University of Pisa

MSc in Computer Engineering

Process-driven Information Systems

LECTURE 9

http://www.iet.unipi.it/m.cimino/wdis/

Mario G.C.A. Cimino

Department of Information Engineering

An example of social enactment

A business collaboration on the order planning of a machinery

*Pilot scenario. T*he participants involved in the business are (on the left in figure): the client, the mechanical and the electrical firms. Both design and development activities (in the middle), are made of two main tasks: a mechanical task and an electrical task, carried out by the two respective firms. Finally, the management activity (on the right) consists in the coordination of the participants and in the orders planning tasks. With regard to the orders planning, each company schedules tasks on the basis of its own private business rules.

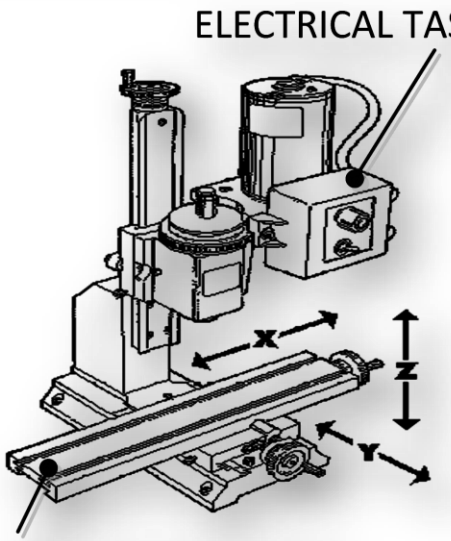
PARTICIPANTS

CLIENT

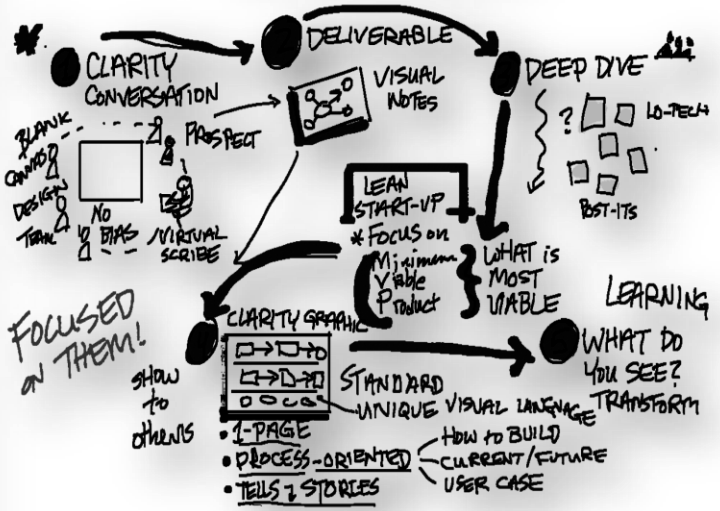
MECHANICAL FIRM

ELECTRICAL FIRM

DESIGN AND DEVELOPMENT



MANAGEMENT



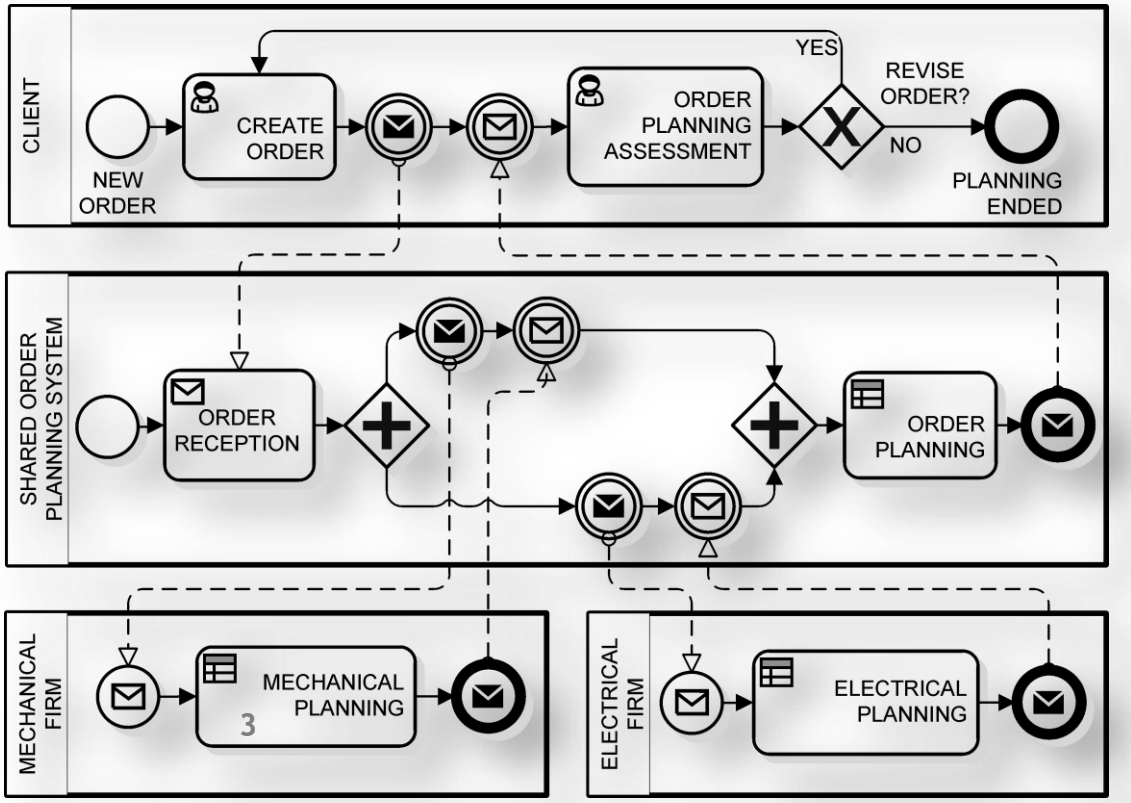
COORDINATION + ORDERS PLANNING TASK

An example of social enactment

A) BPMN process diagram of the collaborative planning of an order

A new order is created in a user task of the Client. A message with the order is sent from the client to the Shared Order Planning System. The Planning System splits the order into two parts, i.e. a mechanical and an electrical part, and sends them to the mechanical and electrical firms, respectively. Then, each firm performs its planning, represented as a business rule task. In a business rule task, one or more business rules are applied in order to produce a result or to make a decision, by means of a Business Rule Management System (BRMS) which is called by the process engine.

The BRMS then evaluates the rules that apply to the current situation. Each pool of a firm is supposed to be executed in a firm's private server, whereas the Planning System and the Client pools are supposed to be executed in a shared server. This way, the business rules of each firm are completely hidden to the Community.



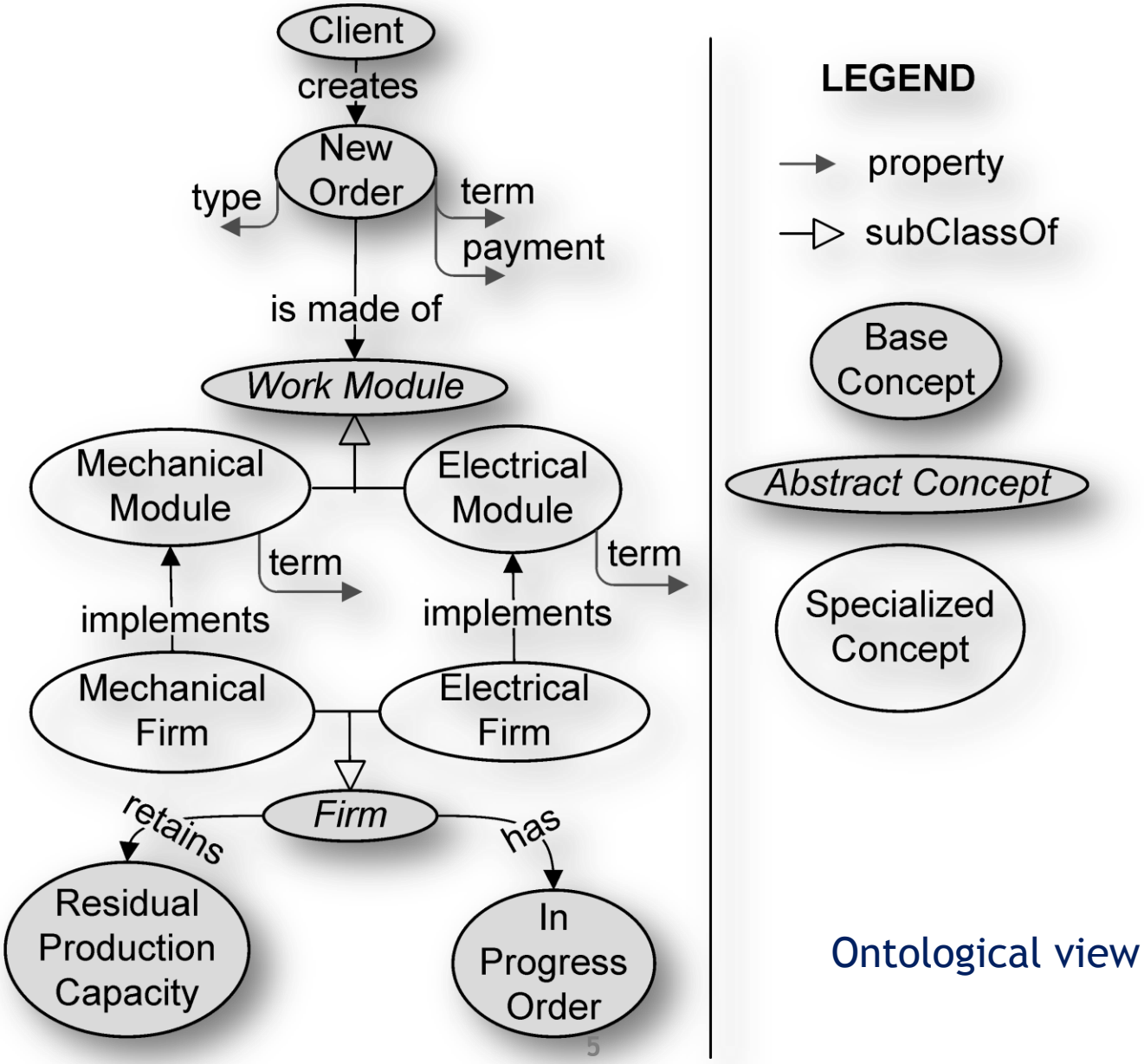
The decision of each firm is then sent to the Planning System, which carries out a logical combination via another business rule task, i.e., Order Planning, providing the Client with the overall planning of the order. Subsequently, the Client receives the planning and performs an assessment of it. The planning can either be revised, by creating a new order, or accepted, which causes the end of the workflow.

B) Business rules

An order type can be either *standard* or *innovative*, i.e., an order very similar or completely different with respect to the past orders, respectively. An order can be performed either in the *short* or in the *long* period, depending on the following of factors: the order type, the number of "in progress" orders, the payment time, and the residual production capacity. The coordination task consists in conducting an iterative communication between the client and the firms, whose result is the order's planning or its rejection.

An ontological view of the collaborative planning of an order is represented in the next slide, where base concepts, enclosed in gray ovals, are connected by properties, represented by black directed edges. More formally, a *Client creates a New Order*, which is characterized by a *type* (which can assume the value "standard" or "innovative"), a *term* (which can assume the value "short" or "long") and a *payment* (which can assume the value "fast" or "slow").

An example of social enactment



Ontological view

An example of social enactment

The new order *is made of Work Modules*. Work module is a generalized and abstract concept, i.e., it cannot be instantiated. In figure, the name of abstract concepts is represented with italic style. *Mechanical Module* and *Electrical Module* are work modules specialized from Work Module. In figure, specialized concepts are shown with white ovals and are connected by white directed edges to the generalized concept. Each module is characterized by a *term* (which can assume the value "short" or "long"), and *is implemented by* a *Mechanical* or *Electrical Firm*, respectively. Each firm inherits two properties from the generalized concept *Firm*. A firm *has* an *in progress order*s and *retains a Residual Production Capacity*. Both properties can assume the value "true" or "false".

For the sake of brevity, in the scenario the ontology is globally shared between participants and the business rules are different for each participant. However, the ontology can be also modularized, to avoid sharing private concepts.

C) Natural-language business rules

- a mechanical firm places a new order in the short term if its type is standard and there are no in-progress orders; otherwise the order is placed in the long term;
- an electrical firm places a new order in the short time if there is a residual production capacity and the payment is fast or if the payment is slow and its type is standard;
- the planning system places a new order in the short term only if both modules have been placed in the short term.

An example of social enactment

D) Formal IF-THEN rules

TASK: MECHANICAL PLANNING

RULE 1:
If *newOrder.type* **Is** standard
And *inProgressOrder* **Is** true
Then *mechanicalModule.term* **Is** long

RULE 2:
If *newOrder.type* **Is** innovative
Then *mechanicalModule.term* **Is** long

RULE 3:
If *newOrder.type* **Is** standard
And *inProgressOrder* **Is** false
Then *mechanicalModule.term* **Is** short

TASK: ELECRICAL PLANNING

RULE 1:
If *residualProductionCapacity* **Is** false
Then *electricalModule.term* **Is** long

RULE 2:
If *residualProductionCapacity* **Is** true
And *newOrder.payment* **Is** slow
And *newOrder.type* **Is** innovative
Then *electricalModule.term* **Is** long

RULE 3:
If *residualProductionCapacity* **Is** true
And *newOrder.payment* **Is** fast
Then *electricalModule.term* **Is** short

RULE 4:
If *residualProductionCapacity* **Is** true
And *newOrder.payment* **Is** slow
And *newOrder.type* **Is** standard
Then *electricalModule.term* **Is** short

TASK: ORDER PLANNING

RULE 1:
If *mechanicalModule.term* **Is** long
Then *newOrder.term* **Is** long

RULE 2:
If *electricalModule.term* **Is** long
Then *newOrder.term* **Is** long

RULE 3:
If *mechanicalModule.term* **Is** short
And *electricalModule.term* **Is** short
Then *newOrder.term* **Is** short

An example of social enactment

E) Collaborative Analytics

- Business rules are usually designed according to goals which are measurable via related Key Performance Indicators (KPIs), for each company and for the community itself.
- For this reason, the usability of the data flow connected to the workflow is a fundamental requirement.
- In a collaborative network the computation of KPIs must preserve the marketing value of data source to be aggregated, avoiding industrial espionage between competitors.
- The focus here is not on specific KPIs: the technique is suitable for any business measurements that need to be aggregated handling company's data.
- The problem in general can be brought back to comparing providers' performance. In practice, a collective comparison is related to the "*to share or not to share*" dilemma, an important reason for the failure of data sharing in collaborative networks.

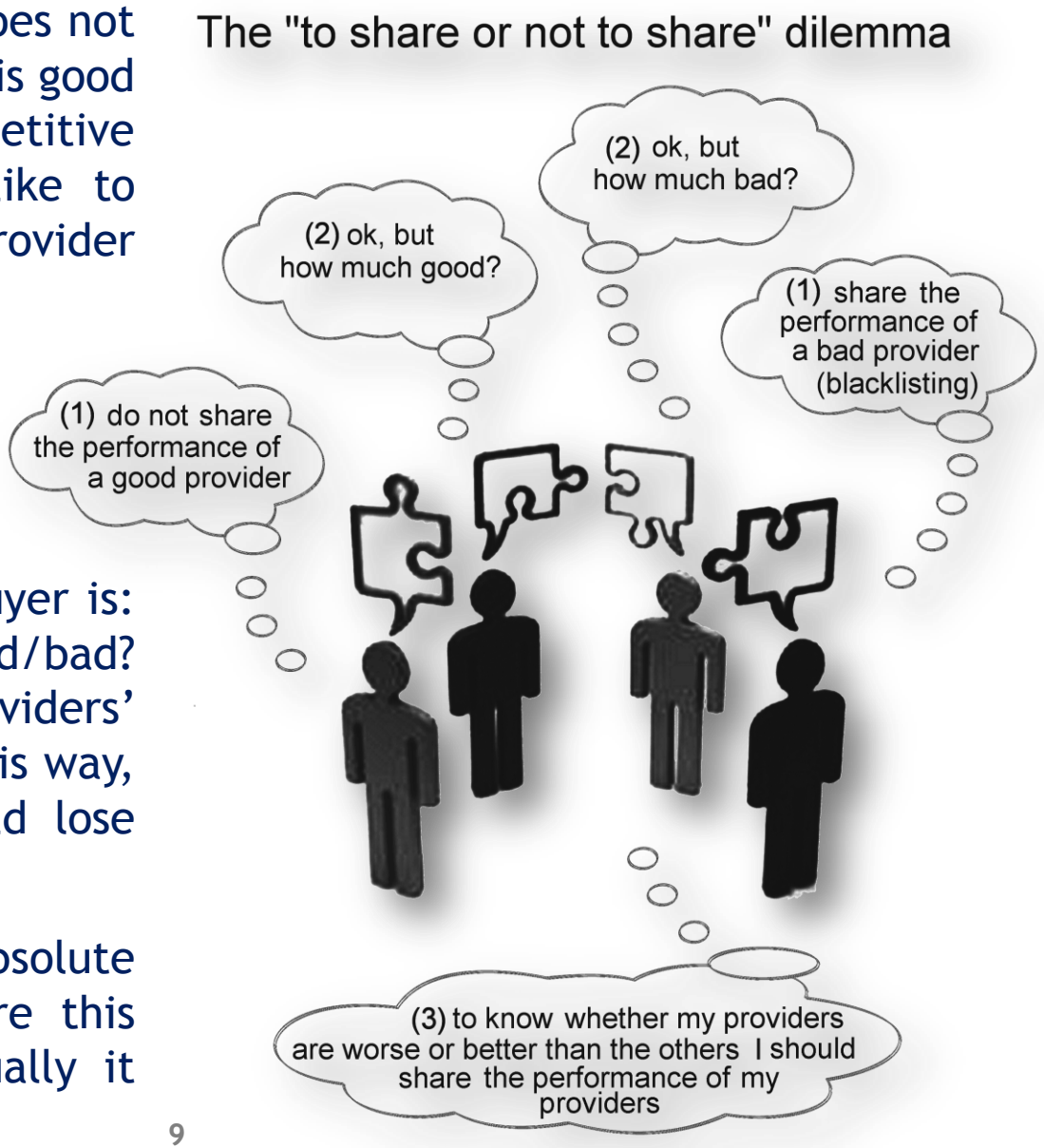
An example of social enactment

In the dilemma, a typical buyer does not like to share the performance of his good providers (keeping a competitive advantage over its rivals) and like to share the performance of a bad provider (showing his collaborative spirit).

However, each buyer knows a subset of the providers available on the market.

The fundamental question of a buyer is: how much are my providers good/bad? To solve this question, providers' performance should be shared. This way, buyers with good providers would lose the competitive advantage.

Given that nobody knows the absolute ranking of his providers, to share this knowledge is risky and then usually it does not happen.



An example of social enactment

Let us consider an extension of the pilot scenario, with a new behavior in the workflow: when the mechanical or the electrical planning does not satisfy the client requirements, the Planning System must be able to select an alternative partner.

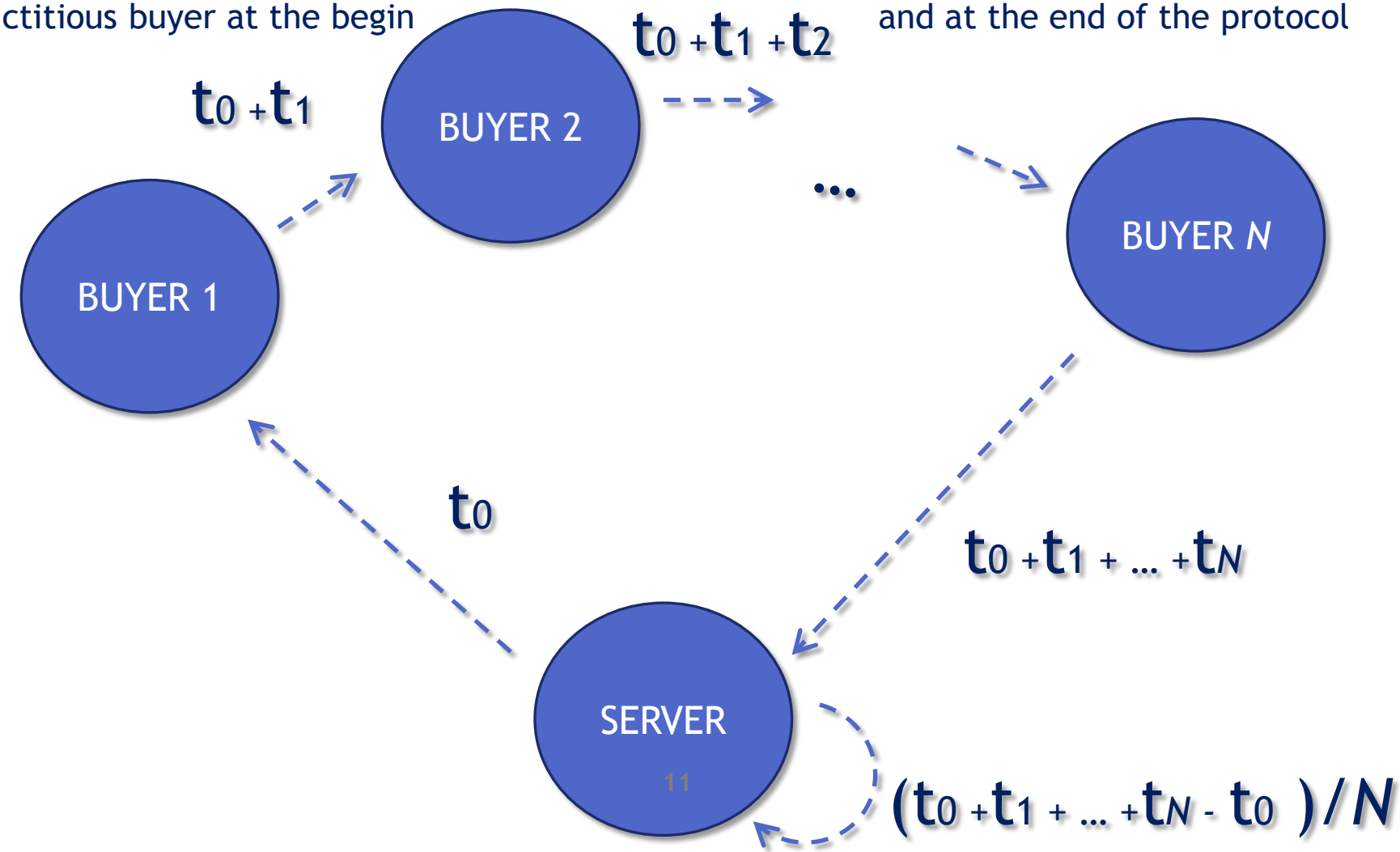
To achieve this extension, an *Order Planning Assessment* activity should be carried out by the Planning System too. Then, another activity, called *Select Alternative Partner*, should compare partners' performance to carry out a selection. Such performance must be made available by a collaborative analytics process.

The next slide shows an example of data flow designed to implement a privacy-preserving collaborative analytics process. The Collaborative Analytics System (called hereafter "System" for the sake of brevity) is the main pool located on a shared server and coordinating pools of registered buyers. Each buyer's pool is located on a private server.

The main goal of the data flow is to create a public collective data by aggregating buyers' private data. For instance, let us consider a community of N buyers B_1, B_2, … B_N, and a community of M vendors V_1, V_2, … V_M, each buyer being supplied by a small subset of the vendors. The average delivery time of the vendors of a buyer is an example of private datum, whereas the average delivery time of the vendors of all buyers is an example of collective datum.

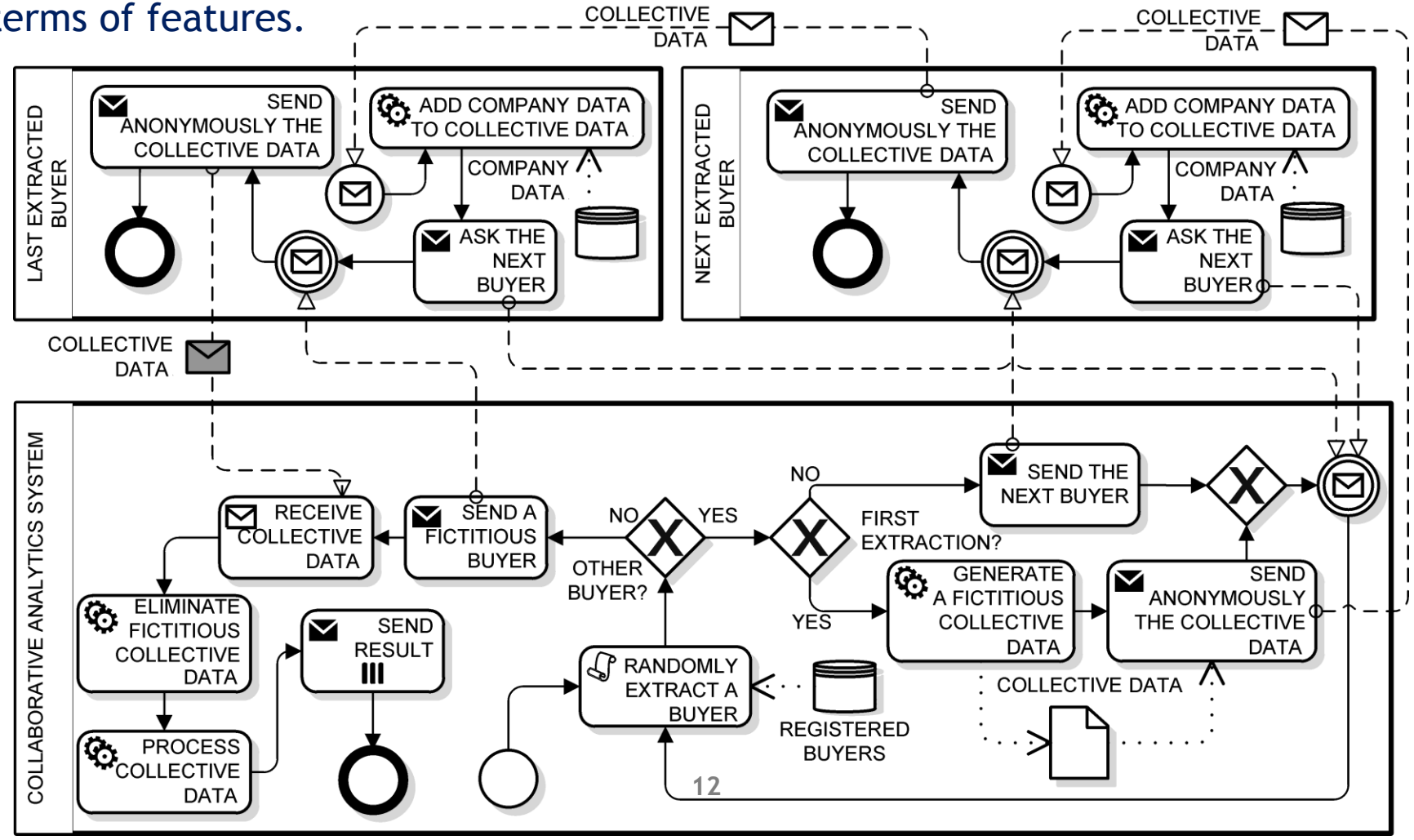
An example of social enactment

The problem: how to calculate the average without sending each term t_k to the server?
The solution: each buyer receives a partial summation, adds its own term and sends the next partial summation to the next buyer. The server orchestrates step-by-step a random sequence of buyers. At each step, the next buyer is asked to the server, which does not manage partial summations. The messaging is trusted but anonymous and the server can act as a fictitious buyer at the begin and at the end of the protocol



An example of social enactment

In general the aggregation process protects buyers' datum from being publicized. More specifically, at the beginning the System randomly extracts a buyer and generates a fictitious collective datum. A fictitious datum is an artificial creation that mimics real-world datum, and then cannot be distinguished from actual datum in terms of features.

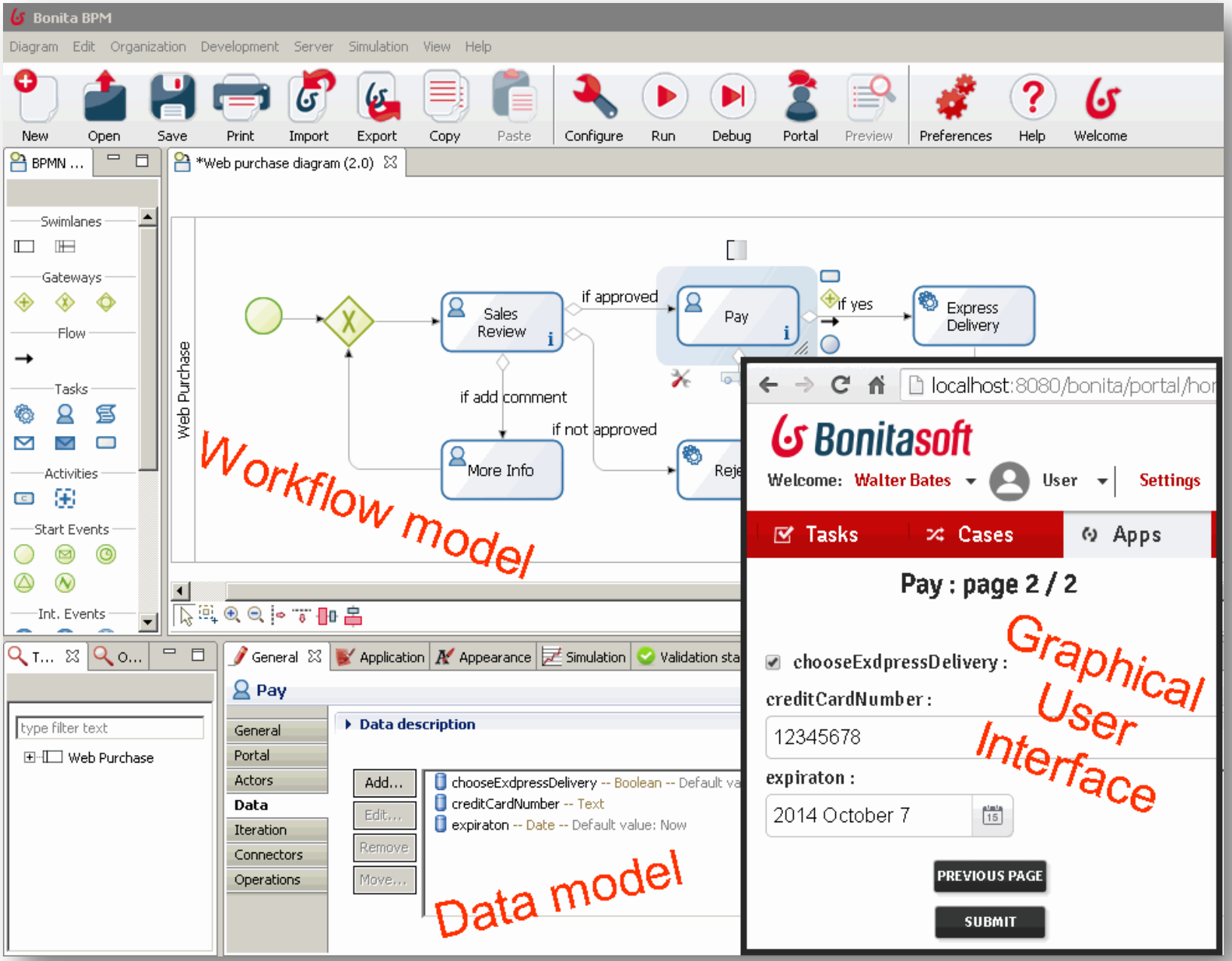


An example of social enactment

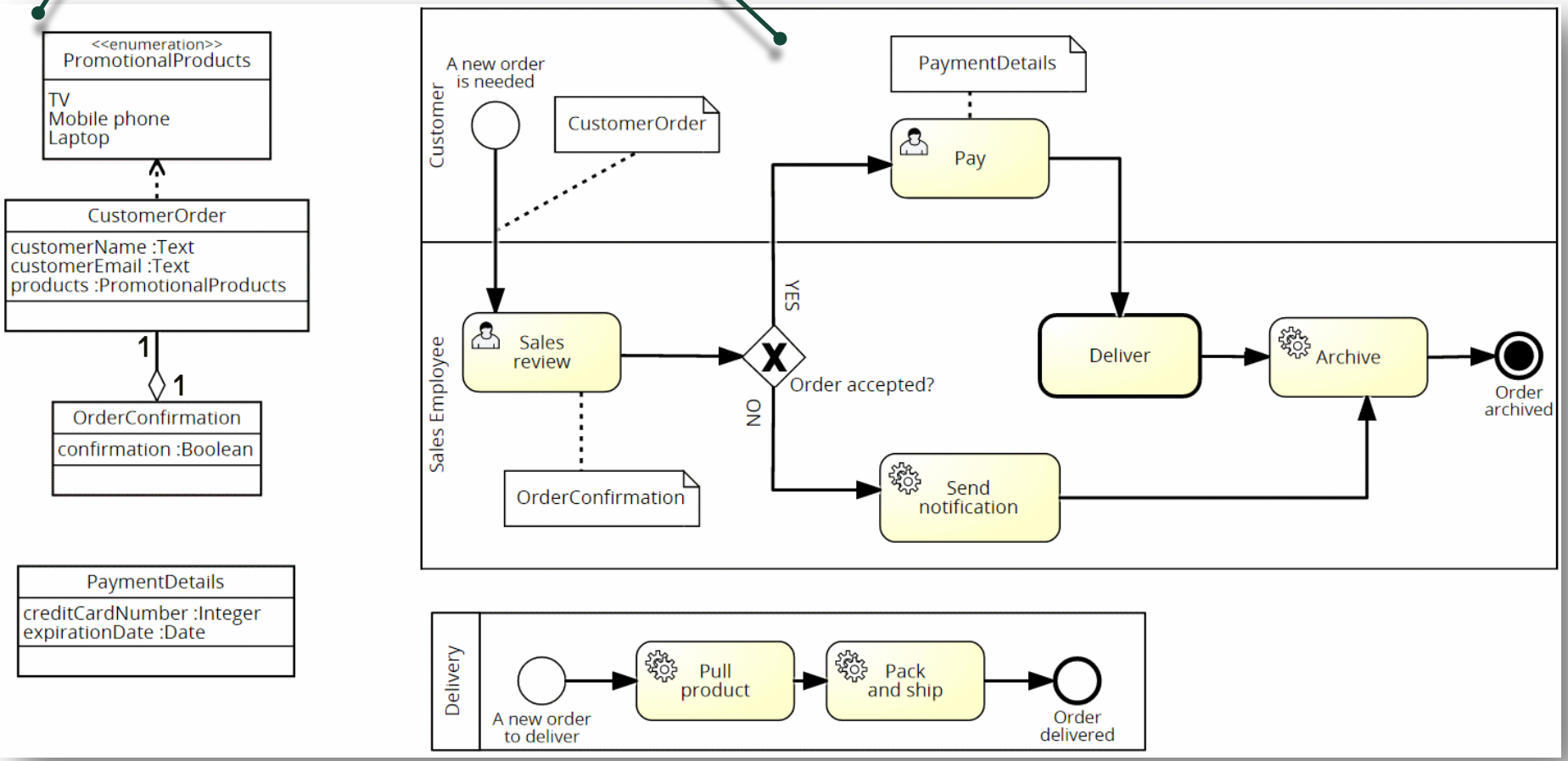
- Collective datum is then anonymously sent to the extracted buyer, who adds his private datum to it and ask the System for the next buyer.
- The system will answer with a randomly extract next buyer. Then, the buyer sends anonymously collective datum. This way, collective datum is incrementally built and transferred from a buyer to another one, under orchestration of the System.
- Each buyer is not aware of his position in the sequence. This is because the first extracted buyer receives a fictitious collective datum, and because the sender is always anonymous.
- The last extracted buyer will be provided with a fictitious buyer by the system. Such fictitious buyer actually corresponds to the System itself. After receiving the collective datum, the System subtracts the initial fictitious datum, thus obtaining the actual collective datum, which is then processed (so as to extract some common features) and sent to all buyers.
- By comparing the collective datum with his private datum, each buyer will be able to assess his position with respect to the collective performance. The results of this process can be used by to select a partner whose performance is higher than the collective performance.

- ✓ Bonita BPM 7 is a powerful application platform for building personalized, process-based business applications that adapt to your business changes in real time.
- ✓ Bonita BPM has two parts: the development environment, Bonita BPM Studio, and the runtime environment, Bonita BPM Platform.
- ✓ Bonita BPM adopts the **model-driven approach**, a software design methodology for the development of software systems, launched by the Object Management Group (OMG) in 2001.
- ✓ With model-driven engineering, specifications are expressed as models. Models can be expressed with standards, such as the executable Unified Modeling Language (UML), and the BPMN.
- ✓ Models are then processed to automatically generate software. Code generation means that an automated tool derives from the models parts or all of the source code for the software system.

BP Management: Web purchase example with Bonita BPM15 of 17



- ✓ Our first model, edited with Signavio
- ✓ **Class diagram** (data model)
- ✓ **Business Process diagram** (workflow model + data objects)



1. Download the Bonita BPMS from
   http://www.iet.unipi.it/m.cimino/wdis/res/
   “Process Management suite: Bonita BPM 7.x [local]”
   http://www.iet.unipi.it/m.cimino/wdis/res/BonitaBPMCommunity-7.1.2.zip

2. Extract it to c:\wdis

3. If needed, change the JDK (from MS-DOS console)
   > set JAVA_HOME=C:\wdis\jdk8
   > set PATH=C:\wdis\jdk8\bin;%PATH%
   > java -version (1.8)