Formal Specification of an Auction System Using VDM++ and UML

An Industrial Usage Report

M. van den Berg, M. Verhoef and M. Wigmans
Chess Information Technology
the Netherlands
Flower Auction (1)

- largest covered market place in the world
- spot market for fresh cut flowers and plants
- “Dutch auction” process
- flowers are (re)distributed all over the world
- 13 auction clocks
- 4 auction halls
- 50000 transactions per day
- 4 million EUR daily turnover
- mission critical!
Flower Auction (2)

- set clock in starting position
- start clock by auction master
- clock moves counter clockwise @ 30 steps / sec
- buyer presses Stop button
- buyer negotiates by voice with auction master about the amount
- auction master generates the transaction
- clock moves clockwise, back to starting position @ 60 steps / sec
Why this project? (1)

• Buyers demand close integration into their IT systems to improve performance and reduce costs
  – better match supply and demand
  – optimize logistics to customer
  – improve lead time financial transactions

• Do not want to be physically present at auction theatre (influences product price)

• Electronic Commerce
Why this project? (2)

• In the past, the flower auction bought their clock systems in the market place (closed systems)

• Due to need for close integration with buyers’ systems, need to get in control of system development again

• Strategic choice: build own system
  – Start by re-engineering the existing clock system
  – Implement “Distance Buying”
  – Define a future clock system architecture

• Our role: supplier of knowledge and resources
Project prerequisites

• System should be implemented using commercial-off-the-shelf products based on open standards (no proprietary solutions)
  – Auction side:
    • industry standard VME bus computer systems
    • POSIX compliant real-time operating system: VxWorks
    • fast Ethernet tcp-ip backbone for real-time data
    • ip multicasting, voice over ip
    • ISDN PRI routers allow remote access using ppp links
  – Buyers’ side
    • personal computer running NT 4.0
    • ISDN BRI interface card
    • audio card
Project setup

- **Network group**
  design and implement the auction network backbone

- **Hardware group**
  define the hardware architecture of the auction system

- **Software group A**
  design and implement the auction interface application

- **Software group B**
  design and implement the auction buyer’s application

- **VDM++ and UML used in Software group A**
Why Formal Methods?

- New application area for flower auction
- No existing tools or techniques implemented in the organization
- Unclear requirements
- Large knowledge gap between designers and implementers (need for concise and elaborate specification)
- Some experiments with UML (Rational Rose)
- Previous successful experience on a case study using IFAD VDM++ Tools and Rational Rose
How To Evaluate?

- Evaluate against three well-known papers on the use of formal methods:
  - “Seven Myths of Formal Methods”, [Hall90]
  - “Seven More Myths of Formal Methods”, [Bowen&95a]
  - “Ten Commandments of Formal Methods”, [Bowen&95b]

- Originally published in IEEE Software / Computer
- Reprinted in “High Integrity System Specification and Design”, FACIT series, Springer Verlag, 1999
I : Thou shall choose an appropriate notation

- Determining the actual problem is non trivial task
  - little time to decide at start of project
  - biased by clients perception of the problem

- Choice is heavily influenced by non-technical issues
  - Client might have strong preference for (non-formal) methods or tools, local flavor or dialects
  - Need to be “politically correct” - mainstream IT

- Combination VDM++ and UML is strong
  - UML is de-facto industry standard
  - Good tools for both methods exist
II : Thou shall formalize but not over formalize

- Organizations can only deal with limited amount of change (no giant leap for mankind)
  - adapt level of formalism to level of maturity of the organization
- UML in itself is not sufficient
  - Need higher level of specification detail, pseudo code level
  - OCL is too restricted for this purpose
  - Quote from building team member: "VDM++ fills in the UML gaps"
- Introduction of formal methods introduces “side effects” - less elbow room for clients.
III : Thou shall estimate costs

- Not very specific for formal methods
- Be aware: more time is spent in specification phase
- Use cyclic delivery techniques to show progress in specification phase (DSDM workshops)
- Builders used the VDM++/UML model to estimate the time needed to implement the system
- Very accurate planning was possible
- Very good progress monitoring during implementation phase
IV : Thou shall have a FM guru on call

- Again not very specific for formal methods
- In industrial environment:
  - project management = risk analysis and reduction
  - lack of knowledge = project risk
  - hire outside help (I.e. Chess)
V : Thou shall not abandon thy traditional development methods

- Formal Methods do not make problems less complicated - just better tools to attack problems
- Still need for software process (project phasing, quality assurance, version and configuration management)
- Integration with existing standards and techniques (to compensate each other weaknesses) is essential
- Using another paradigm forces you to look at the problem from another perspective; this enhances specification completeness
VI : Thou shall document sufficiently

- Industrial applications = long life time
- Large number of non-technical requirements that can not / need not be expressed using formal methods
  - business strategy
  - system context
  - overall design decisions
  - systems architecture
- Challenge: keep documentation in sync with formal specification
  - Review ^ 3
  - Integrated tools (VDMTools, Rose, Word, SourceSafe)
VII : Thou shall not compromise thy quality standards

• Good tools do not imply quality products!

• Quality standards “ensure” a review process

• Forces client involvement (whether they like it or not)

• If no quality standard exist, introduce one!
VIII : Thou shall not be dogmatic

- In industry, there is only one dogma:
  
  “Client is King”

- Make what the client wants as soon as possible and at reasonable cost

- Use of formal methods if there is a clear benefit

- We consider it to be “good engineering practice”
XI : Thou shall test, test and test again

- Validation: Did we build the right product?
- Verification: Did we build the product right?

- In industrial applications: use of COTS products
- In embedded context: push these products to their limits - need for exhaustive testing

- Formal Methods provide excellent framework for testing
X : Thou shall reuse

• Reuse is of no use in technical automation:
  – once off systems
  – low volume
  – highly specific application area

• extra investment to make reusable components is not regained fast enough (ROI)

• Reuse only knowledge gained from projects