Title	ディスコネクティッドオペレーションをサポートする適応可能なアプリケーションの構築
Author(s)	嶋本，堅司
Citation	
Issue Date	1998-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1162
Rights	
Description	Supervisor:中島 達夫，情報科学研究科，修士

Building Adaptive Applications to Support Disconnected Operation

Kenji Shimamoto

School of Information Science,
Japan Advanced Institute of Science and Technology

February 13, 1998

Keywords: disconnected operation, adaptive applications, tradeoff, object-oriented, consistency.

Recently we can execute various kinds of applications on mobile computers, because of technological advancement, such as smaller electronic devices, higher CPU performance and more effective power supply. We often connect to conventional distributed computing environments, LANs, etc., from outside with our mobile computers, using communication facilities such as cellular phone, PHS(Personal Handyphone System), and so on.

However these communication media have many constraints to execute applications in mobile computing environments, compared to conventional computing environments, for example, lower bandwidth, lower speed, higher cost, and more intermittent communication. We have no appropriate communication media in mobile computing which satisfies all constraints, even if we would make use of some communication facilities anywhere. It is impossible to execute applications in mobile computing, unlike in conventional one.

Recent researches on disconnected operation deal with their constraints on communication. Users in offices or schools execute applications on conventional networks, such as LANs, with their mobile computers. While they are out of their buildings, they usually access to the networks through the constrained communication media, such as public phone. Just before going out, or before disconnecting from the conventional networks, users copy the applications and files on servers to mobile computers as duplicated ones. They can make use of the duplicated applications or files, though they cannot connect to their original servers, or when they do not have to. When they can connect, or when they want to access the updated shared data, they can access to the original servers. As a result, users avoid connecting if not necessary. They execute applications with higher performance and lower cost as a whole, and it is possible to send data efficiently on lower bandwidth, higher cost, and intermittent communication media.

The former researches focus on only one requirement that users and applications demand, such as performance, cost, consistency of shared data, and so on. They try to realize the one requirement efficiently. There are huge gaps between the researches and real world, since real users and applications require that all metrics are satisfied ideally. However, it is actually impossible to satisfy every metric. Therefore it is necessary to consider tradeoff between such metrics to reduce the gaps.

Suppose a user to consider consistency of shared data as most important, and immediate connection as necessary, she intends to connect to the original server even if the communication cost is expensive. On the other hand, suppose that she cannot expense higher cost, and do not have to access the updated data, they can make use of the duplicated server on her mobile computer. If users consider tradeoff, and if it is reflected on implementation, we think, they execute applications adaptively according to their requirements.

We have to take into account tradeoff in the future. It is true that communication cost become lower and lower. Also it is true that the cost become relatively higher, when we make use of new advanced communication media produced by remarkable technological innovation. In this situation users are always required to consider tradeoff between various metrics to execute applications comfortably.

In this thesis, we present a interface taking into account tradeoff which users consider and describe, and also we present a framework and a mechanism which maps the interface to the implementation. Users specify metric kinds and numeric parameters which mean the importance of the metrics. Mapping layer, which maps the interface to an implementation, calculates the parameters to decide which implementation should be selected. We also prepare a interface between the mapping layer and implementation, so mapping layer can specify the kind of implementation, which the implementation layer supports, through the interface. Both interfaces are well-defined, so it is easy to change the inside of the mapping layer as long as it obeys the interface specification. Using the interface taking into account tradeoff, application programmers can build adaptive applications which consider users or applications requirements.

We carry out the following steps to prove the effectiveness of the interface. First of all, we design and implement a toolkit which supports disconnected operation for client-server-based object-oriented applications. The toolkit has three kinds of implementation which are adapted to the tradeoff between communication cost and consistency of shared data. Second, we develop a group scheduler with C++ as an example application, which manages the schedules of some staff members and a group. Finally, we simulate the actual utilities of the group scheduler, and evaluate the effectiveness under the consideration of the tradeoff between communication cost and consistency of shared data.

In addition, we try to realize maintaining the consistency based on users requirements in developing the toolkit, by introducing some kinds of conflict resolution strategies. This kinds of conflict resolution on optimistic concurrency control have not been discussed so far. We also propose some problems which have not been considered, such as semantics dependency of applications and guarantee of serializability. If we realize a transparent toolkit for object-oriented applications, we have to solve such problems.