RISK SYSTEM ARCHITECTURE

Dr Philip Symes
Agenda

I. INTRODUCTION

III. EXISTING RISK ARCHITECTURE

V. CALYPSO RISK ARCHITECTURE

VII. I.T. UPGRADE PHASES

IX. PROJECT WATCHPOINTS

XI. VI. SUMMARY
Introduction

- Three different risk types must be aggregated to calculate a portfolio-level measurement of risk.

- This risk measurement can then be used to calculate the Risk Adjusted Return On Capital:
Introduction

The following constraints on the IT architecture must be matched:

- Maximum possible overlap with the planned credit risk initiatives
- Avoid multiple pricing models
- Minimise need for data mapping and data warehouses; goals:
  - Single point of entry for each data element
  - Single storage location for each data element
- Maximum reusability for existing IT components
- Scalability/Extendibility of proposed architecture
  - New products
  - New analyses/metrics
  - Significantly higher volumes
- Compliant with the bank’s overall IT strategy
Introduction

- Calypso's Enterprise Risk Service (ERS) is a fully integrated part of their Front-to-Back software system.

- Calypso ERS provides the Middle Office risk function.

- Risk control and risk management solution.

- The system is designed to manage:
  - market risk;
  - credit risk;
  - limits.

- The back testing functionality included in the system meets the requirements from Basel2 IRB approach.
Much of the input data is shared, and many of the tools (e.g. pricing models) have common application → Common CORBA based current architecture
Existing Architecture

Existing risk architecture

- FO-Systems adapted to CORBA in Phase 1 approach Credit Exposure Measurement
- CORBA-based infrastructure
- Existing Components needing Market Risk Extensions
- Components Phase 2 approach Credit Exposure Measurement
- Components Phase 1 approach Credit Exposure Measurement
Calypso Architecture

- Calypso ERS separates the user interfaces and back end processes.

- The user interface is the ERS Web Browser:
  - Thin-client presentation layer;
  - Web services – DHTML, XML, etc.

- The back end works using ERS engines:
  - A distributed computing GRID;
  - Service-oriented architecture.

- The ERS engines:
  - Manage portfolio risk analysis;
  - Use existing FO pricing libraries to create P&L vectors.
The Calypso ERS Results service acts as a warehouse of official risk numbers.

1. The service can work with numbers from within Calypso or from an external database.
2. The results can be exported to different formats—e.g., MS Excel.

- P&L vectors are used as the main building blocks of risk analysis:
  - This is because VAR is not a coherent risk measure (see RiskMetrics presentation);
  - In particular, VAR is not subadditive.
Calypso Architecture

- ERS deals with both types of simulated scenarios:
  - Historical simulation;
  - Model base simulation, e.g. MC.

- Portfolios are repriced based on the simulated results
  - This revaluation is done by creating a new pricing environment (PE).

- Trades are revalued using this PE to get a P&L vector of changes in the trade's net present value
  - Each vector element is the P&L for one day in the simulation period.

- VAR is calculated from these P&L vectors
  - They are stored for trades and portfolios.
Calypso Architecture

- Time Series DB
- Mkt Data EOD
- Generate Scenarios
- Trades from GUI
- Trades Feed
- Scenarios Market Data Trades
- ERS Engines
- Simulation Requests
- PnL Vectors VaR
- Portfolio Hierarchy Leaf Nodes

Generate Scenarios

Trades.xml

Results

Aggregate

PnL Vectors

XML

Excel

External System

© Calypso

© Philip Symes, 2006
I.T. Blueprint

The outline for the future of market risk measurement at the bank seems clear . . .

Foundation phase (four months - six months)
1. Refine details of bank wide historical simulation approach, informed by
   - Conversations with regulators
   - Industry best practice
   - IT and software constraints
2. Define additional risk measures and their use
3. Improve and extend the market data supplied by Asset Control
4. Define an appropriate VaR based limit structure
5. Disseminate VaR knowledge throughout appropriate business areas
The outline for the future of market risk measurement at the bank seems clear (cont.) …

Implementation phase (around six months)
1. Implement solution defined in foundation phase

Post implementation phase (four months - six months)
1. Economic capital allocation for all risk sources
2. Introduce risk adjusted performance measures

... but many details still need to be clarified
I.T. Blueprint

Phased approach focusing on regulatory requirements and synergies between market and credit risk architectures

Phase I  Stabilization and Enhancement of the Existing Systems
- Meet regulatory requirements
- Approve data quality
- Get better user buy-in
- Achieve tangible benefits in short term

Phase II  Align Market Risk and Credit Risk Architecture
- Improve flexibility
- Improve time to market for new instruments
- Unified IT architecture
I.T. Blueprint

- Phase I - stabilization and enhancement of existing environment
- Improve Market Data
  - Improve quality of raw data (daily data and historical data)
  - Enhance scope of market data to add missing data items
  - Add additional data suppliers (Telerate, Bloomberg, …)
  - Verify calculation functionality for derived data (i.e. implied volatilities)
  - Reduce implementation time for new data items
- Benefits:
  - Meet regulatory requirements
  - Significantly improve quality of VaR numbers
  - Increase user buy-in
I.T. Blueprint

- Phase I - stabilization and enhancement of existing environment (cont.)

- Increase Scenarii Consistency for Historical Simulation
  - Reach firm-wide agreement on methodology
  - Use Asset Control as only source for market data and scenarii
  - Distribute scenarii to every system performing historical simulations
  - Ensure that each scenario can be uniquely identified across the different systems

- Benefits:
  - Meet regulatory requirements
  - Increase consistency of calculated numbers
I.T. Blueprint

- Phase II - synergy benefits of a shared architecture between market and credit risk system

- Reuse of middleware technology
  - Reuse of components
  - Scenario Generation Service
  - Pricing Engine
  - Translation Engine
  - Aggregation Engine
  - Market Data Service ...
I.T. Blueprint

- Phase II - synergy benefits of a shared architecture between market and credit risk system (cont.)

- Benefits:
  - Avoid remapping of front office trade and position data - only extensions are necessary
  - Ensure consistency between market and credit risk input data
  - Allow consistency between market and credit risk calculations
  - Full scalability and extendibility to needed scope and performance
Phase II - alternatives for scenario pricing in front office systems vs. separate pricing engine

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Pricing in FO Systems</th>
<th>Separate Pricing Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position Data Mapping</td>
<td>Not needed</td>
<td>Mapping needed for each instrument and every source system</td>
</tr>
<tr>
<td>Pricing Functions</td>
<td>‘Best of breed’</td>
<td>Single proprietary pricing function for each instrument</td>
</tr>
<tr>
<td>Performance</td>
<td>Limitations inherent to the system</td>
<td>Scalability enabling multi-process and multi-site</td>
</tr>
<tr>
<td>Market Data Feeds</td>
<td>Connections needed to every FO system</td>
<td>Only one connection needed</td>
</tr>
<tr>
<td>Result Aggregation</td>
<td>Necessity for integrate data from different systems</td>
<td>Flexible aggregation due to consolidated data</td>
</tr>
<tr>
<td>Traceability and Control</td>
<td>End user must master all FO systems or a separate tool is necessary</td>
<td>Only one system involved</td>
</tr>
<tr>
<td>Risk Methodologies</td>
<td>Limitations inherent to the system</td>
<td>Transition of methodologies possible</td>
</tr>
<tr>
<td>Process Alignment</td>
<td>Independent calculation in FO system</td>
<td>Synchronization needed to align all necessary information</td>
</tr>
<tr>
<td>Regulators</td>
<td>To be validated</td>
<td>Has been validated</td>
</tr>
</tbody>
</table>
Watchpoints

- Certain key parts of the project must be monitored
  - See the *POC Kit* for more details.

- The next few slides deal with points in 3 key areas of project implementation from philipsymes.com’s experience:
  - Project level watchpoints;
  - Business level watchpoints;
  - Data watchpoints.
Watchpoints

- The scope of the project:
  - Must have detailed requirements documented;
  - Need to be aware of “scope creep”.

- Production release cycle:
  - Analyse testing requirements (POC);
  - Ensure the release is controlled (implementation).

- All elements of the project should be properly documented.

- Acceptance (success) criteria for the project must be pre-defined.
Watchpoints

- Business needs to understand the risk measures being presented:
  - This includes VAR, stress tests and expected shortfall;
  - Stakeholder involvement is essential.

- Systems integration:
  - Software systems must be compatible;
  - Use banking and industry standards where possible.
Watchpoints

- Data requirements are always underestimated:
  - In house data sources preferable;
  - Need to analyse gaps in required data.

- Data and systems must be integrated:
  - Calypso ERS uses FO pricing libraries;
  - Need the ability to drill-down on data provided;
  - Shared asset control across business units.
Summary

Much of the market risk related activities can proceed in parallel to the credit risk work ...

**IMPLEMENTATION TIME LINE**  
**MARKET RISK**

<table>
<thead>
<tr>
<th>Methodology Foundation Phase</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
<th>Q9</th>
<th>Q10</th>
<th>Q11</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Implementation Phase I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-Implementation Phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IT Implementation Phase II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

light gray = Existing Environment - most issues are already being addressed  
dark gray = Proposed Architecture/ Methodology