Workshop 22: OO Technology in Large Financial Institutions

Monday, October 16, 1995, Austin, Texas, USA.
Report by Chris Laffra, Morgan Stanley, Email: laffra@ms.com.

This workshop brought together people that have experience with applying OO techniques for developing application in (large) financial institutions. We highlighted specific problems, solutions, and lessons learned, together with their individual applicability, strengths, and limitations.

Goals and Focus

Financial institutions are believed to have special requirements:

- Rapid changes. The business changes very fast. Application development is a matter of months, weeks, sometimes even days. The average life span of a typical developer is measured in less than a couple years.
- Difficult technical problems to be solved (such as analysis of a given market.) Challenge is to encapsulate (black–box) legacy systems into the "new" OO environment. These legacy systems often have been developed by programmers that have long since disappeared without making any documentation.
- Reuse. Most firms have settled the "object" reuse matter by reusing many different GUI development toolkits and foundation classes. However, a more important sort of reuse is in developing an enterprise wide reuse strategy, based on distributed OO systems, where end user applications issue request with object servers running on LANs and WANs. Different standards for distribution need to be considered, in addition to performance, robustness, and security.
- "Build versus Buy". We want to have robust software with a large user base, exactly matching our requirements, yet being flexible to the volatile business. Often, time is more important than investing some money in an external tool. However, requirements or bugs will not be taken care of until the "next release". Build versus buy is a difficult tradeoff, and each specific situation will warrant a different decision.
- Choice of language. C++ vs. SmallTalk vs. Objective C vs. APL. Often, languages are chosen with religious arguments. Sometimes for efficiency reasons, ease of development, or other more realistic arguments. Although it always leads to interesting discussions, we want to avoid this matter at this workshop and spend more time on special architectural, environmental, and infrastructural issues.

Organizers

Chris Laffra, Morgan Stanley
Mamdouh Ibrahim, EDS/Object–Oriented and AI Services
Peter Lowes, EDS/Securities Industry Group

Report

We took some risk by not organizing anything prior to the workshop. Our plan was to come up with interesting items during the workshop, discuss them and determine issues, positions, and conclusions during the day. Attendence ranged from institutions like Morgan Stanley, CAL FP bank, Greenwich Captial Markets, American Management systems, Knight Ridder, Ernst & Young, EDS, Front Capital, First Union, to Goldman Sachs. The common background of the attendents was a primarily technical function in a financial institution.

Here follows a summary of the workshop result. Most are rather fragmentary, and need further explanation. I hope that you can get a feeling for the kinds of issues that came up at the workshop, though.

The first hour of the workshop was spent on introductions and determining interesting topics, anywhere in the range of:

- OO business modeling,
- Distribution,
- How to apply OO firm-wide,
- Integration of legacy systems,
- Flexibility and adaptibility
- Class libraries/frameworks for financial domain
- Database integration
- Top-down versus bottom-up use of OOT
- Reuse of objects and process; a myth??
- OO analysis and design methodologies
- Real-time issues
- Rapid changes

After an uncomfortable silence we decided to concentrate on the following three topics:

- modeling,
- distribution,
- reuse

The 24 attendents were split up in three groups, and each one was told to come back right before lunch with a sensible set of issues and positions. We were lucky the thing workded out so well, as we had a lot of very interesting discussions.

As I did not attend all three subsection, most of this summary reflect reports from other attendees, and sometimes I don't even know what they mean with certain bullets. Anyway, it should give you an impression of the kind of issues that were raised at the workshop. A rudimentary summary of the conclusions from the three groups is given here:

- **distribution**.
  We came up with many general isssues (not limited to the financial domain), and few specific solution for our application area. Anyway, The issues directly influencing distribution of object in a financial domain that came up comprised:

  - a quickly changing market situation
  - attrition
  - 7x24 support

  Each of those three aspects directly leads to the need for distribution, or in some way influences distribution. A more general list of issues is formed by:

  - how about load balancing, maintenance, and monitoring. We need tools!
  - real-time. No delays, please.
  - from two-tier towards three-tier architectures.
  - major goal: platform transparancy.
  - componentry; reuse at a higher level.
  - performance.

  Implementation specific aspects that came up were:

  - the web; Java.
  - CORBA vs IP (socket based)

- (distributed) Smalltalk versus C++
- OLE
- Again, support tools????

o **modeling challenges in large financial institutions.**

- enterprise wide business modeling
  - Top–down inititiatives. Not succesful!
  - Bottom–up inititiatives. Result in fragmentation.
  - Approach suggestions:
    - Education of executives ("buy in").
    - Focused Business Objectives in context of enterprise framework.
    - Must have results to spread to other areas.
    - Patterns to seed new projects.
- enterprise wide risk management. Drives cross–functional modeling initiatives:
  - Educate business analysts in OOA&D.
  - Force common methodology??
- Why has so much successful OO development in large financial institutions been restricted to the OTC Derivatives business, especially interest rate instruments?
- How to fit the detailed model's scope to the business cycle time. "Reality constraints".

o **reuse.**

The reuse group came up with a nice table to offset the different aspects influencing and defining reuse aspects:

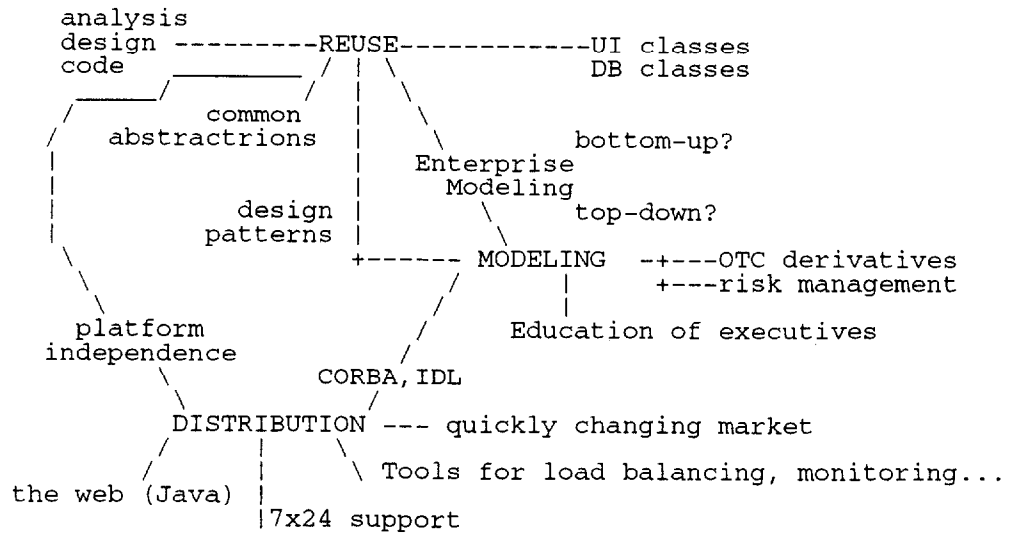
	What?	How?	Examples?
Analysis	- common business representation - common abstractions	enterprise model	intentionally left empty
Design	- problem -> solution - Ideas	- patterns - technology focused model - tools ????	catalogues of design in notes
Code	- Class - Part - Application	- common env. - standards - class libs.	- UI class lib. - DB libraries

Issues:

- must plan for reuse
- common industry models (–> OMG special interest groups.)
- organizational and cultural impacts.
- sponsorships and vision
- cost benefits and infrastructure projects
- methodology and repository
- people
- heterogeneous environments
- common toolset

The afternoon

After lunch time, we went through the list of different issues and tried to put them all into one picture:



From this picture we distilled three main questions, to be answered during the afternoon session:

- What is a **workable** definition for reuse of financial objects?
  - "real" reuse in large financial institutions sadly only happens on an individual, personal basis. How can we extend reuse to reach a larger audience?
  - The "P word". Is a design pattern catalogue the answer? Once determined, should it be mandated, should it be encouraged and suggested?
  - Most important aspect of reuse is organizational. Involves identification and communication of reusable code.
  - Limiting scope of domain increases changes for valuable reuse. Most people in the workshop seem to take a pragmatical approach towards reuse. Which is not surprising, considering our background. We want to concentrate on **concrete** objects, distinguishing it from the reuse **hype**. Simply abstracting an object does not work. Its relationships with other objects needs to be considered (i.e., patterns). Business object are abstract. We don't like/understand them...

- The "Enterprise Model" -- or, how do we make a valid business case for OO investments in financial institutions?
  Objectives of an enterprise model:
  - Identify "core" objects and define distinct firm-wide objects, and develop a common understanding of these objects. Then, develop recurring design patterns.
  - Presentation of the model.
  - Allows us to manipulate the business, because we now understand it a lot better.

  We could identify certain abstract objects, like "position", however these yet undefined object should be representated as blank spots in our enterprise model, allowing us to proceed with the process. Individual application areas of the enterprise model would fill in the empty blanks, and instantiate the model for the particular business.

  A top down approach does not work [remember, this is the opinion of technical people]. We should leave freedom in the design and do interpretation based on different applications. Agreed, this is a pragmatic HACK, and not a generic solution to the problem.

Certain characteristica and observation were that an enterprise model represents a high level, broadly scoped view of the enterprise. Its goal is stability over time. It is not meant to be a mold for implementing concrete code. No **fundamental** changes in the way we do business are anticipated, right?

- How to enforce common OO standards and integrate legacy systems in heterogeneous environments?
  First of all, let's consider different aspects of distribution models:
  - Assume: light-weight clients
  - Assume: heavy-weight server
  - We will continue to suffer from standard wars.
  - How to solve expected network latency?
  - How to integrate TP monitoring like DCE?
  - How to abstract interfaces, to make us independent of existing "standards"?
  - Impementation independent analysis models (templates)

  Windows NT is a target "standard" for desktops. How to integrate with existing/planned OO distribution standards?
  We should keep an eye on trends, watching the future... [seems a very empty remark, now that I type it in. It made sense during the workshop, though :-)].
  Question: How to use object distribution in heterogeneous environments? Answer: Simple, use abstraction of interfaces.
  A common problem that was identified was how to map business processes into objects?!?!

This workshop lacked a set of concrete results. That was anticipated, as it was the first one of its kind at a conference like OOPSLA. Perhaps the workshop raised more questions than answers, and it certainly addressed the fact that the use of OO technology in large financial institutions is not something one can cover in simply one day at a workshop. It also shows that the use of OO technology, especially when OO modeling, reuse, and distribution is concerned still needs a lot of time to mature.

On-line version

Submitted position papers can be found at: http://home.navisoft.com/peterl/oopsla95.htm

Atttendees

Mamdouh Ibrahim	EDS/OOAIS	mhi@gmr.com
Peter Lowes	EDS Global Sec. Ind.	fignet03.peterl@eds.com
Martin Fowler	independent consult.	100031.3311@compuserve.com
Patrick Fitzgeralnd	Morgan Stanley	pfitz@ms.com
Paul Kritz	Export Developm. Corp.	krizpa@edc3.edc.ca
Xavier Alvarez	Lifia/UNLP - Argentina	xavi@info.unlp.edu.ar
Richie Bielak	CAL FP	richieb@calfp.com
Chris Brown	First Union Nat. Bank	ckb@vnet.net
Sharon Lubert	First Union Nat. Bank	ott@vnet.net
Gabe Beged-Dov	Rogue Wave Software	gabe@roguewave.com
Martin Buis	Ernst & Young Fin.Syst.	102114.1011@compuserve.com
Cyrus Deyhimi	BSG Alliance/IT Inc.	cyrus_deyhimi@notes.bsginc.com
David Dixon	Ernst & Young Fin.Syst.	dixond@ey.geis.com
Micheal Kishinevsky	Knight-Ridder Finance	73621.703@compuserve.com
David Dnytryk	Export Developm. Corp.	70612.1546@compuserve.com
Chris Laffra	Morgan Stanley	laffra@ms.com
Guy Rackham	Goldman Sachs	----
Doris Tyghe	USAA	doristighe@aol.com
Steven Feldman	Jump Start Systems	sfeldman@intac.com
Jerry Tylman	American Mgmt. Sytems	gerald_tylman@mail.amsinc.com
Gordon Sheppard	American Mgmt. Sytems	gordon_sheppard@mail.amsinc.com
Kuan-Tsae Huang	IBM	huangkt@vnet.ibm.com
Mikael Hammarlund	Front Capital Systems	mikaelh@front.se
Tomas Stenlund	Front Capital Systems	tomass@front.se
Rebecca Burkey	Greenwich Capital Syst.	burkeyr@gcm.com