

Fig. 1. The goal context model defined in terms of a goal path and goal constraints

2.2 Adaptation Layers

Service adaptation refers to the adaptation of service instances and to the dynamic customization of services. The adaptation of service instances can be multi-valued, involving reconfiguration of weights to determine how a goal path is generated according to the combination of one or multiple goal constraints e.g., for a spatial routing application [6], the weights of how fast, how scenic, and how short the route is can be adjusted. Applications execute the expectation of the user through instantiating service customization. External contexts drive service adaptation to change the current situation to the goal situation.

Most designs for context-aware computing use built-in mechanisms such as condition-action rules for service adaptation. There is a lack of a holistic approach to combine the use of multiple individual contexts to adapt to goal contexts, hindering more complex adaptation. This often involves deliberating about context changes rather than purely reacting to context changes. Static rules lack flexibility because the knowledge that supports decisions is not represented explicitly and cannot be modified. A layered adaptation model for a context-aware system is defined comprising a Context Adaptation Layer (CAL) and a Service Adaptation Layer (SAL). CAL is made up of a context model and a context path. In the CAL, current contexts and goal contexts need to be combined so as to establish an ordered (fully or partially) path to guide service adaptation in SAL. Context adaptation is a process which guides service adaptation. In this case, a context adaptation specification will be transformed into a service adaptation specification, which links (external) context adaptation to (internal system) service adaptation. Therefore an interface is defined between the external world (context) and internal world (service). A service adaptation layer actually executes the services which are driven by the external context adaptation layer (Fig. 2).

The multi-lateral context-aware adaptation architecture (Fig.2) is made up of a context adaptation layer, a service adaptation layer and middleware. Goal contexts are abstracted from the user expectation to the application. Planning is used to generate a context adaptation specification for a context path from the current to the goal

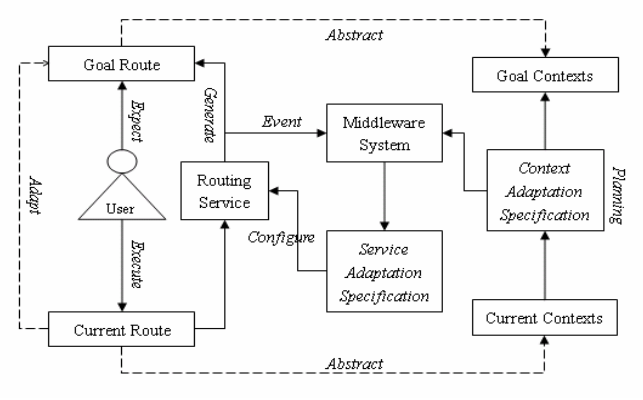

Fig. 2. A multi-lateral context-aware adaptation architecture

context. When a configured service does not generate a goal route that meets the goal constraints, the inconsistency is spotted and reported back to the middleware as an event to be resolved (see next section).

2.3 The Reflective Middleware

With loosely-coupled context and service models, inconsistencies between context changes and service changes can occur. For example, in spatial routing applications, if there is a traffic accident deviation to the planed route, CAL will normally plan to avoid a blocked road, minimizing the added deviation to satisfy a multi-valued goal context such as a particular shortest, fastest and scenic route. This leads to a routing failure in SAL because the traveling time from such a routing plan will exceed the time limits the user expected. To solve this problem, reflection about such potential failures takes place before the actual route is generated. To this end, Adaptation Middleware Layer (AML) also adds a reflective model about the internal system to the existing external environment model.

A reflective system focuses on meta-computation, computation to reason about its own operational computation [7]. A reflective system needs a representation of its own computation. The operational status and computation of the system complies with this representation [8]. The key to applying a reflective model to the existing system is that the self-representation can be modified at run-time and these modifications actually have an impact on the run-time computation. Regarding our system, we add AML to the existing spatial routing service which is located at the SAL to make the service adaptation reflective (Fig. 3).

Context composition combines different context types in a multi-valued context and instances of contexts. It uses a planner to link the current contexts and goal contexts. The planning process is achieved by partitioning the goal contexts further into sub-goal contexts and reasoning about the corresponding knowledge defined in the context domain to meet those sub-goal contexts. A context adaptation specification will be generated from the context composition, which will be mapped

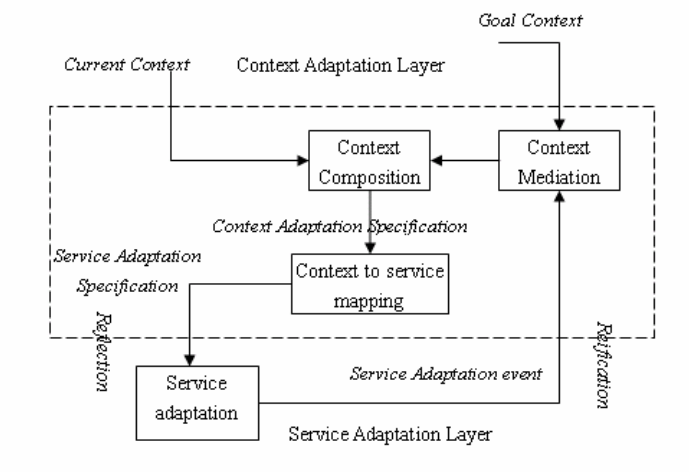

Fig. 3. The reflective middleware layer model in more detail

to the service adaptation specification, the representation of the actual service adaptation. Service adaptation is treated as the base system to be reflected on. The service adaptation specification defines policies and parameters for service adaptation.

While a service is being adapted, internal events may occur because of unexpected inconsistencies arising from external context changes. These events will be reported to the context mediator in the AML, which mediates between the context changes and the context composition leading to more realistic goal contexts. Mediation can occur in two ways: the goal contexts can be mediated automatically if they are represented semantically, e.g., short-time travel can sometimes be viewed as short-distance travel if the travel speed is not important. If the mediation cannot be done automatically, user intervention is needed to revise the goal contexts. Any changes taking place in the context composition will lead to a change in the service adaptation specification, which therefore reflects the change back to the service itself.

3 Application Trial and Discussion

A map demonstrator [9] has been implemented by using Geotools [10] where our system has been deployed to acquire some results from test cases shown in figure 4. When re-route is required for a traffic block at the point of Queen Mary, the case on the left shows the pre-planed route R1, and the other one shows the adapted route R2. R2 takes a longer distance and less scenic part of area, but will be much quicker to get to the destination in time. In R2 planning, the user goal contexts have been reconfigured, because the routing service first tried to get a route across more park areas which could maximize the aggregated benefits by the old weighting, but turned out to exceed the time limit; therefore the reflection is called to amend the weighting.

Fig. 4. A pre-planed static routing (left) and a reflective goal-based adaptive routing (right)

4 Conclusion

We investigated and applied a reflective goal-based system for context-aware adaptation in a spatial routing application. The use of the goal context model and the multilateral adaptation framework can resolve the design issues as proposed. The system enables dynamic goal context configuration and hence service adaptation. In the future, we need to develop more use cases and to consider the overhead in deploying the system.

References

1. Fawcett, J., Robinson, P.: Adaptive routing for road traffic. IEEE Comput. Graph. Appl. 20(3), 46–53 (2000)
2. Rogers, S., Fiechter, C.N., Langley, P.: An adaptive interactive agent for route advice. In: The third annual conference on Autonomous Agents, Seattle, Washington, United States (1999)
3. Hochmair, H.: Towards a Classification of Route Selection Criteria for Route Planning Tools. In: Developments in Spatial Data Handling 11th International Symposium on Spatial Data Handling (2005)
4. Lee, J., Park, S., et al.: ACE-INPUTS: A Cost-Effective Intelligent Public Transportation System. IEICE Trans. Inf. & Syst. E90–D(8), 1251–1261 (2007)
5. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. In: Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness. ACM Press, New York (2000)
6. Meng, D., Poslad, S.: A reflective context-aware system for spatial routing applications. In: MPAC 2008, pp. 54–59 (2008)
7. Poslad, S.: Ubiquitous Computing: Smart Devices, Environments and Interactions. Wiley, Chichester (2009)
8. Maes, P.: Concepts and Experiments in computational Reflection. In: OOPSLA 1987 Proceedings, pp. 147–155 (1987)
9. Liang, Z., Poslad, S., Meng, D.: Adaptive Sharable Personalized Spatial-Aware Map Services for Mobile Users. In: The GI-Days 2008 Conference, Munster, German, pp. 267–273 (2008)
10. Geotools The open source Java GIS toolkit, `http://geotools.codehaus.org/`