BoF-PSS2

Payment and Settlement System Simulator

- A tool for analysis of liquidity, risk and efficiency

Harry Leinonen
Bank of Finland

Many dimensions in payment systems

- Complex processes
  - Settlement account characteristics
  - Processing and settlement algorithms
  - Behavioural patterns and incentives
  - Rules defined by user community and authorities
- Interdependency
  - A network of ancillary systems and main settlement systems
  - PVP and DVP processing
  - International relationships
- Susceptible to external shocks
  - Technical, criminal, liquidity, financial and other shocks
- Payment systems have hidden characteristics
  - Internal credits and counterparty obligations

Why simulate?

- **Simulation models are suitable for analysis of a number of payment system issues**
  - **Incorporation of relationships that are complex and close to reality**
  - **Real and massive data sets can be used**
  - **Results generally reliable when behavioural effects can be controlled or anticipated**
  - **Models based on enumeration rather than calculus**
- **Multiple scenarios can be simulated; impossible in real systems**
  - **Various risk scenarios**
  - **Possible changes in settlement conventions, methods and pricing**
  - **Changes in behavioural patterns and official policies**

New tools needed to understand the complexities and risks in this increasingly critical area

Where is simulation applicable?

- **Payment/settlement system policy**
  - Developing liquidity programs
  - Developing advanced settlement services
- **Payment/settlement system oversight**
  - Analysing settlement, credit and systemic risks
  - Assessing impacts of proposed regulation
- **Payment/settlement system-related research**
  - Gridlock-resolution and liquidity-saving algorithms
  - Changes in payment flow patterns caused by international consolidation and electronification
  - Relationships between liquidity circulation and monetary policy

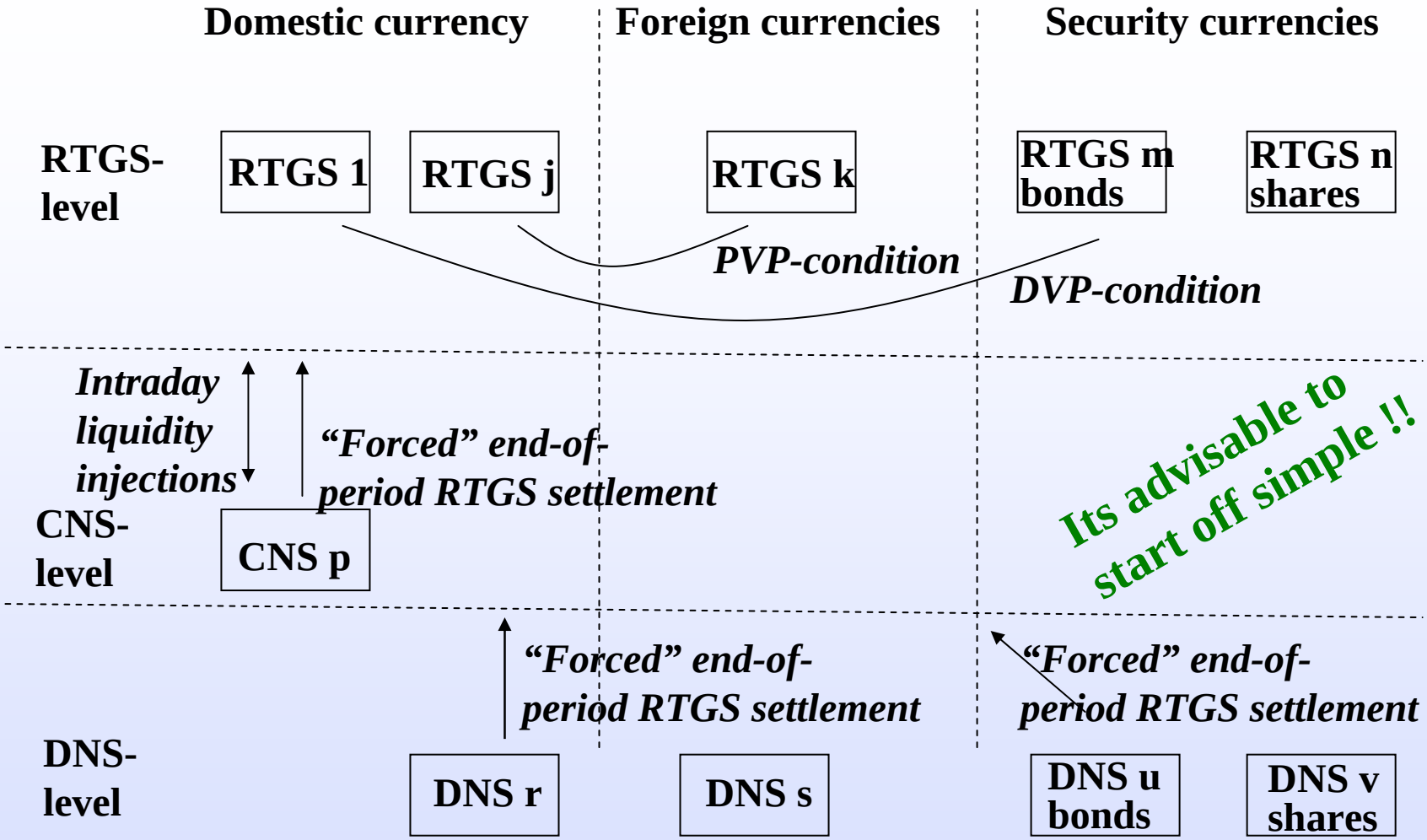
Simulation models can be adapted to many research topics

BoF-PSS Simulator development objectives

- **Professional tool for payment and settlement research**
  - analyses of most common payment and settlement issues
  - large data sets and processing as in real systems
- **Basic services**
  - input, simulation and analysing support
  - most common settlement conventions and algorithms
  - multiple system and currency support
- **Open and common interfaces and standard**
  - Java and MySQL development tools
  - CSV and Excel interface
  - User module expansion possibility
- **User-friendliness**
  - easy-to-use user interface
  - user guide, help-function and tooltips
  - format conversion support (separators, date, time, file format)

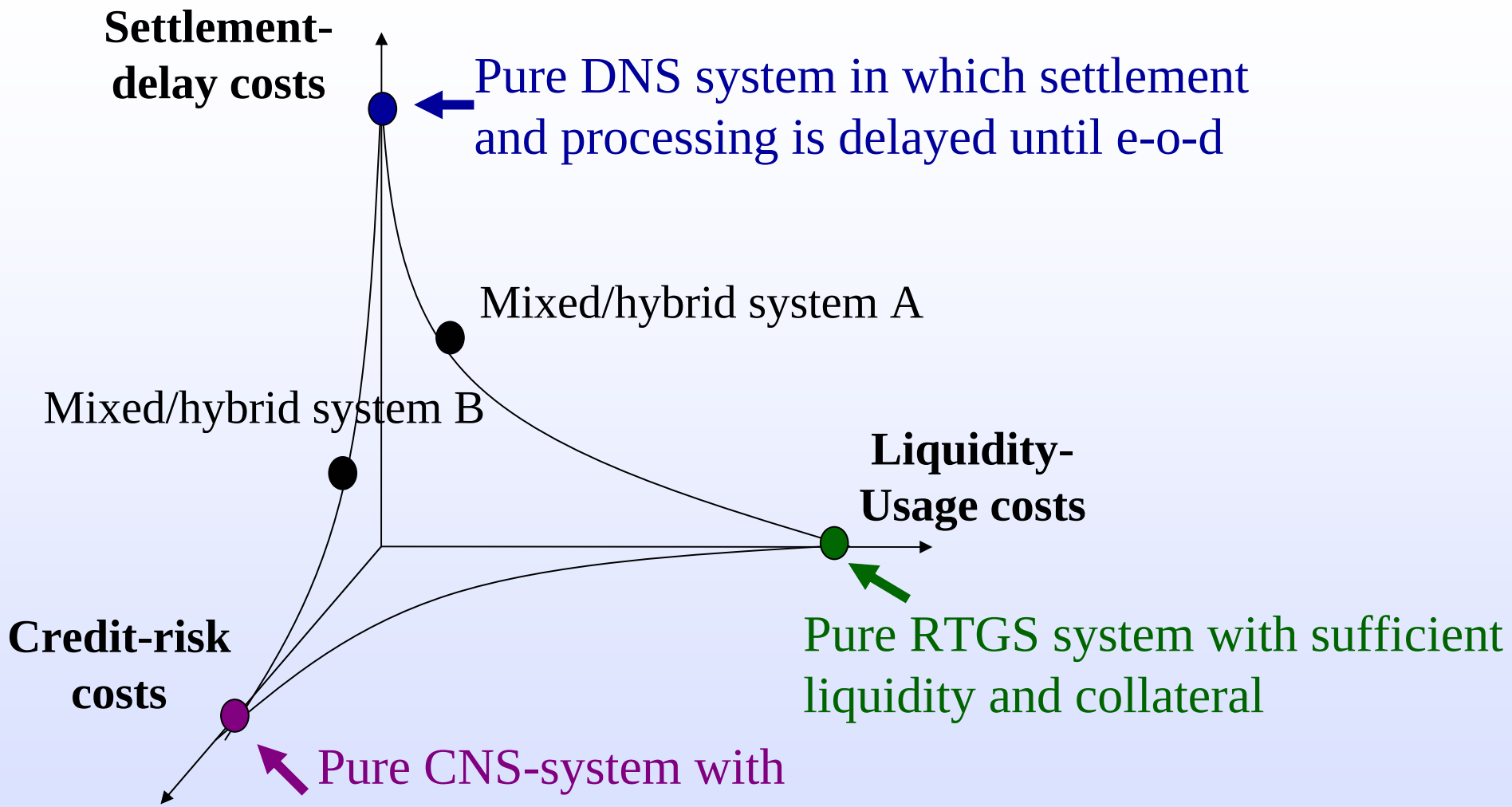
Balancing act between desires, resources and timetables (output and input is partly rough and ext. editors are needed.

Possible system structures in BoF-PSS2



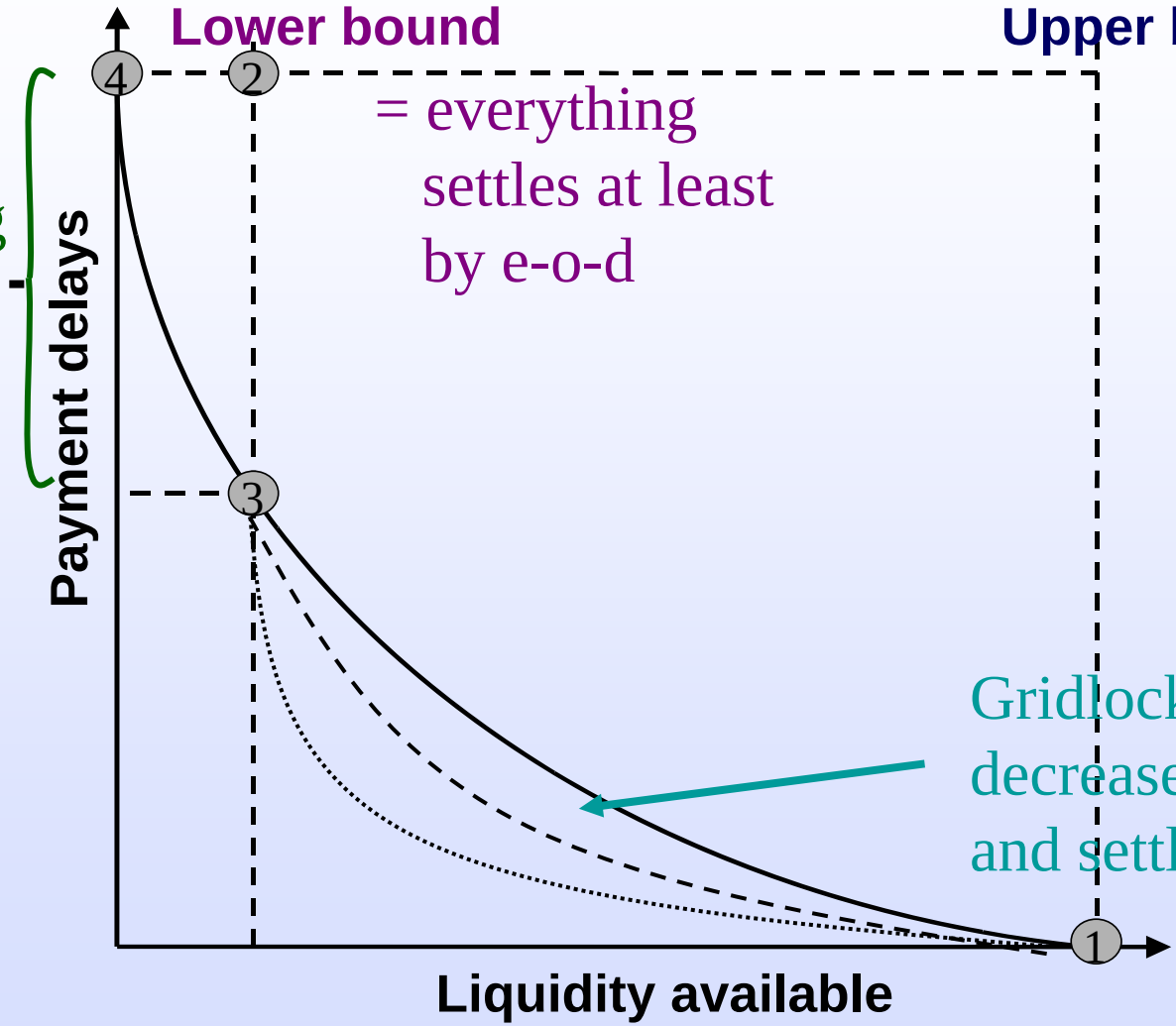
The simulator supports a large combination of different systems on same and different levels and in different currencies.

System type and cost structures



Mixed systems balance cost factors
RTGS speed up processing

RTGS benefit over DNS when same liquidity is used during the day



Gridlock algorithms can decrease liquidity needs and settlement delays

The liquidity needs

- **<u>Upper bound</u> = all transactions can be settled immediately**
  - Run simulation with free credit and Upper bound = participants' maximum intraday negative account balances (or zero)
- **<u>Lower bound</u> = all participants can settle at least by end-of-day and now requirement for settling before that**
  - Runs simulation with free credit and Lower bound = participants' negative end-of-day account balanaces (or zero)
- **<u>Effective lower bound</u> = all participants can settle important payments immediately and the rest at end-of-day**
  - Run simulation with lower bound credit limit and entry algorithm ENTFORURG, which will settle urgent payment violating limits, effective lower bound = participants maximun negative balance (or zero)

Typical studies: **liquidity needs and risks**

- Efficient liquidity usage
  - Too much or too little liquidity
- General liquidity drainage
  - Effects of reduced liquidity (90%, 80% ….)
- Indirect systemic effects
  - One/some system(s) stopped
  - One/some participant(s) stopped
- System problem in liquidity supply
  - Liquidity supply stopped during day
- Liquidity need for priority payments
  - Minimum needs for different payment flows
- Changing liquidity rules
  - New versus old regime comparisons

Typical studies: **credit levels and risks**

- Actual credit risks during day
  - Risky participants, overall risks
- Effects of tighter and slacker credit limits
  - Increased risks versus increased speed
- Indirect systemic effects
  - One (biggest)/some system(s) stopped
  - One/some participant(s) stopped and/or in bankrupcy
- Probability for insufficient collateralisation and cushions
  - Exposures versus reserves
  - Loss sharing exposures
- Changing environment
  - New participants
- Changing credit risk rules
  - New versus old regime comparisons

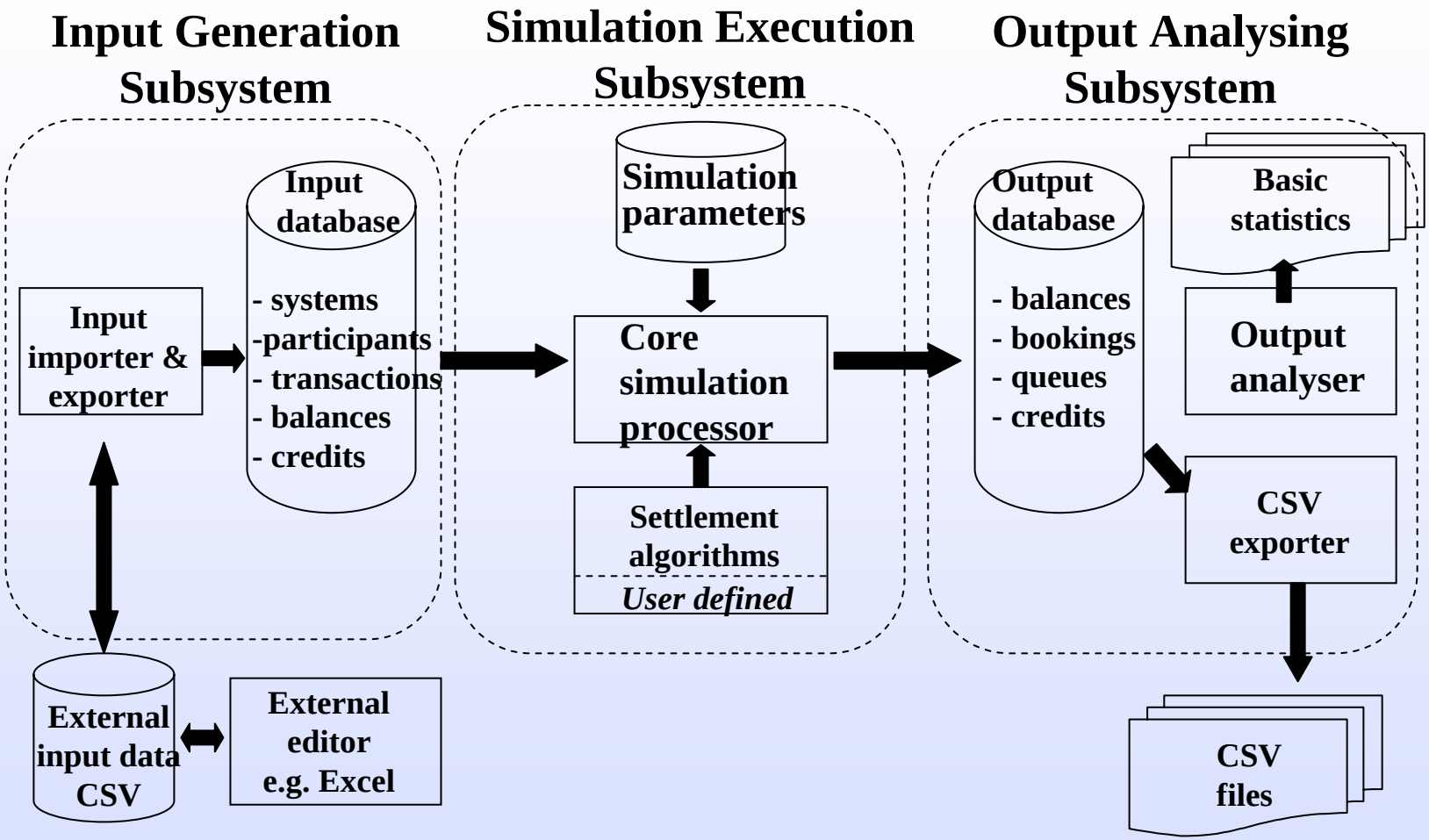
Typical studies: **efficiency**

- Queue length and overall queuing time
  - Delaying participants
- Efficient settlement rules
  - Open hours, syncronisation, pricing
- Efficient settlement algorithms
  - Prioritization, queuing methods, netting features, splitting, intersystem dependencies
- System hierarchy and tiering
- Overall balance between cost of delay, risks and liquidity
- Changing system rules
  - New versus old regime comparisons

Remember that

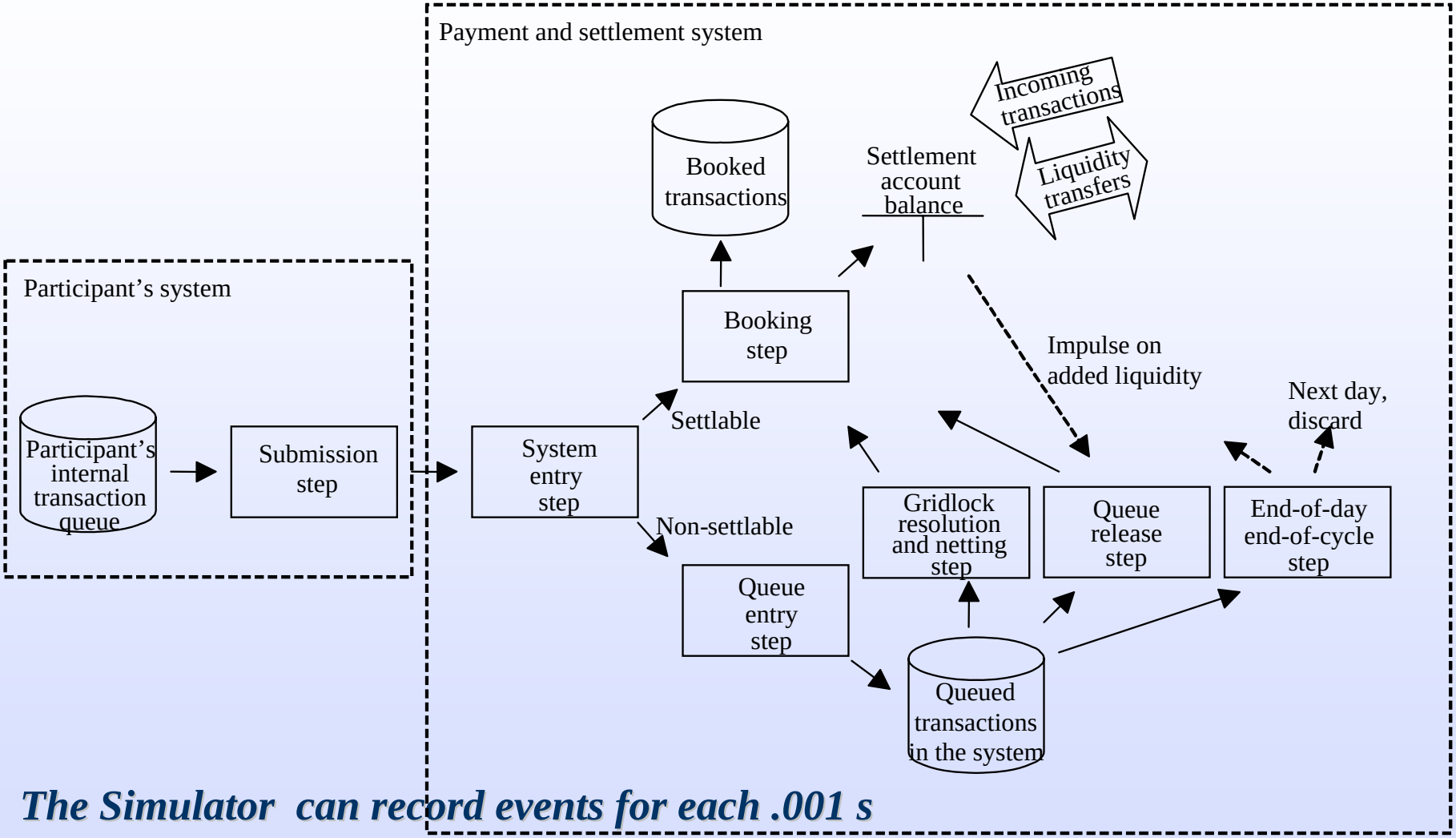
- historic data is history (not the future)
- historic data represents normal situations
- participants change their behavior in new circumstances
- behavior in stress situations can be surprising

Simulator structure



The Simulator is event driven

Events are occurring in the same sequence as in real-systems, but in some processing phase to simulator is faster and in other slower.



The Simulator can record events for each .001 s

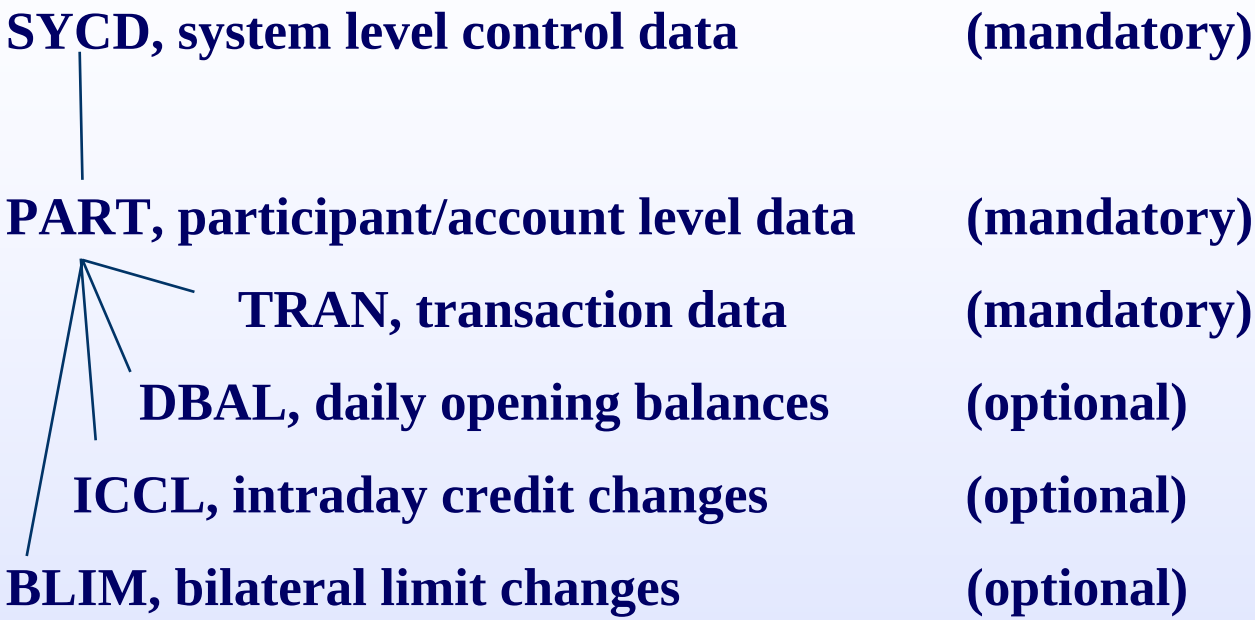
A typical simulation process

- Define input data
  - Participants/accounts, transactions, system data
- Execute simulations
  - Different data sets, algorithms, liquidity, etc.
- Analyse results
  - Compare current with potential system structures/policies and check ‘what if’ special circumstances are realised
- Iterate

Large simulations are time-consuming

- Good data preparation
- Clear simulation plan
- Systematic analysing/reporting concept
- Remember backups

Input database data table structure



All data for the PART, TRAN, DBAL, ICCL and BLIM data tables are imported via CSV-files (comma separated values). SYCD system level data is defined via a separate screen.

Data sets

- Data set IDs allow storage of parallel data tables in data base
- Simulations may use different data sets for varying the input data, e.g. more or less intraday credit, normal or exceptional transaction flows

Use a clear naming convention for different data sets

Main algorithms

- **SUB** (submission) algorithm determines when a transaction is submitted for processing, i.e. chooses next transaction to be processed
- **ENT** (entry) algorithm is first processing phase for a transaction. Generally transferred to bookings when liquidity available; queued/discarded if there is a lack of liquidity
- **SET** (settlement) algorithm processes queued transactions, e.g. invoking gridlock- resolution algorithms
- **END** (end-of-day) algorithm clears up end-of-day situations

Sub-algorithms

- Can be invoked by ENT, SET and END algorithms
- **QUE** (queue release) algorithms release transactions from waiting queues in a defined order
- **SPL** (splitting) algorithms split large transactions into small, easy-to-process transactions
- **INJ** (injection) algorithms transfer liquidity from/to accounts to/from other systems
- **BOS** (bilateral offsetting) nets queued transactions between two counterparties in a given order
- **PNS** (partial net settlement) algorithms seek multilateral payment batches that can be netted
- **MNS** (complete multilateral settlement) netting of all transactions in queues

Expanding the algorithm list

- Current algorithm list includes most common settlement algorithms and conventions
- General and parameter-driven algorithms facilitate user adaptation
- Users can also develop own algorithms with user module interface
- Modular design of algorithms and interfaces facilitates easy expansion

Comments and proposals on algorithm development are always welcome.

User modules for advanced users

- Facilitates introduction of user algorithms
- Java-based
- Standardised simulator interfaces
- Ready-made functions to retrieve data from databases and runtime main memory
- Examples/templates of user module designs
- User modules should conform to algorithm categories

A library of shared user-developed modules could also be distributed eventually

Use of Excel

- Excel is used for viewing CSV-files
- Excel can be used for editing input CSV-files
- Excel can be used for making reports out of CSV-files
- Current Excel versions have a limitation of 65.000 rows
- Excel is often producing extra empty rows/columns (,,,,,)
- Check that delimiters (decimal and data separators) and presentation formats (date and time) are identical with simulator specifications
- Large values may be distorted (less accuracy)
- The actual content of CSV files stored by Excel can be checked with eg Notepad

Use of MySQL

- MySQL provides advanced functions for database management (see www.mysql.com)
- Augments advanced user capabilities when simulator functions are insufficient
- Some special administration features omitted in the simulator and MySQL required when e.g. deleting projects, templates, comparison views, user module definitions and batch-run IDs
- MySQL has ready-made interfaces eg to Access
- Detailed database description available on web-site

Hardware and software requirements

- PC Intel Pentium 4 class processor with min. 256 MB main memory (512 MB or more is recommended for large simulations and 1-2 GB for very large simulations)
- Microsoft Windows NT/2000 or Microsoft Windows XP operating systems with Office/Excel installed
- MySQL database system with Java connector (can be loaded down from Internet without charge)
- For large simulations (millions of transactions and/or over 50.000 accounts) 64bit processors with >8GB memory

A typical simulation site would be a stand alone micro and in the network environment parallel usage of MySQL requires special attention

Ordering and delivery

- The simulator is available free of charge to researchers, but carries no BoF warranties
- Fax ordering,
- Download from the Internet
- Automated installation
- Published research results should be reported
- Users free to make additions (user modules and analysis tools)

Ordering form and guide available at www.bof.fi/sc/bof-pss

Documentation

- Product information sheet
- Presentations, tutorials, example runs(PowerPoint)
- “Simulating Interbank Payment and Settlement Mechanisms” (discussion paper)
- Licensing terms and conditions
- User guide
- Installation guide
- Database description
- User module development guide + java.doc
- Simulation examples with data
- Seminar and workshop proceedings

Documents available at www.bof.fi/sc/bof-pss

User support

- Most user support should be available from help features and documentation
- Bank of Finland provides limited user support (errors, in particular, should be reported)
- MSG Software Oy provides technical assistance and programming services in line with their normal service offerings
- Planned annual seminars and workshops for simulator user community to give opportunities to exchange experiences, views and ideas
- Comments welcome to help us enhance simulator offerings

E-mail address

bof-pss@bof.fi

is available for

- questions

- comments

- ideas

- etc