Position Summary: Supporting Disconnected Operation in DOORS

Nuno Preguiça, J. Legatheaux Martins, Henrique Domingos, Sérgio Duarte
Departamento de Informática
Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa
Quinta da Torre, 2825-114 Monte da Caparica, Portugal
{nmp, jalm, hj, smd}@di.fct.unl.pt

1. Summary

The increasing popularity of portable computers opens the possibility of collaboration among multiple distributed and disconnected users. In such environments, collaboration is often achieved through the concurrent modification of shared data. DOORS is a distributed object store to support asynchronous collaboration in distributed systems that may contain disconnected computers. In this summary we focus on the mechanisms to support disconnected operation.

The DOORS architecture is composed by servers that replicate objects using an epidemic propagation model. Clients cache key objects to support disconnected operation. Users run applications to read and modify the shared data (independently from other users) – a read any/write any model of data access is used. Modifications are propagated from clients to servers and among servers as sequences of operations – the system is log-based.

Objects are structured according to an object framework that decomposes object operation in several components (figure 1). Each component manages a different aspect of object execution. Each object represents a data-type (e.g. a structured document) and it is composed by a set of sub-objects. Each sub-object represents a subpart of the data-type (e.g. sections).

A new object is created composing the set of sub-objects that store the type-specific data with the adequate implementations of the other components.

The following main characteristics are the base to support disconnected operation in DOORS.

Multiple concurrency control/reconciliation strategies: To support the different requirements posed by multiple data-types we rely on the flexibility provided by the DOORS object framework. The concurrency control component allows the use of different log-based reconciliation strategies. The capsule component allows the definition of different data configurations – e.g. the tentative and committed versions of an object can be easily maintained duplicating the adequate components under the control of the capsule.

Integrated awareness support: The reconciliation among concurrent streams of activity is often performed when users are no longer connected to the system. In DOORS, awareness information may be generated and processed during the reconciliation phase – this approach makes it possible, for example, to provide shared feedback about data evolution and/or to explore off-system communication infrastructures, such as the use of SMS messages.

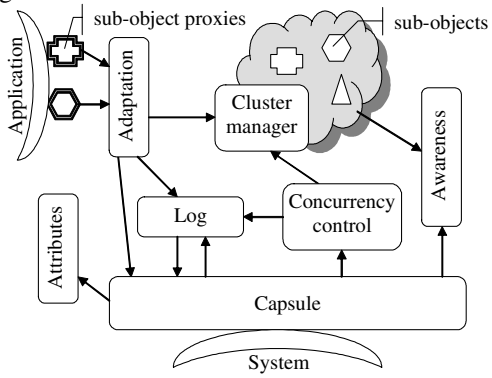

Figure 1. DOORS object framework.

Partial caching: As caching a full data object is sometimes impossible for resource-poor mobile devices, DOORS allows sub-objects to be cached independently.

Blind operation invocation: When some sub-objects are not present in the cache, a disconnected user is still allowed to execute operations on them. A replacement sub-object may be created to present the tentative result of these operations.

Adaptation: The adaptation component allows the use of different strategies to adapt to variable network conditions – as a result, operations may be performed immediately on a server or on the local copy.

The interested reader may find more information on the DOORS system in [1] (including an extended version of this summary). This work was partially supported by FCT, project number 33924/99.

2. References

[1] http://dagora.di.fct.unl.pt

Proceedings of the Eighth Workshop on Hot Topics in Operating Systems (HotOS-VIII '01)
1530-1621/01 $10.00 © 2001 **IEEE**