

XVSM Tutorial and Application Scenarios

Masterstudium
Software Engineering and
Internet Computing

Michael Wittmann

Technische Universität Wien
Institut für Computersprachen
Programmiersprachen und Übersetzerbau
Betreuerin: A.o. Univ. Prof. Dr. Dipl.-Ing. eva Kühn

XVSM (eXtensible Virtual Shared Memory) is middleware to let applications write data to a shared memory that can be read by other applications. The data is distributed over a set of peers, which can read, write, take and delete it, independent on which peer the data is really stored.

This master thesis should provide the necessary information to learn and understand the features and possibilities of Space-Based Computing by means of XVSM. It was shown to be usable for this task by handing it to about 80 students and incorporate their feedback to the work. The application scenario document should give application designers an insight into the possibilities they have when using XVSM.

Tutorial

Target:

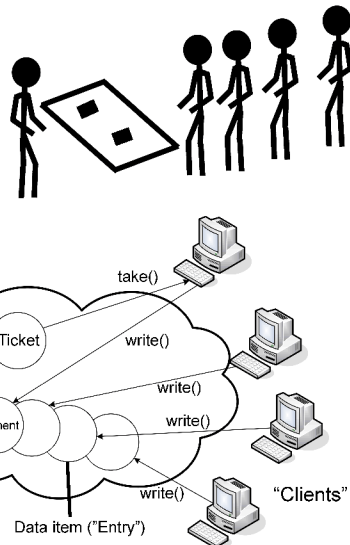
Let the reader understand the concept of Space-Based Computing by means of XVSM. Moreover, it is the target to show how to program using MozartSpaces, which is the Java implementation of XVSM.

The given examples and exercises are based on each other. Thus, the reader should be motivated to do the next exercise and to finish the complete tutorial.

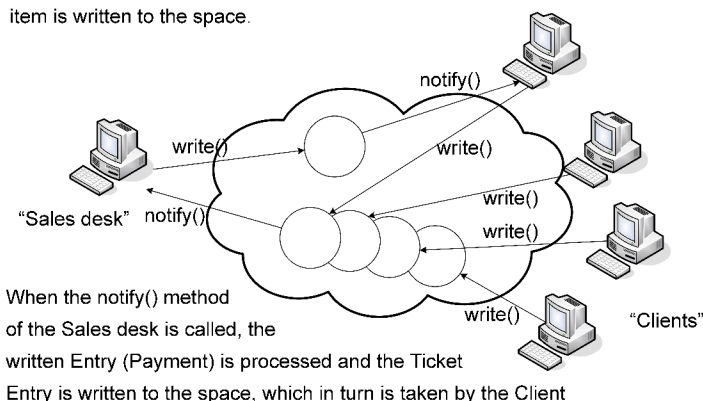
One of the exercises is as follows:

TicketQueue

A queue of people is standing in front of the Sales desk, waiting to buy a ticket. They are served one after the other. The aim of this exercise is to show the concept of the FIFO (first-in-first-out) coordinator



A subsequent exercise that is built on the TicketQueue exercise mentioned above is an enhanced version of the TicketQueue. While the previous example performs blocking read, this one tries to show the usage of Notifications. In this exercise, a Notification method is called whenever a data item is written to the space.



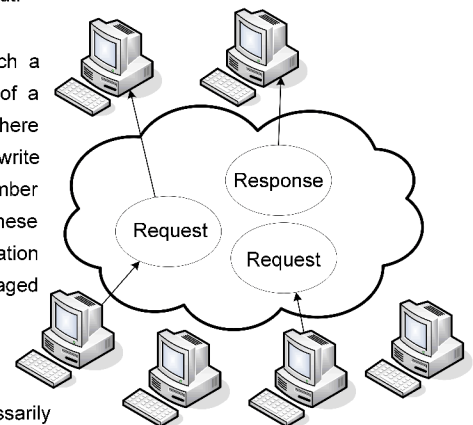
Application Scenarios

Target:

The target group for this part is the software designer that would like to get an idea of the application possibilities of the Space-Based approach.

There often is the problem that a big number of clients is requesting a single or a small number of servers, which in turn is/are overloaded and cannot reply to the request within a timeout.

A possibility to prevent such a behaviour is the usage of a common data space, where the requesting Clients write their requests and a number of servers can process these requests. The communication and coordination is managed transparently by XVSM.



The Request and the Response are not necessarily synchronous. The Request and the Response can be picked up at a later point in time, even if the requesting / responding party is no longer existing in the common data space.

An additional possibility to use XVSM is to implement "Staged Event Driven Architecture" (SEDA). It consists of various stages and an item (request) that is written to the stage. This item is processed and the result is written to the next stage. It is easy to add more computation power to a stage while it is in use, as multiple computers can work concurrently on this dataset.

In XVSM, these stages can be designed by using the data space ("Container"), while the processed items are the XVSM Entries. Using this approach, one can implement systems that are automatically balancing their load, process high-priority requests first, and much more.

