A Study on a Methodology of Constructing Dynamically Adaptable Software Systems

Noriki Amano
Graduate School of Information Science,
Japan Advanced Institute of Science and Technology

February, 1999

Abstract

Nowadays open-ended distributed systems and mobile computing systems have come into wide use. In such systems, we cannot obtain accurate information of dynamically changing runtime environments beforehand. The changes of runtime environments have given a strong influence on the execution of programs, which we cannot ignore. Thus, the software systems that can adapt themselves to dynamically changing runtime environments are required. We call such software systems *dynamically adaptable software systems*. Since it is difficult to realize such dynamically adaptable software systems, the need for useful methodology to construct them increases.

In this work, we propose a software model called *DAS* for dynamically adaptable software systems, and design & implement the basic mechanism of the language based on the DAS model. The software systems based on the DAS model can effectively adapt themselves to dynamically changing runtime environments. Its language effectively support to describe such dynamically adaptable software systems. We present the effectiveness of the DAS model and its language. Moreover, we also present the effectiveness of such software systems based on the DAS model.

To realize dynamically adaptable software systems, it is an effective way to change the behaviors of each software system depending on the states of runtime environments. However, it is not practical to develop several versions of the same software system depending on each runtime environment and/or its states. Moreover, such behaviors of each software system depending on the states are related with any other parts of it. Thus, it is difficult to control such behaviors of each software system from its outside. From such reasons, each software system should have the ability that can adapt itself to dynamically changing runtime environments. We call such ability of software systems *dynamic adaptability*. The dynamically adaptable software systems (namely, software systems with dynamic adaptability) not only adapt themselves to dynamically changing runtime environments, but also change their own functionalities flexibly to make full use of the properties in the runtime environments.

However, there is a limitation on runtime environments and their states that software engineers can anticipate beforehand. Thus, there is also a limitation on the dynamic adaptability that the software engineers can give to software systems beforehand. From

the reason, the mechanism of dynamic adaptability must be extensible. Namely, the mechanism must be able to change depending on various runtime environments and their states afterward.

We propose a software model with dynamic adaptability in extensible way called DAS. The following are main concepts of the DAS model.

- The basic components of adaptable behaviors in a software system are procedures, and the control of adaptable behaviors is based on procedure's invocation.
- The DAS model has a variant of generic procedures (functions) based on the states of runtime environments called *adaptable procedures*.
- The control mechanism of adaptable procedures is realized by using *reflection*.
- A software system based on the DAS model forms a *meta-level architecture*.

These concepts are based on our following objectives.

- High compatibility for existing software systems.
- High flexibility of dynamic adaptability.
- High readability, maintainability and extensibility.

We have a great many requirements that we want to use many existing software systems effectively in open-ended distributed systems and mobile computing systems. Therefore, we require the methodology that we can introduce dynamic adaptability into software systems in existing languages. In the DAS model, the basic components of adaptable behaviors in a software system are procedures. Then, the adaptable behaviors are realized as generic procedures (functions) based on the states of runtime environments called *adaptable procedures*. An adaptable procedure consists of several *adaptable methods* that have runtime codes depending on the states. When an adaptable procedure is invoked, a well-suited adaptable method is selected depending on the states, and its runtime codes are executed. This approach can be applied to many existing languages that have the procedure invocation's mechanism in principle. Consequently, we have a great potentiality for effective utilization of existing software systems. In this work, we had implemented the basic mechanism of the DAS model as the class libraries in Java (DAS class libraries). Then, we actually introduced dynamic adaptability into applications in Java. Through the implementations of them, we presented that the DAS model has high compatibility for existing software systems.

In our approach, to realize dynamic adaptability of software systems in flexible way, the control mechanism of adaptable procedures must be able to change depending on the states of runtime environments. In the DAS model, the mechanism itself is realized using an adaptable procedure in *reflective* way. Therefore, it can adapt itself to the states (namely, it can change its own behaviors depending on the states). By changing the mechanism depending on the states, we can realize adaptable behaviors of software systems in flexible way such as the temporary restrictions & releases of their functionalities without modifying their original programs. Moreover, since the mechanism is realized by using an adaptable procedure, its customization is also realized by adding adaptable methods incrementally without modifying its structure. We confirmed such advantages

through the implementations of dynamically adaptable software systems using DAS class libraries. Then, we presented the effectiveness of such adaptable behaviors in flexible way from the examples.

A dynamically adaptable software system based on the DAS model forms a *meta-level architecture*. The descriptions of the mechanism that controls adaptable behaviors in the software system (meta-level) are completely separated from the descriptions of the primary subject domain in it (base-level). The descriptions of the control mechanism are hidden into the meta-level. This separation of concerns improves the readability, maintainability and extensibility of dynamically adaptable software systems. We actually confirmed such advantages through the implementations of dynamically adaptable software systems using DAS class libraries.

Key Words: **Software Adaptation, Reflection, Adaptable Procedures, Open-Ended Distributed Systems, Mobile Computing Systems**