Patterns and Processes for Increasing the Business Reactivity with MDA

Birol Berkem

GOObiz - Goal Oriented Business

www.goobiz.com

Synchronize your IT with your changing business.
Business Reactivity with MDA

**Business Reactivity**: Response time necessary to a business system for implementing requested changes as a reaction to a new requirement, in its controlled process of adaptation to its environment.

Birol Berkem

How MDA can help us to increase the business reactivity? (to reduce the response time to react to changes)
Review of MDA Principles for Increasing the « Business Reactivity »

• **Portability** : Platform Independency for Specifications. To do this : Specifications need to be rendered **Identifiable, Evolutive and Executable**

• **Interoperability** : Business components need to communicate for realizing **coherent behaviors** according to **high-level business strategies**

• **Reusability** : Ability for applications to **use changing business behaviors** with a total transparency when changes happen!

• Primary goals of MDA : Text for an MDA Guide : ormsc/02-08-01- J.Miller
Portability (Platform Independency for Specifications)

Specifications need to be **Identifiable, Evolutive and Executable**

- **IDENTIFIABLE SPECIFICATION (PIS)**: Makes identifiable specifications using Goals (Classes) and Aptitudes (Operations)
- **EVOLUTIVE SPECIFICATION (PES)**: Assures easy evolutions to system behaviors by allowing traceability of specifications of a given refinement level toward lower-level specifications using goal-oriented refinement techniques
- **EXECUTABLE SPECIFICATION (PEX)**: Guarantees platform independency of specifications at a given abstraction level by rendering them executable without using particularities of the immediate lower-level

A Goal-Case contains a set of responsibilities that belong to the same unit of intention.

<<<< Goal Case>>>

**Goal-Oriented Objects GOObiz**
PI-S - Pattern for Identifiable Specifications

A Goal-Case is created in the system when a complex requirement is received by the system. It contains a set of responsibilities that belong to the same unit of intention.

Goal-Oriented Object (GOO) : A Platform Independent Structure

GOObiz

List of High-Level Strategic Goals

- G1 : Increase volume of transactions (500 Transactions a day)
- G2 : Increase rate of visits
  - G2.1 Links from other sites
  - G2.2 Site reviewed in medias
- G3 : Increase rate of visits
  - G3.1 Links from other sites
  - G3.2 Site reviewed in medias
- G4 : Invent a bonus system
- G5 : Increase the Internet Site
- G6 : Increase profits for Sales
- G7 : Increase profit of the Internet Site

Visitor [Registration] (Goal-value = 100 registrants a week,)

- entered_ok : boolean
- bonus_affected : boolean
- lottery_realized : boolean
- visitor_notified : boolean

+ register_visitor()
+ cancel_register()
+ modify_register()
- enter_visitor()
- fill_questionnaire()
- notify_visitor()
- notify_rate_of_registrant()
- abort_transaction(...)

Activities of a Goal-Case may be identified via an Activity Diagram...

Description for the Goal-Case

Register Visitor -V1
(Goal-value = 100 registrants / week, duration = 'till new order', Means = Lottery and Bonus)

Process of Development for "Increasing Market Parts -V1"

- G1 : Enhance Production Process
  - G1.1: Efficient Production
  - G1.3 : Efficient Purchase
- G2 : Motivate staff (means ...)
  - G2.0 : Develop a program for staff motivation
  - G2.1 Communicate program
  - G2.2 Accompany program
- G3 : Increase rate of visits
  - G3.1 Increase rate of visits
  - G3.2 Motivate visitors to Register via a bonus system
- G4 : Enhance productivity of the Production and Delivery Chains
  - G4.1 Increase profitability of the Production
  - G4.1.1 Products conform to Mktg Specs
  - G4.1.2 Respect delays for production
- G5 : Increase registered 
  - G5.0 : Develop a program for visitors
  - G5.1 Communicate program
  - G5.2 Accompany program
- G6 : Increase profits for Sales
- G7 : Increase profit of the Internet Site

- G4.2 : Products conform to Mktg Specs
- G4.2.1 : Products conform to Mktg Specs
- G4.2.2 : Respect delays for production

- G6.0 : Develop a program for visitors
- G6.1 Communicate program
- G6.2 Accompany program

- G7.0 : Develop a program for visitors
- G7.1 Communicate program
- G7.2 Accompany program

- G8 : Increase reliability
- G9 : Enhance production process
- G10 : Motivate staff

- G4.3 : Invent a bonus system
- G4.4 : Attractive pricing

- G5.3 : Attractive pricing
- G5.4 : Enhance reliability

- G6.3 : Invent a bonus system
- G6.4 : Attractive pricing

- G7.3 : Invent a bonus system
- G7.4 : Attractive pricing

- G8.3 : Attractive pricing
- G8.4 : Enhance reliability

- G9.3 : Attractive pricing
- G9.4 : Enhance reliability

- G10.3 : Attractive pricing
- G10.4 : Enhance reliability
PES - Pattern for Evolutive and Traceable Specifications

- Guarantees evolutivity of specifications by providing explicit description of complex behaviors at each refinement level and by assuring traceability toward lower levels.
- Each complex operation in a GOO may be identified as a separate GOO within a Component of Goal-Oriented Objects (GOO_Comp).

Visitor [Registration] {Goal-value = 100 registrants a week ...}

entered_ok : boolean
bonus_affected : boolean
lottery_realized : boolean
visitor_notified : boolean

+register_visitor()
+cancel_register()
+modify_register()
-enter_visitor()
-fill_questionnaire() {Pre : v_entered}
-notify_visitor() {Pre : quest_filled}
-notify_rate_of_registrant()
-abort_transaction()

A GOO_Comp regroups behaviors that act as Controller, Machinery, Exception or Post-Actions in the achievement of operations of their controller.
PEX - Pattern of Executable Specifications

- Guarantees independency of specifications from elements of lower abstraction levels by rendering them executable at each level.

- Requirement Analysis focus on elements of the: ‘Functional What’
- System Analysis focus on the ‘Functional How’
- Design focus on elements of the ‘Technical What and How’

---

**CONTRACT for Register_Visitor()**

**Pre-Conditions:**
- visitor_connected

**Post-Conditions:**
- visitor_registered = (visitor_entered and questionnaire_filled and visitor_notified);

**Exceptions:**
- [visitor_abandoned] : transaction_aborted;

---

**CONTRACT for Notify_Visitor()**

**Pre-conditions:**
- questionnaire_filled

**Post-conditions:**
- visitor_notified

---

**Visitor [Registration]**

- visitor_registered : boolean
- visitor_entered : boolean
- questionnaire_filled : boolean
- visitor_abandoned : boolean

- register_visitor() {Pre :v_connected}
- cancel_register()
- modify_register()
- enter_visitor()
- fill_questionnaire() {Pre :v_entered}
- notify_visitor() {Pre :quest_filled}
- notify_rate_of_registrant()

Guarantees platform independency of specifications at a given abstraction level by rendering them executable without using particularities of the immediate lower-level.
PEX - Pattern of Executable Specifications
From Contract to the Execution

Visitor [Registration]

- visitor_registered : boolean
- questionnaire_filled : boolean
- link_visitor_interests : boolean

+ register_visitor()
+ cancel_register()
+ modify_register()
- enter_visitor()
- fill_questionnaire()
- notify_visitor()
- notify_rate_of_registrant()
- update_visitor_interests()

Visitor [Entry]

- visitor_entered : boolean

+ visitor_entry (name, e-mail)
+ cancel_entry()
+ find_article(designation)

...
**Interoperability** requires communication between components to realize **coherent** behaviors according to business strategies - It necessitates:

(a) respect to high-level business goals and constraints in face of changes
(b) communication of required behaviors to actors of the application layer

Assures coherent evolution to the system by allowing its components integrate:

- Respect to high-level business goals
- Controls to assure feed-backs in order to bring corrections to the running system

(a) **COHERENT EVOLUTION-(PCE)**

(b) **TRACEABLE ABSTRACTION LAYERS-(PTAL)**

**EXECUTABLE SPECIFICATION-(PEX)**

**EVOLUTIVE SPECIFICATION-(PES)**

Affects relevant responsibilities to the actors of the application system layer
(a) PCE - Pattern for Coherent Evolution- Mapping Changes

WebSite_Mngt [Turn to Buyer] {initial constraints}

Product [Promotion] {..}

Visitor [Registration] {...}

Visitor [Registration]
(Goal-value = 50 registrant/a week)

Visitor [Notification]
[entry ok]

Visitor [Entry]

Existing Business Architecture

To assure coherent evolution to the system

Questions are:
• What are top level components that might be concerned by the changes?
• Which controls become necessary in order to bring corrections to the running system?

G 3.2 - Motivate Visitors to Register via a bonus system
(Goal-value = 100 registrants a week)

G3.2.1: Register Visitor()

G3.2.2: Fill Questionnaire
– G3.2.2.1 If the abandon rate reaches x %, trigger the process of Review of the Questionnaire

G3.2.3: Once visitor is registered, compute bonus and launch lottery system
– G3.2.3.1 If winner rate is reached, start the process of Review_LotteryRules
– G3.2.3.2 If stock_alert, process Material Purchase

G3.2.4: Notify results to visitor by e-mail, once lottery and bonus calculation are completed

New Requirements to Map
(a) - Pattern for Coherent Evolution - Mapped Changes

WebSite_Mngt [Turn to Buyer] {Motivate Visitors to Register with a Bonus System}

Product [Promotion] {Motivate Visitors to Register with a Bonus System.}

Visitor [Registration] {with a Bonus System
  Goal-value = 100 registrant per a week.}

Visitor Notification

Bonus Assignment

Bonus Rules [Review]

Lottery [Realization]

{Goal-value =
  1 winner /50
  goal-value <1 winner
  00 registered visitor

{winner_rate > 1/100}

Lottery [Realize]

Visitor [Entry]

Visitor [Registration]

{with a Bonus System}

visitor entered

[visitor entered]

[register]

Visitor Notification

Material [Purchase]

I-Purchase

Assures coherent evolution to the system by allowing its components integrate:

- Respect to High-level business goals
- Controls for ensuring feed-backs in order to bring corrections to the running system

GOObiz

Goal-Oriented Objects™
(b) PTAL - Pattern for Traceable Abstraction Layers...

WebSite_Mngt [Turn to Buyer] {Motivate Visitors to Register with a Bonus System}

Product [Promotion] {Motivate Visitors to Register with a Bonus System..}

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System}

Visitor [Notification]

Visitor [Entry]

Visitor [Registration] {with a Bonus System}

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]

Visitor [Registration] {with a Bonus System
Goal-value = 100 registrant per a week..}

Visitor [Entry]
Reusability: Ability to use changing business behaviors within applications with a total transparency from changes.
PUB-BAL - Pattern for Using Business Behaviors

BUSINESS SYSTEM

WebSite_Mngt [Turn to Buyer] {Motivate Visitors to Register with a Bonus System}

Product [Promotion] {Motivate Visitors to Register with a Bonus System}

Visitor [Registration] (with a Bonus System)

Company [Presentation] (Promote Registration with a Bonus System)

Visitor [Entry]

Visitor [Registration] (with a Bonus System)

Visitor [Notification]

Visitor [Assignment]

Bonus [Critical rate]

Lottery [Realization] (Goal-value = 1 winner /100)

Lottery [Realize] (goal-value < 1 winner)

Lottery [Realize] (goal-value = 0 winner)

Lottery [Realize] (goal-value > 1/100)

Visitor [Notification]

Visitor [Assignment]

Bonus [Critical rate]

Lottery [Realization] (Goal-value = 1 winner /100)

Lottery [Realize] (goal-value < 1 winner)

Lottery [Realize] (goal-value = 0 winner)

Lottery [Realize] (goal-value > 1/100)

Visitor [Notification]

Visitor [Assignment]

Bonus [Critical rate]

Material [Purchase]

Purchase [Material]

Purchase_Emp

Mktg_Emp

Goal-Oriented Objects
GOObiz
Summary of Patterns for Business Reactivity with MDA

Assures coherent evolution to the system by allowing its components integrate
- High-level business goals
- Controls to assure feedbacks in order to bring corrections to the running system

COHERENT EVOLUTION-(PCE)

TRACEABLE ABSTRACTION LAYERS-(PTAL)

EXECUTABLE SPECIFICATION-(PEX)

USE BUSINESS BEHAVIORS-(PUB-BAL)

IDENTIFIABLE SPECIFICATION-(PIS)

EVOLUTIVE SPECIFICATION-(PES)

Affects relevant responsibilities to actors of the application system layer

Guarantees platform independency of specifications at a given abstraction level by rendering them executable without using particularities of the immediate lower-level

Makes identifiable specifications using Goals and Aptitudes

Assures easy evolutions to system behaviors by allowing traceability of specifications of a given refinement level toward lower-level specifications using goal-oriented refinement techniques

GOObiz

Goal-Oriented Objects
Goal Driven Development with MDA for a Business Reactive to Changes

1. Process Stratégique d’Accroissement
de Parts de Marché - V1

Liste des Objectifs Stratégiques
- GL : Motiver le personnel (moyen : prime par lot de vente)
- GL : Motiver la Direction (moyen : prime par lot de vente)
- GL : Augmenter les ventes (moyen : prime par lot de vente)
- GL : Augmenter le profit (moyen : prime par lot de vente)
- GL : Augmenter la rentabilité (moyen : prime par lot de vente)
- GL : Augmenter la productivité (moyen : prime par lot de vente)
- GL : Augmenter la qualité (moyen : prime par lot de vente)
- GL : Augmenter la satisfaction du client (moyen : prime par lot de vente)
- GL : Augmenter la fidélité des clients (moyen : prime par lot de vente)
- GL : Augmenter la rentabilité du site internet (moyen : prime par lot de vente)
- GL : Augmenter la rentabilité des sites internes (moyen : prime par lot de vente)

2. Impact Appropriate Components by Related Changes

3. Structure Goals by Components

Build Goal-Cases

4. Structure Goals by Components

5. Integrate Reactions in the Business Architecture

6. Integrate Use Cases Constraints in the Overall System Architecture

Goal-Oriented Objects
DESIGN USE CASES, BASED ON RESPONSIBILITIES IDENTIFIED IN THE SYSTEM ARCHITECTURE

GATHER APPLICATION REQUIREMENTS USING SCENARIOS

IMPLEMENT COMPONENTS IN THE TECHNICAL ARCHITECTURE

DETAIL APPLICATION REQUIREMENTS THAT USE BUSINESS COMPONENTS

Specify Constraints Upon Domain Classes / GOO_COMP

APPLICATION - CTRL

ENTITIES - CTRL-ACCESS

USE COMPONENTS OF THE BUSINESS ARCHITECTURE
(TRANFORMED FROM AN UNDERSTANDABLE LANGUAGE AFTER VALIDATION BY NON-TECHNICAL PEOPLE)