Rollback Recovery

Rollback recovery is a recovery class that ensures that failures do not cause information loss. More specifically, it guarantees that the effects of all input elements are always preserved until Odysseus can continue processing the elements (at-least-once). Therefore, rollback recovery techniques result in a complete output stream (relative to its failure-free counterpart), but typically they result also in duplicate values (all values between the last checkpoint and the system failure). (Reference: J.H. Hwang et al., High-Availability Algorithms for Distributed Stream Processing, 2005)

In Odysseus, the rollback recovery uses an external application, Backup of Data Streams (BaDaSt), that preserves all incoming elements after the last checkpoint. After a crash of Odysseus and a restart, Odysseus receives all preserved elements after the last checkpoint from BaDaSt and processes them before it processes the newer elements (which are again preserved by BaDaSt, so there is no data loss). Note that Odysseus will try to use the original data source again after some time, but for data streams with a high data rate and a long time span between crash and restart, it is possible that Odysseus can not switch back to the original source. In that case, Odysseus continues to process the elements from BaDaSt.

Requirements: To use the rollback recovery, you have to (1) define a BaDaSt recorder for each data source (in the same file as the source definition) that elements shall be preserved (see BaDaSt), (2) choose the rollback recovery and (3) manage the checkpoints (see Checkpointing):

Example

```
#PARSER PQL
[Definition of data source "bid"]
#BADASTRECORDER type=TCPStringRecorder sourceame=bid

#RECOVERYCONFIGURATION RollbackRecovery protectionpointunit=MINUTES protectionpointperiod=1
[Definition of queries to apply rollback recovery]
```