

# Workshop for advanced users

Payment and settlement simulator  
seminar - 2005

Matti Hellqvist, Bank of Finland



# Topics

- New and advanced features
- Performance optimization
- Database tools and tricks
- Process flow and logic of the simulator
- User modules



# Some existing possibilities

- Assessing scenarios in network of interlinked systems.
  - Simulations with several interlinked systems with different logic are possible. See e.g. examples 2&3
- Borrowing liquidity from specified account when necessary
  - Mimic behavioral response to liquidity shock (?)
  - implemented with "Liquidity injections"
  - Cost or friction terms are lacking however.



# More existing possibilities

- Simulation batches
  - Combine multiple (timely non overlapping) simulations into one.
  - Define batch from a set of simulation ID:s.  
=> Changes in underlying data or system specifications create efficiently new scenarios.
- Arbitrary queue order
  - LIFO, Smallests first, alphabetical...
  - All these in strict queue order or bypass-mode
  - Use QUUSEDEF algorithms and import the sorting criteria in user defined fields.



# New possibilities in 2.0.0

- Time / Date transpositions
  - Automatic transpositions in import or export of data.
  - Practical for systems that are open over midnight.
- Database optimization
  - ...or "Defragmentation" button for input and output database.
  - Reported to prevent slowing down of simulations in projects with massive data sets and repeated simulation runs.



# New possibilities in 2.0.0

- Intraday liquidity management features
  - Known limits for multilateral or bilateral flow of liquidity or granted credit can be replicated
  - Bilateral limits in v.2.0.0
- Bilateral statistics
  - Statistics on intraday flow on bilateral level: (Set large enough values for bilaterla limits)
  - Can be utilized e.g. to reveal bilateral limits used by the participants by observing history of actual bilateral balances



# Coming soon...

- Group codes of DVP data i.e. linking of arbitrary many transactions together
- Receipt reactive gross settlement
  - See Johnson – McAndrews – Soramäki (2004): 'Economizing on Liquidity with Deferred Settlement Mechanisms'
- PNS-algorithm for DVP data



# Topics

- Recent new features
- Performance optimization
- Database tools and tricks
- Process flow and logic of the simulator
- User modules





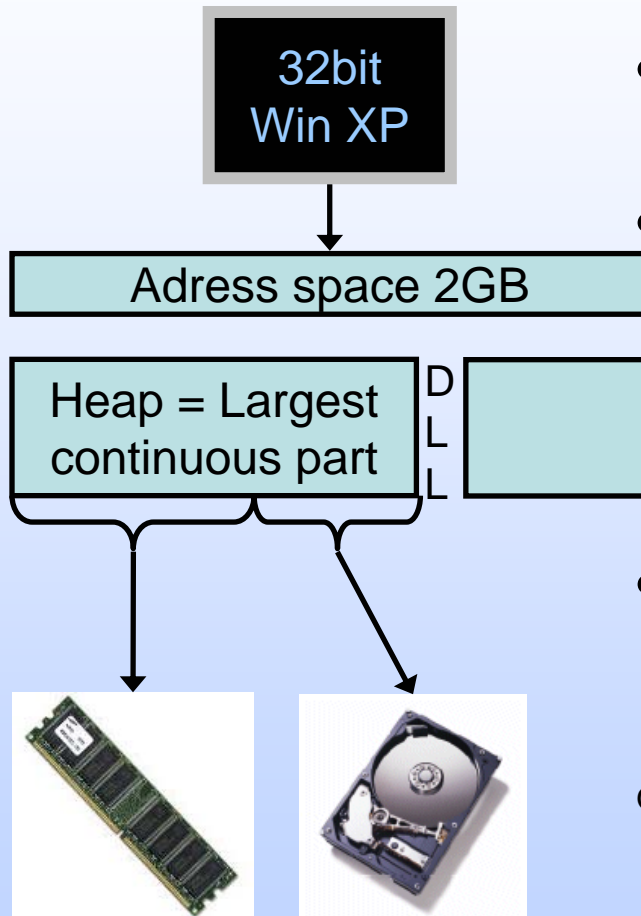
# When "tuning" is needed?

- If you make hardware modifications after simulator installation
- If the simulations run out of memory due to
  - Increased number of transactions / day
  - Decreased liquidity and accumulating queues
  - Increased nr of liquidity constraints
    - Bilateral limits

Complex processing rules will allways take their time (DVP, continuous gridlock resolution etc.)



# OS constraints



- 32bit Windows => max 2G memory per application ("address space")
- Java requires continuous block of address space for "heap" = available amount of memory for e.g. the simulator.
  - Max ~1.5G
- Available "virtual memory" used: RAM (+ Hard disk = "paging")

One detailed explanation:

<http://forum.java.sun.com/thread.jspa?threadID=584329&messageID=3009798>

Available resources



# Start up parameters

- Simulator:

c:\BoF-PSS\startup.bat

```
"jre-1.3\bin\java -Xms128m -Xmx512m ..."
```

- Xms = initial heap size
- Xmx = Maximum heap size

- MySQL: c:\my.cnf

– For alternative configurations see:  
c:\bof-pss\program\ or c:\mysql\

Simulations with paging will be **severy** slower regardless which program is out of memory (MySQL or BoF-PSS)



# Topics

- Recent new features
- Performance optimization
- Database tools and tricks
- Process flow and logic of the simulator
- User modules



# Managing projects

- Creating a project creates the structure of folders and databases.
  - After this, contents can be changed simply by copying files
- ⇒ Easy backups, cloning, transferring etc. of entire contents of a project.  
(Handy also for reporting bugs)



# Database tools and tricks

- Database of the simulator can be accessed directly for
  - Modifying the installation (templates, projects, user modules...)
  - viewing (or altering) the data directly
  - More powerful or tailored exports / imports of data



# Database tools and tricks

- In practice
  1. Start the database server:  
c:\bof-pss2\program\**database.bat**
  2. Open viewing or editing tool. Command line and graphical versions available
    - Simulator must not be running (DB locking)
- **Carefully** with the direct modifications...
  - e.g. removing the project defined to be default will paralyse the simulator



# MySQL tools

- Several easy to use monitor applications available
  - MySQL Query browser (freeware by MySQL)  
<http://www.mysql.com/products/tools/query-browser/>  
(connect to "localhost" as "root")
  - MySQL Front, MyCon, ... (Shareware)
- ODBC drivers for MySQL
  - Allows connection with e.g. MS Access  
<http://www.mysql.com/products/connector/odbc/>





# Topics

- Recent new features
- Performance optimization
- Database tools and tricks
- Process flow and logic of the simulator
- User modules



# The actual building blocks

## Main algorithms

**Submission:** What happens next?

**Entry:** Initial processing for transactions: settle immediately if possible, call sub algorithms if defined or send to queue.

**Settlement:** Execute sub algorithms to settle trans from queue.

**End:** Perform final procedures of day or settlement period.

} Common for all systems simulated concurrently

## Sub algorithms

**Queue:** Settle individual transactions from queue in defined order.

**Partial Net Settlement (PNS):** Settle a subset of queued trans

**Multilateral Net Settl. (MNS):** Settle queues with "All or nothing"

**Bilateral offsetting:** Match entered payments with queued payments having opposite direction (sender & receiver)

**Splitting:** Split larger trans into sub-transactions

**Injections:** Perform liquidity transfers between defined accounts

} Logics of one individual simulated system.



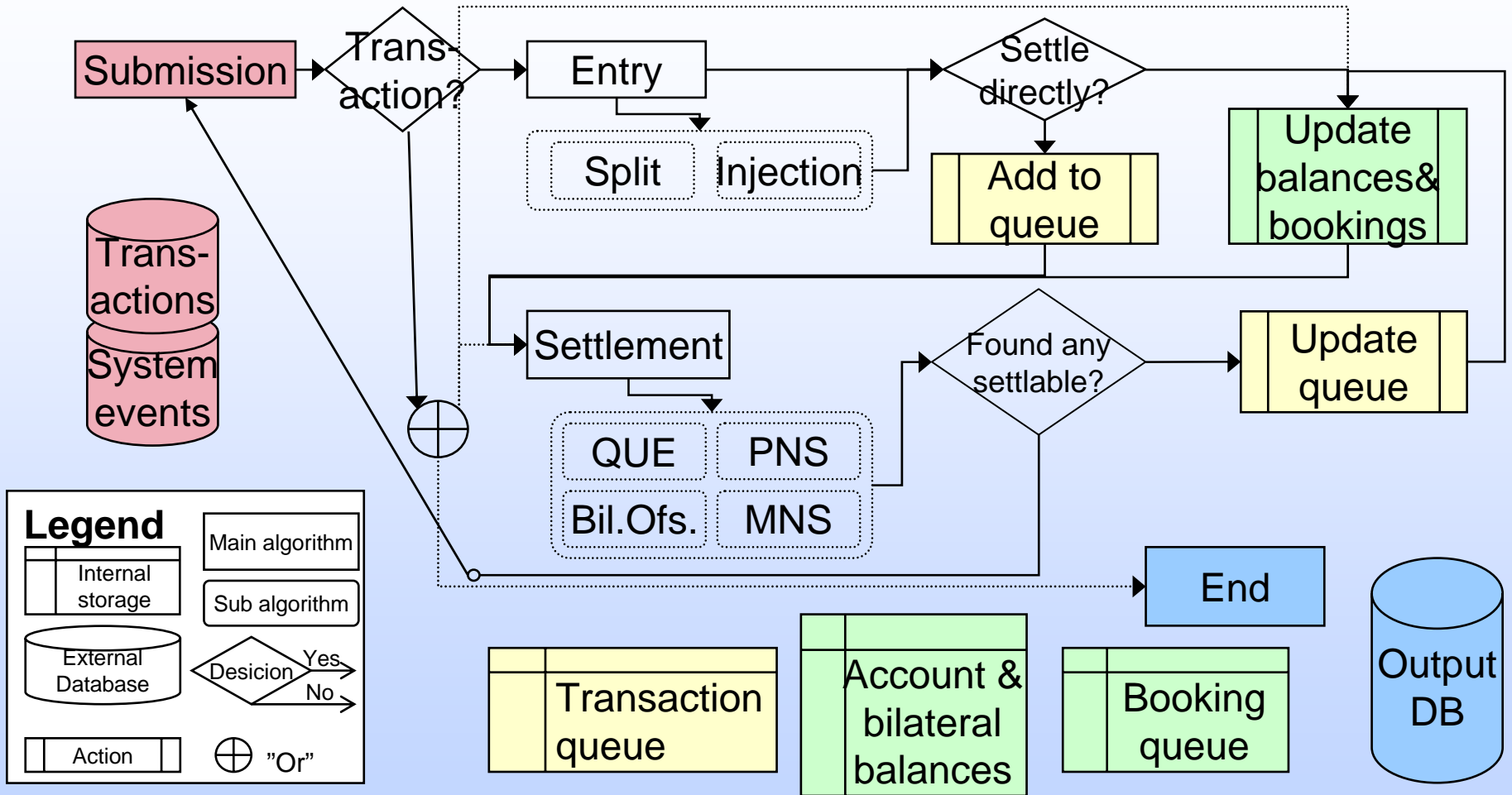
# Some definitions

- "Settling": Booking or execution of a transaction. Account balances are updated.
- "Netting" : Simultaneous settlement of independent transactions. Results in allowed balances for all involved accounts after all the transactions in the "netting" are settled.



# RTGS process

**Simulator engine:** User interface, process control, all common data



Subalgorithms are executed in the same order which they have in system definition <sup>20</sup>



# Topics

- Recent new features
- Performance optimization
- MySQL tools and tricks
- Process flow and logic of the simulator
- User modules



# User modules

- When existing logics/algorithms are too limited you can build new ones of your own
  - Setting up development environment
  - Case study: Group Codes
  - JavaDoc
  - Behavioural algorithms



# Development environment

- Java coding is needed
  - Some tools to recommend: Jbuilder, eclipse ([www.eclipse.org](http://www.eclipse.org)), NetBeans...



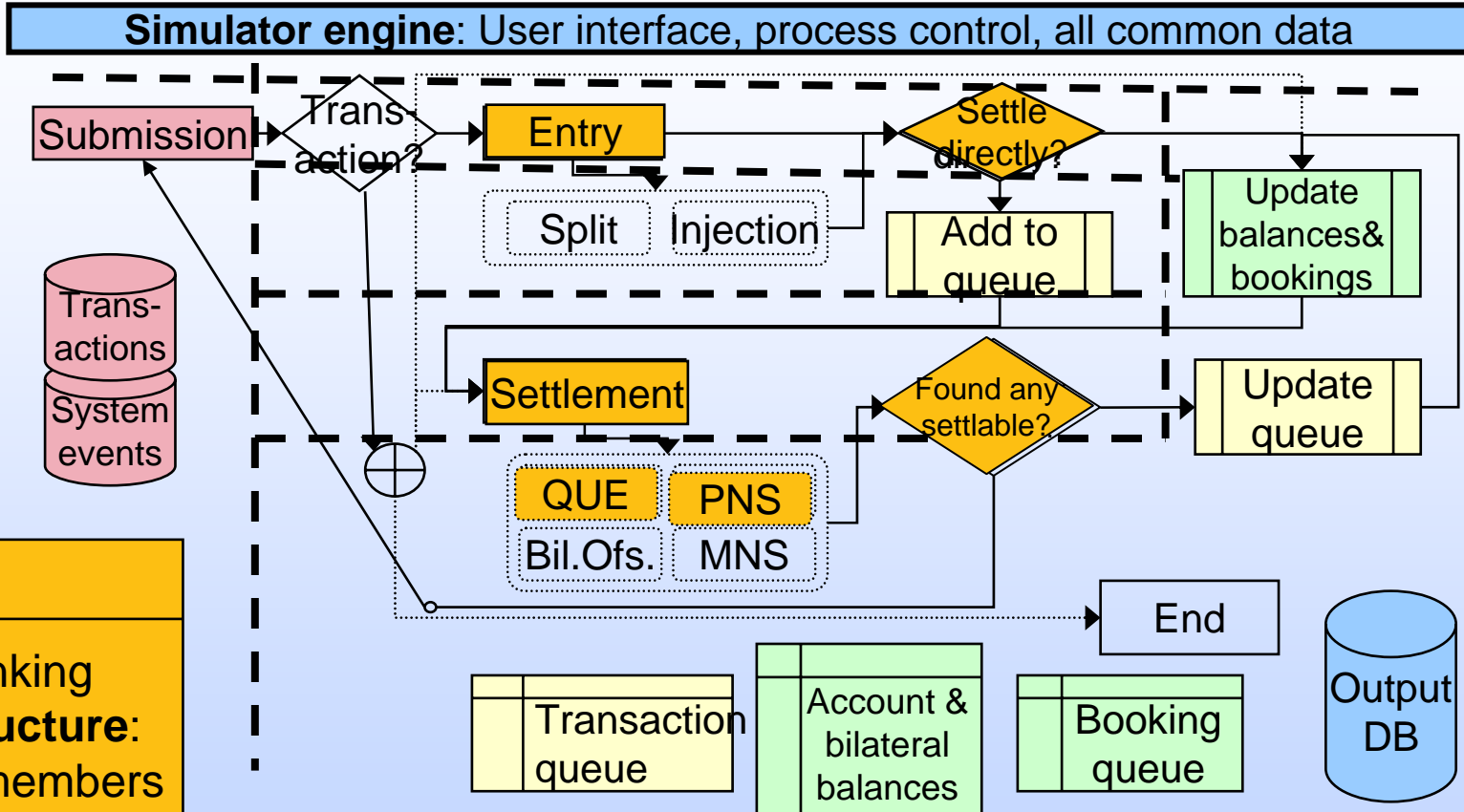
# Case Group codes

- Purpose: To allow efficient linking of arbitrary many transactions together
  - DVP-link only works for pair of transactions
- Additional data required:
  - Group key, count of group members.  
⇒ Usercode 1 & 2 used to import these
- Efficient implementation requires a prepared data structure for linking transactions of one group together.





# GroupCodes: implementation



# Java doc

- New documentation is available of the technical side of the simulator
  - Listings of all methods in all classes
    - Brief descriptions of the most important classes.
- (Extensive) HTML-document created automatically from code



# Behavioral algorithms

- Natural place is in submission:
  - Observe the simulator situation & decide when to submit or delete transactions.
  - It should be possible to write a interface algorithm and combine simulator with some other software.
    - e.g. Matlab has Java-application compatibility.

