Business Processes Modelling

MPB (6 cfu, 295AA)

Roberto Bruni

http://www.di.unipi.it/~bruni

04 - Models and Abstraction
Overview of the conceptual models and abstraction mechanisms in business process modeling

Ch.1 of Workflow Management: Models, Methods, and Systems
Ch.1 of Business Process Management: Concepts, Languages, Architectures
Ch.3.1--3.3 of Business Process Management: Concepts, Languages, Architectures
Some definitions
Workflow management coalition (WfMC)

Founded in the ‘90s by vendors, users, academia:
fix standard for Wf representation and execution

http://www.wfmc.org
Workflow

Definition: a workflow is the automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.
Definition: a workflow management system is a software system that defines, creates, and manages Wfs execution, running on one or more workflow engines, able to interpret the workflow definition, able to interact with workflow participants, and able to invoke the use of IT tools and applications.
Kinds of workflow

**Definition**: a *system workflow* consists of activities that are implemented by software systems without any user involvement.

**Definition**: Workflows in which humans are actively involved and interact with information systems are called *human interaction workflows*. 
Human interaction workflows

Goal:
support automation by driving the human activities according to the process model

Benefits:
reduce idle periods
avoid redundant work
improve human/machine work integration
Human collaboration

When task performed by humans are involved in the workflow, it is not sufficient to equip workers with adequate software:

their collaboration must be supported

shared data repositories and work handover can speed-up office procedure considerably
Example: Human interaction workflow
Some limitations

Problems with knowledge workers:

User acceptance issues

Machine burdening of workers

Little room for creativity and flexibility
Business process

Definition: a **business process** consists of a set of activities that are performed in coordination in an organizational and technical environment.

These activities jointly realize a business goal.

Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations.

- Weske
Definition: **business process management** includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes.

- Weske
Business process management

The basis of business process management is the **explicit representation** of business processes with their **activities** and the **execution constraints** between them.

Business processes can then be subject to **analysis, improvement, and enactment**.
Definition: **business process management system** is a generic software system that is driven by explicit process representations to coordinate the enactment of business processes.

- Weske
Business process model

Definition: business process model consists of a set of activity models and execution constraints between them.

- Weske
Business process instance

Definition: *business process instance* represents a concrete case in the operational business of a company, consisting of activity instances.

- Weske
Model and instances

Each activity model acts as a blueprint for a set of activity instances

Each business process model acts as a blueprint for a set of business process instances (related to cases)
Abuse of notation

If no confusion is possible, the term activity is used to refer to activity models as well as activity instances.

Analogously, the term process is used to refer to process models as well as process instances.
Process-driven software

Business process models are the main artifact for implementing business processes.

This implementation can be done by organizational rules and policies, but it can also be done by business process management (software) system.

In this case the software system is driven by explicit process representations (models).
Da BPMN a Petri nets

Nella seconda fase si è proceduto con la trasformazione dei diagrammi BPMN in Petri nets, quindi con l’analisi di quest’ultime.

Le reti sono state costruite utilizzando il tool WoPeD ed applicando fedelmente le regole di conversione da BPMN a Petri nets.

Analisi semantica e grafo di raggiungibilità

Dall’analisi semantica eseguita mediante WoPeD, la rete, contenente 36 piazze e 40 transizioni, risulta essere:

- free-choice, cioè ogni volta che c’è un arco \((p, t)\), allora c’è un arco da ogni piazza di input di \(t\), ad ogni transizione di output di \(p\);

- s-coverable, ogni piazza, infatti, appartiene all’unico \(s\)-component presente; da qui si può inoltre dedurre che è possibile derivare \(s\)-invariants positivi.

- well structured, caratteristica garantita dall’assenza di TP-handles e PT-handles (in particolare nessuno dei flussi alternativi creati mediante i diversi XOR-splits risulta sincronizzato mediante AND-joins);

- workflow net, ovvero è presente un’unica piazza iniziale \(\text{start}\) con \(\text{start} = 0\), un’unica piazza finale \(\text{end}\) con \(\text{end} = 0\) e tutte le altre piazze e transizioni appartengono al cammino da \(\text{start}\) ad \(\text{end}\);

- sound, la rete risulta soddisfare i seguenti vincoli:
  - non sono presenti dead tasks (né dead transitions, né non-live transitions);
  - la piazza finale è raggiungibile da qualsiasi marking \(M\) (option to complete);
  - quando un token raggiunge la piazza finale, tutte le altre piazze sono vuote (proper completion).

Trattandosi di una workflow net, free-choice e sound, un’altra caratteristica garantita è la safeness. Infine, poiché il marking iniziale non è un home marking (cioè non può essere raggiunto da qualsiasi marking), la rete non è cyclic.

Presentazione del caso

Il presente caso di studio descrive lo scenario di un paziente che deve sostenere un esame medico presso una clinica privata. Il paziente può prenotare la visita mediante il sistema di prenotazioni, a cui può accedere inserendo le proprie credenziali; se non è già iscritto, deve compilare un modulo elettronico coi propri dati, per ottenerle.

Una volta che il paziente seleziona la visita specialistica alla quale è interessato, il sistema entra in contatto col medico, che fissa la data dell’appuntamento.

Ricevuta la data stabilita dal medico, il paziente può decidere se confermare, cancellare o chiedere di modificare la data, proponendo una data alternativa. Il sistema dunque riferirà al medico, rispettivamente la conferma, la cancellazione o eventualmente la richiesta di una nuova data. Rispetto quest’ultima circostanza, il medico a sua volta confermerà la data proposta dal paziente o ne proporrà un'altra.

Dopo la visita, il paziente riceve la fattura per il pagamento e poi il referto.

Modellazione in BPMN

In una prima fase, il processo è stato modellato mediante la Business Process Modeling Notation. Per la gestione del caso, è stato realizzato un collaboration diagram che vede coinvolti i tre partecipanti: paziente, sistema di prenotazioni e medico. Ogni partecipante è rappresentato dal rispettivo pool.

Per la modellazione del processo è stato utilizzato il tool Bizagi Process Modeler.

Visual representations: diagrams and charts understandable by humans (informal, intuitive, BPMN, EPC, BPEL)

Languages: unambiguous machine syntax (process dialects, XML schemes)

Models: rigorous semantics for scientists (automata, Petri nets, workflow nets)
Models and abstraction
Models

A model is a simplified representation of reality

"Essentially all models are wrong, but some are useful"
(George P. Box)
Abstraction

To derive general rules and concepts from specific examples of some phenomenon, by selecting only the aspects which are relevant for a particular purpose

A way to cope with complexity
Abstractions

**Horizontal**: separation at different modeling levels

**Aggregation**: separation at different granularity levels

**Vertical**: separation at different subdomains
Horizontal Abstraction
Horizontal abstraction
(modeling levels)

Abstract entities to define concepts
M3: Meta-Metamodel
Instance-of
M2: Metamodel
describes
Concepts that discipline model definition
M1: Model
Instance-of
Classes of similar instances
M0: Instance
describes
Concrete entities

Graphical symbols (different notations for the same metamodel are possible)
Notation

(better be read bottom-up)
Process models and process instances
A process metamodel (level M2)

Fig. 3.13. Model for process models: process metamodel, MOF level M2

Each node is associated with at least one edge. The different types of nodes are represented by the generalization relation. Activity models reflect the work units to be performed, event models represent the occurrence of states relevant for the business process, and gateway models represent execution constraints of activities, such as split and join nodes.

While the association between nodes and edges are defined at the node level, the cardinality of the association between special types of nodes (activity models, event models, and gateway models) differs. Each activity model has exactly one incoming and one outgoing edge.

Each process starts with exactly one event, the initial event, and ends with exactly one event, the final event. Therefore, certain events can have no incoming edges (initial event) or no outgoing edges (final event). Gateway models represent control flow. Therefore, they can act as either split nodes or join nodes, but not both. Hence, each gateway model can have multiple outgoing edges (split gateway node) or multiple incoming edges (join gateway node).

Figure 3.14 shows a process model based on the process metamodel introduced. The notation used to express this process model is taken from the Business Process Model and Notation:

- Event model nodes are represented by circles; the final event model is represented by a bold circle.
- Activity models are represented by rectangles with rounded edges.
- Gateway models are represented by diamonds.
- Edges are represented by directed edges between nodes.
A process model

1. Enter credit request
2. Analyze client
3. Propose Decision
4. Finalize Decision

Event (node) → Activity (node) → Activity (node) → Activity (node) → Event (node)
Some process instances

1. Enter credit request
   - Analyze client
   - Propose Decision
   - Finalize Decision

2. Enter c.r. (r017, Miller, 10000)
   - Analyze client (r017, Miller)
   - Propose Decision (r017, Miller)
   - Finalize Decision (r017, Miller)

3. Enter c.r. (r018, Brown, 15500)
   - Analyze client (r018, Brown)
   - Propose Decision (r018, Brown)
   - Finalize Decision (r018, Brown)

4. Enter c.r. (r019, McGraf, 12000)
   - Analyze client (r019, McGraf)
   - Propose Decision (r019, McGraf)
   - Finalize Decision (r019, McGraf)

5. Enter c.r. (r020, Carey, 20000)
Process models and process instances
An example: MOF metamodel (OMG)
Aggregation Abstraction
Aggregation abstraction

Multiple elements of a lower level of granularity can be grouped and represented by a single artefact at the higher level of granularity

Different from horizontal abstraction, where all entities lie at the same level of granularity
A sample aggregation

OrderManagement

GetOrder

CheckOrder

AnalyzeOrder

SimpleCheck

AdvCheck
Vertical Abstraction
Guiding principle

Separation of Concerns (SoC)
(to separate a system into distinct features that overlap in functionality as little as possible)
Edsger Wybe Dijkstra was one of the most influential members of computing science’s founding generation. Among the domains in which his scientific contributions are fundamental are

- algorithm design
- programming languages
- program design
- operating systems
- distributed processing
- formal specification and verification
- design of mathematical arguments

In addition, Dijkstra was intensely interested in teaching, and in the relationships between academic computing science and the software industry.

http://www.cs.utexas.edu/users/EWD/
On the role of scientific thought (EWD447)

Let me try to explain to you, what to my taste is characteristic for all intelligent thinking.

It is, that one is willing to study in depth an aspect of one's subject matter in isolation for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects.
We know that a program must be **correct** and we can study it from that viewpoint only; we also know that it should be **efficient** and we can study its efficiency on another day, so to speak. In another mood we may ask ourselves whether, and if so: why, the program is **desirable**.

**But nothing is gained** —on the contrary!— **by tackling these various aspects simultaneously.**

It is what I sometimes have called **the separation of concerns**, which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of.

**It does not mean ignoring the other aspects**, it is just doing justice to the fact that **from this aspect's point of view, the other is irrelevant.**

**Business data processing systems are sufficiently complicated to require such a separation of concerns.**
On the role of scientific thought (EWD447)

... and the suggestion that in that part of the computing world "scientific thought is a non-applicable luxury"

puts the cart before the horse: the mess they are in has been caused by too much unscientific thought....
SoC: an example

HyperText Markup Language (HTML):
organization of webpage content

Cascading Style Sheets (CSS):
definition of content presentation style

JavaScript (JS):
user interactions
<!DOCTYPE html>
<html>
<head>
    <style>
        body { background-color: lightblue; }
        h1 { color: darkblue; text-align: center; }
        p { font-family: verdana; font-size: 20px; }
    </style>
</head>
<body>
<h1>HTML, CSS and JAVASCRIPT</h1>
<button type="button" onclick="document.getElementById('demo').innerHTML = Date();">Click me to display Date and Time.</button>
<p id="demo"></p>
</body>
</html>
Vertical abstraction (domain separation)

BPM includes multiple modelling domains, integrated by Process Modelling

- Business Process Modelling
- Process Modelling
- Function Modelling
- Information Modelling
- Organization Modelling
- IT Landscape Modelling
Function models

Units of work enacted by processes
(at different levels of granularity)

Informal description, textual documents
(coarse-grain business level)

Formal description, function specifications
(fine-grain software layer)
High-level business functions

The value chain of a company has a rich internal structure, consisting of a set of coarse-grained business functions (e.g. Order management, Human resources)

High-level business functions can be decomposed into finer-grained functions (this is called functional decomposition) (e.g. from “Order management” to “storing” and “checking” orders)
Activity models and activity instances
Information models

Data representation is crucial: all decisions made during a business process depends on data values.

Data dependencies between activities are also important (ensure data-availability, reduce waiting time).
Data models

M2: Metamodel
(data meta model, e.g., Entity Relationship Model)

M1: Model
(data model)

M0: Instance
(data values)

Notation
(data model notation, e.g., ER Notation, UML Class Diagrams)

Instance-of
 expresses

describes
Organizational models

Organizational structure must be represented

Activities must be associated to specific roles or departments
Organizational models

M2: Metamodel
(organization meta model)

M1: Model
(organization model)

M0: Instance
(persons, e.g.,
knowledge workers, managers)

Notation
(organization model notation, e.g.,
Organization Chart)

Instance-of

describes

expresses
Roles

Roles are groups of employees that qualify for being responsible of certain activities.

Increased flexibility: different persons can cover the same role at different time in different cases.
An organizational metamodel
IT landscape

Many activities in a business process are supported by information systems

Information systems and programming interfaces needs to be represented because they provide functionalities
Interface Definition
Languages

M2: Metamodel
(Interface Definition Languages)

M1: Model
(Interface Definitions)

M0: Instance
(Executing Software providing Defined Functionality)

Notation
(IDL specifications)

Instance-of

Instance-of

expresses

describes

describes
Process models

Define the glue between the subdomains

Relate functions and execution constraints

Relate data values with process instances
(e.g. the process of a credit approval may depend on the requested amount)
A process model with role information
A process instance with workers information

Clerk

Peter

Enter c.r. (r017, Miller, 10000)

Manager

Charles

Analyze client (r017, Miller)

Anne

Propose Decision (r017, Miller)

Jane

Finalize Decision (r017, Miller)