Problem Frames and UML development

Christine Choppy (LIPN, Université Paris XIII, France)
Gianna Reggio (DISI, Università di Genova, Italy)
Issues

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
Software development

Main characteristics
  Abstraction
  Structuration

Patterns

Reuse

Requirements
  Specification

Design

Code

Issues

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
Software development

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame

Issues

Main characteristics
- Abstraction
- Structuration

Patterns

Reuse

Use cases
- Context diagram

Requirements
- Specification

Design

Code

Problem Frames

Architectural styles
- Design Patterns
- Code patterns

Design Patterns

Code patterns
Software development

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame

Main characteristics
- Abstraction
- Structuration

Patterns

Reuse

Design patterns

Architectural styles

Code patterns

Use cases
Context diagram

Problem Frames

Requirements
Specification

Design

Code

Natural language
- UML
- Formal specifications

Graphics

Programming language
Overview

Issues

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
Overview

Issues

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
Problem Frames (M. Jackson)

- can be used by themselves or in combination
- a good tool to tackle with a first structuring of problems
- diagram: involved domains, requirements, design, interfaces
- five basic problem frames, variants

We propose associated development methods
Overview

Issues

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
A basic problem frame: transformation

Transform machine

Input

Output

IO relation

IN!Y1

TM!Y2

X

X

Y3

Y4
Transformation Frame: Domain Model and Requirement Specification
Problem Frames and UML development

Christine Choppy (LIPN, Université Paris XIII, France)
Gianna Reggio (DISI, Università di Genova, Italy)

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame

Transformation Frame: Design Specification

<table>
<thead>
<tr>
<th>Input</th>
<th>IRelation</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransformMachine</td>
<td>method transform(): Output</td>
</tr>
<tr>
<td>transform: Output</td>
<td>....</td>
</tr>
</tbody>
</table>
Mailfiles Analysis Case Study

**Mail**
**Analysed**

**Mailfiles**

**Analysis**
**rules**

a: MF!{Mail, File, From, To, Length}
b: MA!{ReportLine}
c: {Mail, From, To, Length}
d: {ReportLine}
Mailfiles Analysis: Domain Model and Requirement Specification

inv:
for all rep ∈ analysisRules
for all rl ∈ rep.lines
(exists m ∈ files.mails s.t. m.to = rl.name or m.from = rl.name) and
rl.inNum = {m | m ∈ files.mails and m.from = rl.name}→size and
rl.inAvgLth = {m | m ∈ files.mails and m.from = rl.name}→length.sumAll / rl.inNum and
rl.inMaxLth = {m | m ∈ files.mails and m.from = rl.name}→length.max and
rl.outNum = {m | m ∈ files.mails and m.to = rl.name}→size and
rl.outAvgLth = {m | m ∈ files.mails and m.to = rl.name}→length.sumAll / rl.outNum and
rl.outMaxLth = {m | m ∈ files.mails and m.to = rl.name}→length.max

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame

Message
msg, from, to: String
date: Date
length: Int

ReportLine
name: String
inNum, inMaxLth, inAvgLth: Int
outNum, outMaxLth, outAvgLth: Int

Report
lines {ordered}

Mailfile
files {ordered}

Mailfiles

analysisRules
1..*
Overview

Issues

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
The Commanded Behaviour frame

- **Controlled domain**
- **Controlled machine**
- **Operator**
- **Commanded behaviour**

E4 are operator commands, C1 are Pulses and C2 are Sensors.

CM!C1
CD!C2
OP!E4
C1, C2
E4
Commanded Behaviour: Domain Model

**ControlledDomain**

- +sensor\_1: ...
- ...
- +sensor\_k: ...
- ...
- private attributes...
- -auto\_1(...)
- ...
- -auto\_n(...)

**Pulses**

- pulse\_1(...)
- ...
- pulse\_n(...)

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame
Commanded Behaviour: Requirement Specification - Use case Diagram

ControlMachine

Command1

... 

Commandm

OP: Operator

Require normal behaviour

CD: ControlledDomain

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame
Commanded Behaviour: Requirement Specification - Class Diagram

**ControlledDomain**

- `+sensor_1`: ...
- `...`
- `+sensor_k`: ...
- `... private attributes...`
- `-auto_1(...)`
- `...`
- `-auto_n(...)`

**Pulses**

- `pulse_1(...)`
- `...`
- `pulse_n(...)`

**Commands**

- `command_1(...)`
- `...`
- `command_m(...)`

**Operator**

**ControlMachine**
Lift System Case Study

**Issues**

**Motivation**

**Problem Frames**

**The transformation frame**

**The Commanded Behaviour frame**

**The Commanded Information frame**

**The Rich Workpieces frame**

Christine Choppy
(LIPN, Université Paris XIII, France)

Gianna Reggio (DISI, Università di Genova, Italy)
Lift Case Study: Domain Model - Class Diagram

**Problem Frames and UML development**

Christine Choppy (LIPN, Université Paris XIII, France)
Gianna Reggio (DISI, Università di Genova, Italy)

**Issues**
**Motivation**
**Problem Frames**
**The transformation frame**
**The Commanded Behaviour frame**
**The Commanded Information frame**
**The Rich Workpieces frame**
Lift Case Study: Domain Model - Behaviour of LiftPlant

- **MovingUp**
  - after 0.1 s / cabinHeight = cabinHeight + 0.4
  - motorStop
  - motorUp
  - enters: [doorPos(cabinPos) = open] / peopleInside = peopleInside + 1
  - leaves: [peopleInside > 0 and doorPos(cabinPos) = open] / peopleInside = peopleInside - 1

- **Stopped**
  - closeDoor(f) [f = cabinPosition] / doorPos(f) = closed
  - openDoor(f) [f = cabinPosition] / doorPos(f) = open
  - motorStop
  - motorDown

- **MovingDown**
  - after 0.1 s / cabinHeight = cabinHeight - 0.4
  - motorStop

**Issues**

**Motivation**

**Problem Frames**

**The transformation frame**

**The Commanded Behaviour frame**

**The Commanded Information frame**

**The Rich Workpieces frame**
Lift Case Study: Requirement Specification - Use Case Diagram

U: User

L: LiftPlant

LiftController

StopCabin

Call cabin at a floor

Send cabin to a floor

Cabin must wait at ground floor
Lift Case Study: Requirement Specification - Class Diagram

**LiftPlant**
- `+cabinPos: Int`
- `+doorPos(Int): doorPosition`
- `+noneInSide: Boolean`
- `+cabinHeight: Float`
- `+peopleInside: int`
- `+enter`
- `+leave`

**Pulses**
- `openDoor(Int)`
- `closeDoor(Int)`
- `MotorStop`
- `MotorDown`
- `MotorUp`

**Commands**
- `stop`
- `callAt(int)`
- `sendTo(Int)`

**Controller**

**User**
Use Case Description: Call the cabin at floor f
Commanded Behaviour Frame: Design Specification

ControlMachine

CD: ControlledDomain
command₁(…)
...
commandₘ(…)

Waiting

Command₁

…

Commandₘ

…

NormalBehaviour

…

Problem Frames and UML development
Christine Choppy (LIPN, Université Paris XIII, France)
Gianna Reggio (DISI, Università di Genova, Italy)

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame
Lift System Case Study: Design Specification

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame
Overview

Issues

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
The Commanded Information frame

C1 phenomena are referred to later as Events issued by the Real world
E1 are Enquiries from the Enquiry operator
E2 are Display Acts and Error Messages displayed by the Answering Machine to the Enquiry Operator
Case Study: Company Information System
Company Information System: Domain Model

**Order**
- ofType: ProdCode
- howMuch: Int
- byWho: ClientCode
- code: OrderCode

**ProdCode**

**OrderCode**

**ClientCode**

**Company**

The Company is a commercial one selling products of various kinds, produced by someone else. The orders are received from outside, and from time to time they are examined. If the ordered products are available in the required quantity the order is processed, an invoice is sent to the client and the goods are shipped, otherwise the order will be examined again in the future. If the ordered products are not available for a long time, the order is refused. A client may cancel an order before it is processed. From time to time the products are supplied by the producers and stocked by the Company. The Company product catalog may change, that is products may be removed and new ones added.
Commanded Information: Requirement Specification - Use Case Diagram

EO: EnquiryOperator

AnsweringMachine

Enquiry_1

... 

Enquiry_n

Record Event

RW: RealWorld_E
Commanded Information: Requirement Specification - Class Diagram

Event

History
- cont: Sequence(Event)
  add(e:Event) {post: cont = cont@pre->append(e)}
  hop₁(...)
  ...
  hopₙ(...)

RealWorld

RealWorld_E

Signals

<<interface>>
signalEvent(Event)

AnsweringMachine

his: History

Enquiries

<<interface>>
enquiry₁(...)
...
enquiryₙ(...)

<<interface>>

ErrorMessage

error₁(...)  
...
errorₖ(...)

DisplayActs

dAct₁(...)  
...
dActₘ(...)  

EnquiryOperator
Commanded Information: Requirement Specification - Use Case Descriptions

Use case Record Event

Recording

signalEvent(e) / his = his.add(e)

Answering

enquiry(X)

[ok_cond(X,his)] / EO.dAct(his.hop(X))

[err_cond₁(X,his)] / EO.error₁(...)

[err_cond₁(X,his)] / EO.errorₙ(...)

Generic use case Enquiry
Company Information System: Requirement Specification - Use case diagram

**CompanyManager**
- Show product availability
- Show product quantity sold in a period
- Show how many orders received in a period
- Show how many pending orders
- Show refuse rate
- Show cancellation rate
- Record Event

**CompanyIS**
Commanded Information: Requirement Specification - Class Diagram

- **Company**
- **Company_E**
  - **Signals**
    - `signalEvent(Event)`
  - **CompanyManager**
    - **Enquiries**
      - `available(ProdCode)`
      - `sold(ProdCode, Date, Date)`
      - `recOrders(Date, Date)`
      - `refusalRate()`
      - `cancelRate()`
      - `pendOrders()`
    - **ErrorMessages**
      - `notInCatalog(ProdCode)`
      - `notTimePeriod(Date, Date)`
    - **DisplayActs**
      - `availableIs(ProdCode, Int)`
      - `soldIs(ProdCode, Date, Date, Int)`
      - `recOrdersAre(Date, Date, Int)`
      - `refuseRates(Real)`
      - `cancelRates(Real)`
      - `pendOrdersAre(Int)`

**Problem Frames and UML development**

Christine Choppy (LIPN, Université Paris XIII, France)
Gianna Reggio (DISI, Università di Genova, Italy)

**Issues**

**Motivation**

**Problem Frames**

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
Show product quantity sold in a period

![Diagram]

- **Answering**
- Sold quantity: `sold(pc,d_1,d_2)`
- Conditions:
  - `[his.inCat(pc) and d_2 >= d_1]` / `CM.soldIs(pc,d_1,d_2,his.sQuant(pc,d_1,d_2))`
  - `[d_1 > d_2]` / `CM.notTimePeriod(d_1,d_2)`

**Problem Frames and UML development**

Christine Choppy (LIPN, Université Paris XIII, France)
Gianna Reggio (DISI, Università di Genova, Italy)

**Issues**

**Motivation**

**Problem Frames**

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
Commanded Information: Design Specification - Static View

![Diagram of Commanded Information]

**EnquiryOperator**

- **Enq₁**: `enquiry₁(...)`
- **Enq₂**: `enquiry₂(...)`
- ...  
- **Enqₙ**: `enquiryₙ(...)`

**Hop₁**

**St₁**

**EventRec**

**RealWorld_E**

**Issues**

**Motivation**

**Problem Frames**

- The transformation frame
- The Commanded Behaviour frame
- The Commanded Information frame
- The Rich Workpieces frame
Overview

Issues

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
E1, E2 are resp. LOOKS and OPERATIONS
S (SERVICES) are U!COMMANDS–S, DT!MESSAGES–S, DT!SHOWS

Y2 are MESSAGES
Plumber’s Friend Case Study

Documents

Plumber

Pservices definitions

Plumber’s friend

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame

a: PF!{{SQLSelect}, {SQLInsert, SQLUpdate}}
b: PSERVICES = {AddClient, AddPart, CreateInvoice, ComputeBudget, ChangeClientAddress, ChangePartPrice, UpdateInvoice}
c: {SQLSelect}
Rich Workpieces Frame: Domain Model

Christine Choppy (LIPN, Université Paris XIII, France)
Gianna Reggio (DISI, Università di Genova, Italy)

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame
**Plumber’s Friend : Domain Model**

![UML Diagram]

**Problem Frames and UML development**

Christine Choppy
(LIPN, Université Paris XIII, France)

Gianna Reggio (DISI, Università di Genova, Italy)

**Issues**

**Motivation**

**Problem Frames**

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
Rich Workpieces Frame: Requirement Specification - Use Case Diagram
Plumber’s Friend : Requirement Specification - Use Case Diagram

Problem Frames and UML development
Christine Choppy (LIPN, Université Paris XIII, France)
Gianna Reggio (DISI, Università di Genova, Italy)

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame
Rich Workpieces Frame: Requirement Specification - Class Diagram

```
Workpieces

<<interface>>
LOOKS
lk₁(...) ... lkₘ(...) ...

<<interface>>
OPERATIONS
op₁(...) ... opₙ(...) ...

<<interface>>
MESSAGES
mes₁(...) ... mesₖ(...) ...

DesktopTool
... attributes to keep trace of the execution stages of the services ...

<<interface>>
Shows
......

<<interface>>
Commands
......

<<interface>>
UMessages
......

User
```

---

Issues

Motivation

Problem Frames

The transformation frame

The Commanded Behaviour frame

The Commanded Information frame

The Rich Workpieces frame
Use Case (Service) Create invoice

Problem Frames and UML development

Christine Choppy (LIPN, Université Paris XIII, France)
Gianna Reggio (DISI, Università di Genova, Italy)

Issues
Motivation
Problem Frames
The transformation frame
The Commanded Behaviour frame
The Commanded Information frame
The Rich Workpieces frame
Use Case Create invoice - statechart

createInvoice()
/ ci = create(Invoice); ci.status = "pro-forma";
clsi = DOCS.select(ClientsP); P.show(clsi)

DefineClient

setClient(cr) / ci.client = cr

Filling

Deleting DeleteLine

Modifying ModifyLine

Adding AddLine

Saving

[ DOCS.all->includes(ci) ]
Rich Workpieces Frame: Design Specification

![Diagram showing the structure of the Rich Workpieces Frame with nodes for Workpieces, Operation&Messages&Looks, Ex_1, Ex_h, S_1, S_n, and User.]

### Issues
- Motivation
- Problem Frames
- The transformation frame
- The Commanded Behaviour frame
- The Commanded Information frame
- The Rich Workpieces frame