Stable Storage

by crashing during the
rupted, if the system fails
rited to

Data stored in volatile
such as main
will not sur-
vive program termina-

Data stored to non-va-
storage may get cor-
rupted, if the system fails
in some way, for instance
by crashing during the

The parallel hierarchy of storage parameters allows each stor-
ge to define its own storage parameter class containing all the
information needed to uniquely identify data stored on
the device. At the same time, it allows an application pro-
grammer to create instances of storage classes by means of the
factory pattern. The storage parameter class also implements the Serializable
interface. Objects must implement the Serializable interface with its operations
Read_From (Storage) and Write_To (Storage). Their imple-
mation will call the Read_From and Write_To operations of the concrete storage
to store the state of the object. Using this technique, the object itself has no
knowledge about the external representation format. In some object-ori-
ted programming languages such a serialization mechanism is already
provided (Ada streams, Java serialization package).

When storing the state of an object on some storage, the data must first be
transformed from its representation in memory to some form that can be
stored by the device. This is achieved by applying the Strategy design pat-
ttern. Objects must implement the Serializable interface with its operations
Read_From (Storage) and Write_To (Storage). Their imple-
mation will call the Read_From and Write_To operations of the concrete storage
to store the state of the object. Using this technique, the object itself has no
knowledge about the external representation format. In some object-ori-
ted programming languages such a serialization mechanism is already
provided (Ada streams, Java serialization package).

To prevent storage leaks in the presence of failures the storage parameters of any un-
modified persistent object can be stored in a stable storage directory. The application
programmer will choose the kind of stable storage used to store the directory when
initializing the persistence support. The operation will be performed: the object is
written to the stable storage, and an additional housekeeping operation is
performed to notify the file system that an object has been
written to persistent storage.

Overview

This poster presents a framework providing persistence support for object-oriented programming languages without modifying the run-time system or the language itself. It does only rely on basic object-oriented pro-
tecting techniques, and can therefore be implemented in any object-oriented programming language. It is
based on design patterns. Its strengths are:

• Clear Separation of Concerns: Persistent objects do not know about storage devices or external data formats.
• Modularity and Extensibility: It is straightforward to define new persistent objects or add support for new
storage devices.
• Safe Storage Management: Storage leaks are prevented.

Implementation

This framework has been implemented using the object-oriented programming language Ada 95. Ada's elaborate
design patterns allow to decouple broadcasting, group and topic management from the actual storage used to store the data.

Additional Information

Additional information and complete papers describing the framework and the implementation can be obtained at
the following URL:

http://www.epfl.ch/departement/ing/paradigm.html