Chapter 12

Efficient Adaptation Decision Making Algorithms for Context-Aware Applications

Yves Vanrompay
Katholieke Universiteit Leuven, Belgium

Tim Smits
Consultant at AE, Belgium

Yolande Berbers
Katholieke Universiteit Leuven, Belgium

ABSTRACT

The current trend toward ubiquitous and autonomic computing leads to the necessity to realize adaptive applications. The environment of applications becomes more heterogeneous and user QoS needs and resource availability vary heavily. To maintain the usability, availability and reliability of these applications, adaptive behavior is needed. In this paper, the authors examine the decision-making phase in realizing adaptive behavior of an application. Algorithms reason about the adaptive application and the current context with the goal of selecting a new application variant that better fits the environment. This reasoning happens in a search space that is very complex and consequently poses a problem toward scalability; even when applications are relatively small. Therefore, the authors present two novel algorithms that handle the complex search space in an intelligent way. In this paper, both algorithms are introduced along with a thorough evaluation of their behavior and scalability.

1. INTRODUCTION

Adaptive applications are an integral part of the vision within computer science towards ubiquitous (Weiser, 1991) and autonomic (Kephart & Chess, 2003) computing. In the ubiquitous computing paradigm the user is surrounded by many computing devices in a very dynamic environment. For example, available network bandwidth can vary, systems can go offline and come online unexpectedly, user needs change according to their location, role and time of day. In order to maintain the usability, availability and reliability of the application in every context, adaptive...
behavior is required. In order to realize adaptive behavior three phases must be executed, as will be discussed in section 2. Our research focuses on the second of these phases: the decision making. The responsibility of this phase is to reason about the adaptive application with the goal of selecting a new application variant that better fits the new environment. The challenge in developing the algorithms that perform this reasoning is the complexity of the search space. Even small adaptive applications will have many possible application variants that can be selected. Also, it would be unacceptable to keep the user waiting for too long until an adaptation is performed. Furthermore, adaptive applications are mostly executed on mobile devices with limited computing resources. Because of these reasons, an exhaustive evaluation of the fitness of all application variants in order to find the optimal one is not feasible. The scalability of decision making algorithms is thus the most crucial property. We present two novel algorithms, graceful degradation and graceful upgrade, and show that their performance and scalability are significantly better compared to existing algorithms. This paper is organized as follows. Section 2 elaborates on the mechanisms that can be used to adapt applications. The two novel algorithms are explained in section 3, while section 4 presents an extensive evaluation of the algorithms. In section 5 an overview of related work is given and we draw conclusions in section 6.

2. ADAPTATION

The realization of adaptive behavior can be divided into three phases. In the first phase, information about the environment of the system, also named context, is being sensed and interpreted. Adaptive applications use this information to evaluate their fitness with respect to the current context with the goal to optimize the provided Quality of Service for the users. When, at a certain moment, this fitness becomes too low, an adaptation may be performed. Context is discussed in the first subsection. The second phase is called the decision making and is responsible for the reasoning about the adaptive application after a (significant) change in context took place. Our research is concentrated on the decision making phase which is discussed in subsection 2.2. Subsection 2.3 elaborates on the third phase where the running application is adapted to the changed context.

2.1. Context

Context information is information that provides relevant knowledge about the environment of a system. Well know examples are the available memory resources or battery level of a mobile device, the user’s name, age and location or the light level in a room. Many definitions of context are given in related work, but the most frequently used is given by Dey (2001):

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

It follows that context information is very open for interpretation and is relative to the application needs. Chen and Kotz (2000) presented a classification of context information into three types: computing, environmental, and user context. Computing context describes the information about the infrastructure available to the system. Available memory or processing resources, type of network and battery level are common examples. Environmental context covers all the information about the physical environment of the system and its users. The location of a user and the noise or light level in a room are common examples. Finally user context describes the information about the users and their actions. The profile of a user, his mood and agenda are common examples. The