Requirements Engineering

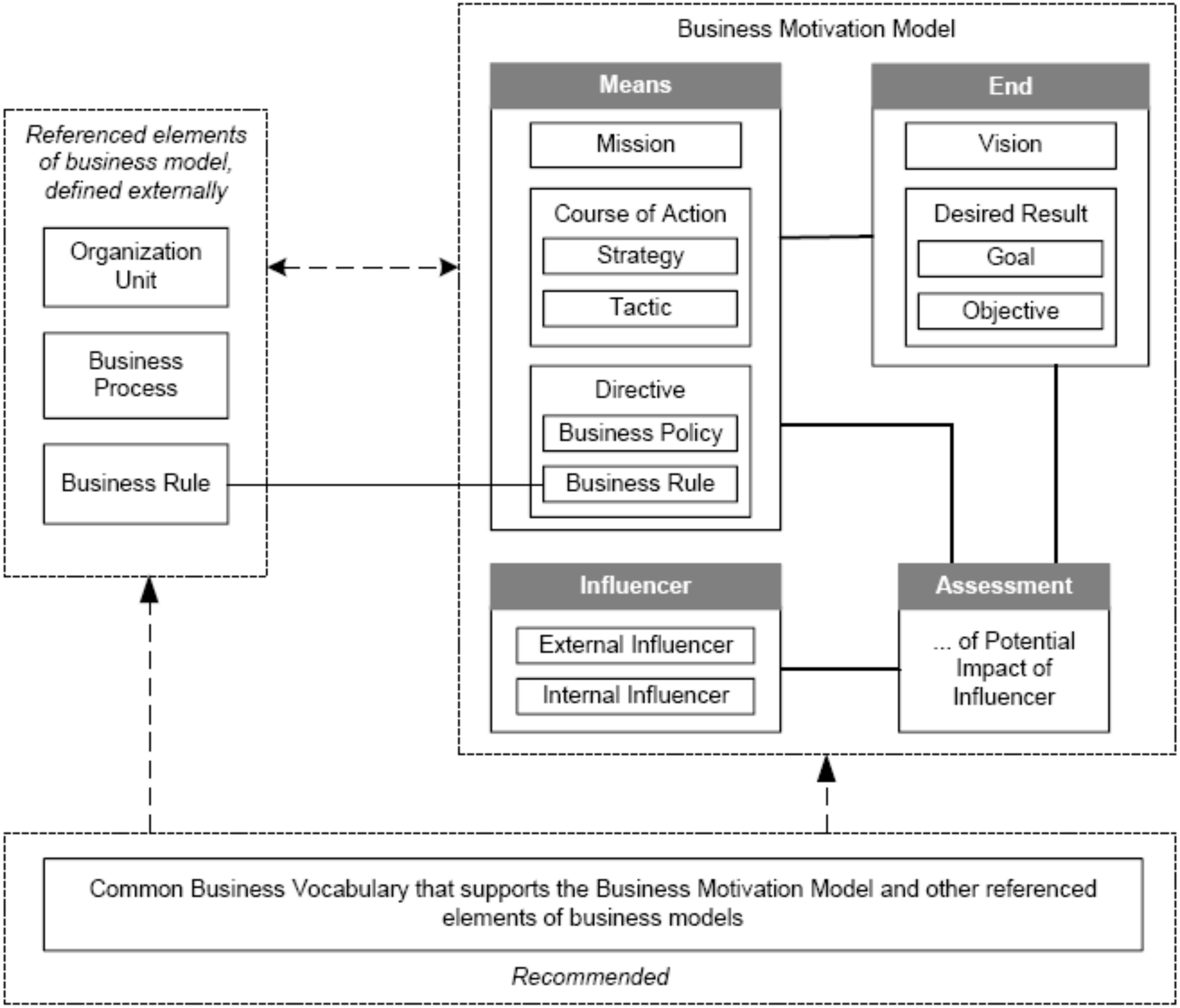
2018/2019
Course 9

Course 9 outline

- BMM to SOA
- Agile Methods and RE

Course 9 Bibliography

- Business Motivation Model (BMM), OMG, 2008. http://www.omg.org/spec/BMM/1.1/PDF
- Birol Berkem. From BMM to SOA, in Journal of Object Technology, vol. 7, no. 8, 2008, pp. 57-70. http://www.jot.fm/issues/issue_2008_11/column6/


Business Motivation Model
Means
Mission
Course of Action
Strategy
Tactic
Directive
Business Policy
Business Rule
End
Vision
Desired Result
Goal
Objective
Influencer
External Influencer
Internal Influencer
Assessment
... of Potential Impact of Influencer
Referenced elements of business model, defined externally
Organization Unit
Business Process
Business Rule
Common Business Vocabulary that supports the Business Motivation Model and other referenced elements of business models
Recommended

SOA Overview

- Service Oriented Architecture (SOA) is a way of describing and understanding organizations, communities, and systems to maximize agility, scale, and interoperability.
- The SOA approach: people, organizations, and systems provide services to each other.
- A service is value delivered to another through a well-defined interface and available to a community (which may be the general public). A service results in work provided to one by another.
- It is an architectural paradigm for defining how people, organizations, and systems provide and use services to achieve results.
- The SOA paradigm works equally well for integrating existing capabilities as well as creating and integrating new capabilities.

From BMM to SOA

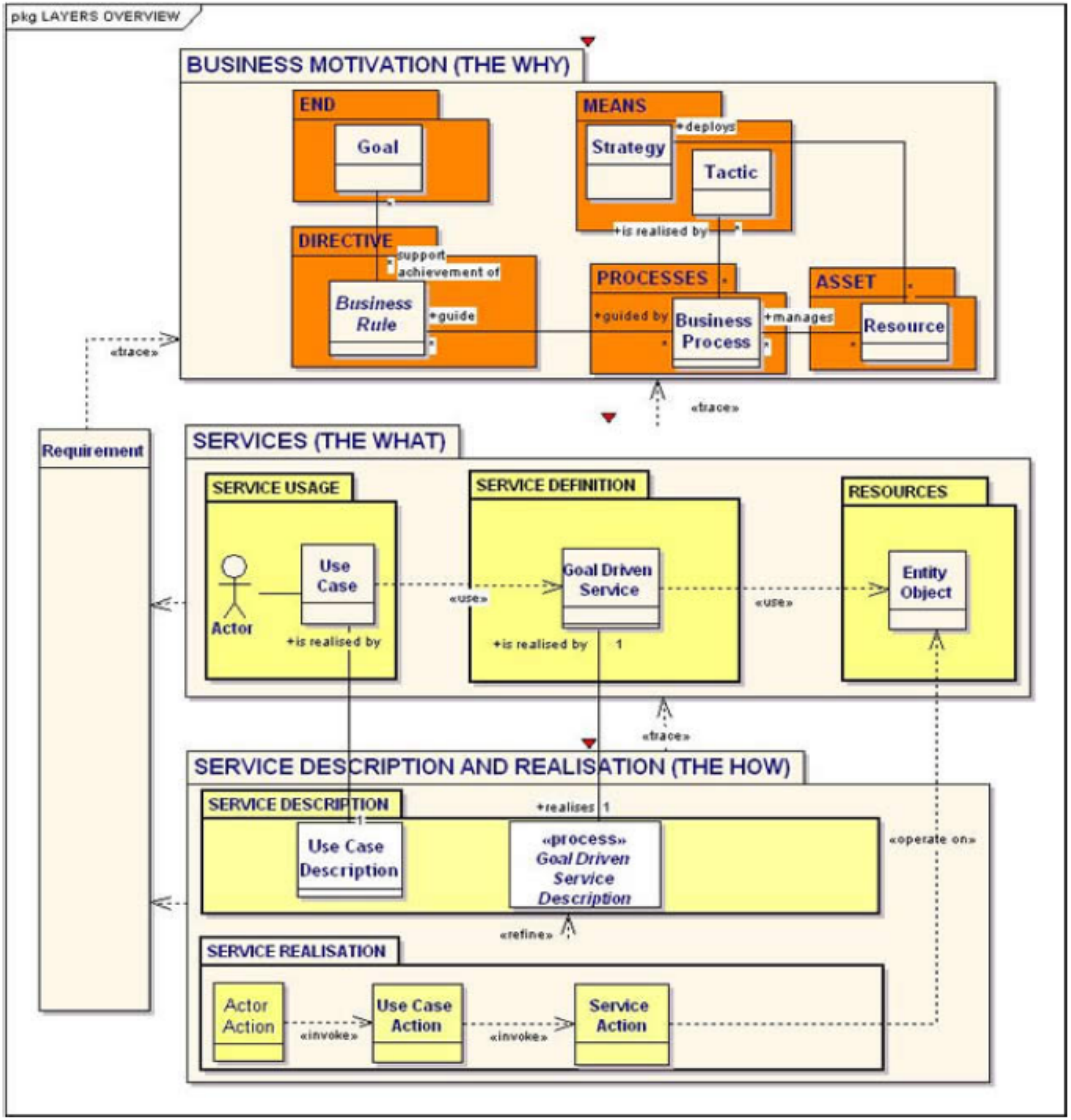
- Most SOA approaches try to determine SOA services on the basis of business processes being totally disconnected from business goals and rules.
- Disadvantage: the resulting system suffer from agility issues in aligning service descriptions to changing decisions of the management.
- The agile SOA architecture must integrate the core elements of the BMM by providing a capability to capture how services realise business motivations (vision, goals, objectives, missions, strategies, tactics, policies, regulations, etc...).
- The SOA development process should support at least the following trace-abilities :
  - IT services to achieve business processes,
  - Requirements that compose these services to the business rules,
  - Governance of these services to tactics that implement business strategies,
  - Use Cases to responsibilities assigned to participants of the business processes.

From BMM to SOA

- Synchronization of the SOA system with the business motivations requires the following abstraction layers:
  - The **WHY** : to reconfigure business processes and responsibilities of their participants according to changes captured at the business layer (changes on the means (tactics) or directives (rules) that support desired results).
  - The **WHAT** : to determine changes on the IT level system components (services that are derived from processes, use cases and entity objects) as well as on system requirements that implement business rules.
  - The **HOW** : to describe realization of these specifications in the architecture (from the presentation layer to the entity layer),
  - The **WHERE** : to describe location of these components starting by their high level descriptions like agencies, headquarters, front and back offices, etc., until physical nodes of the architecture where they should be deployed.

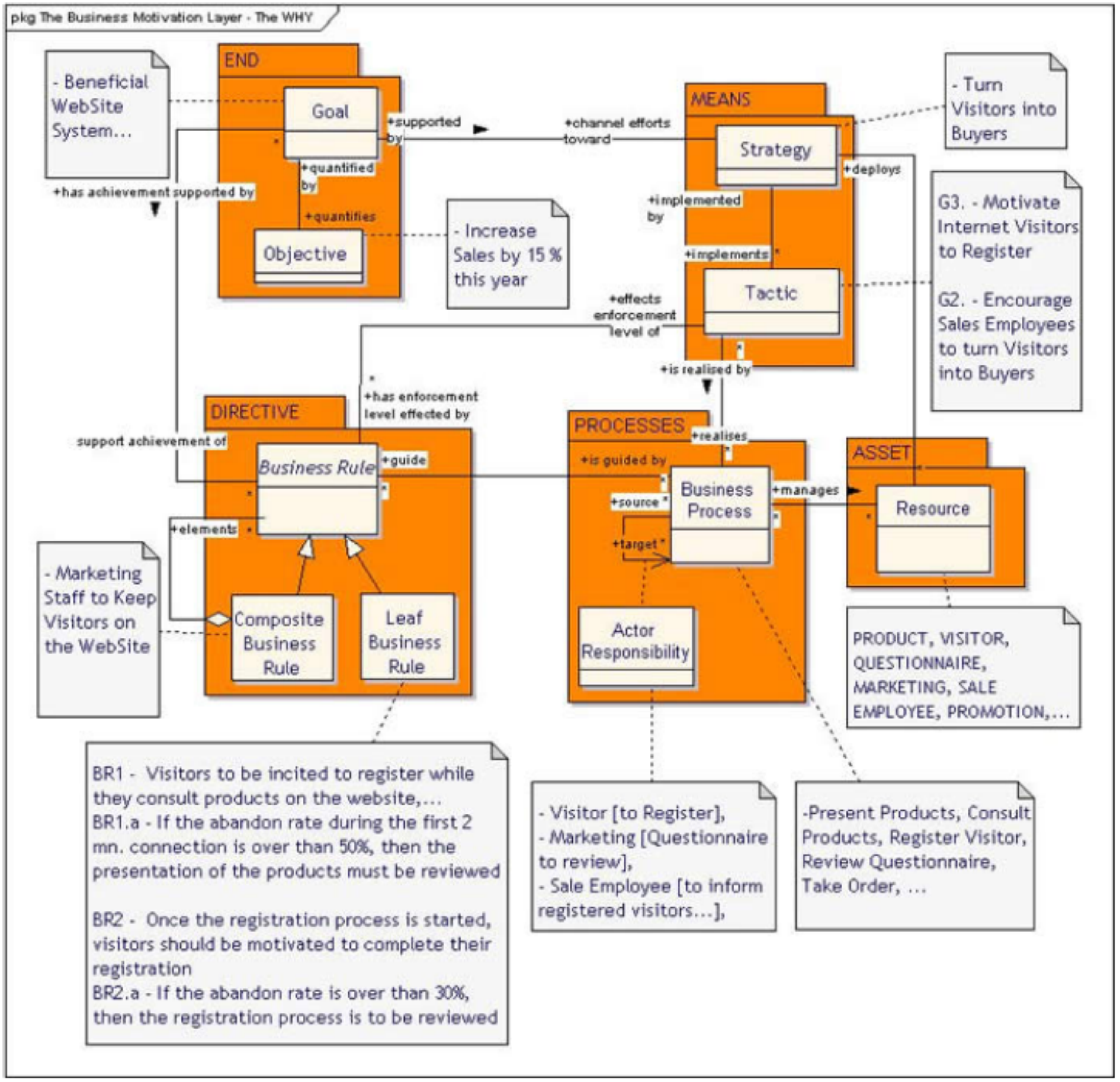
From BMM to SOA

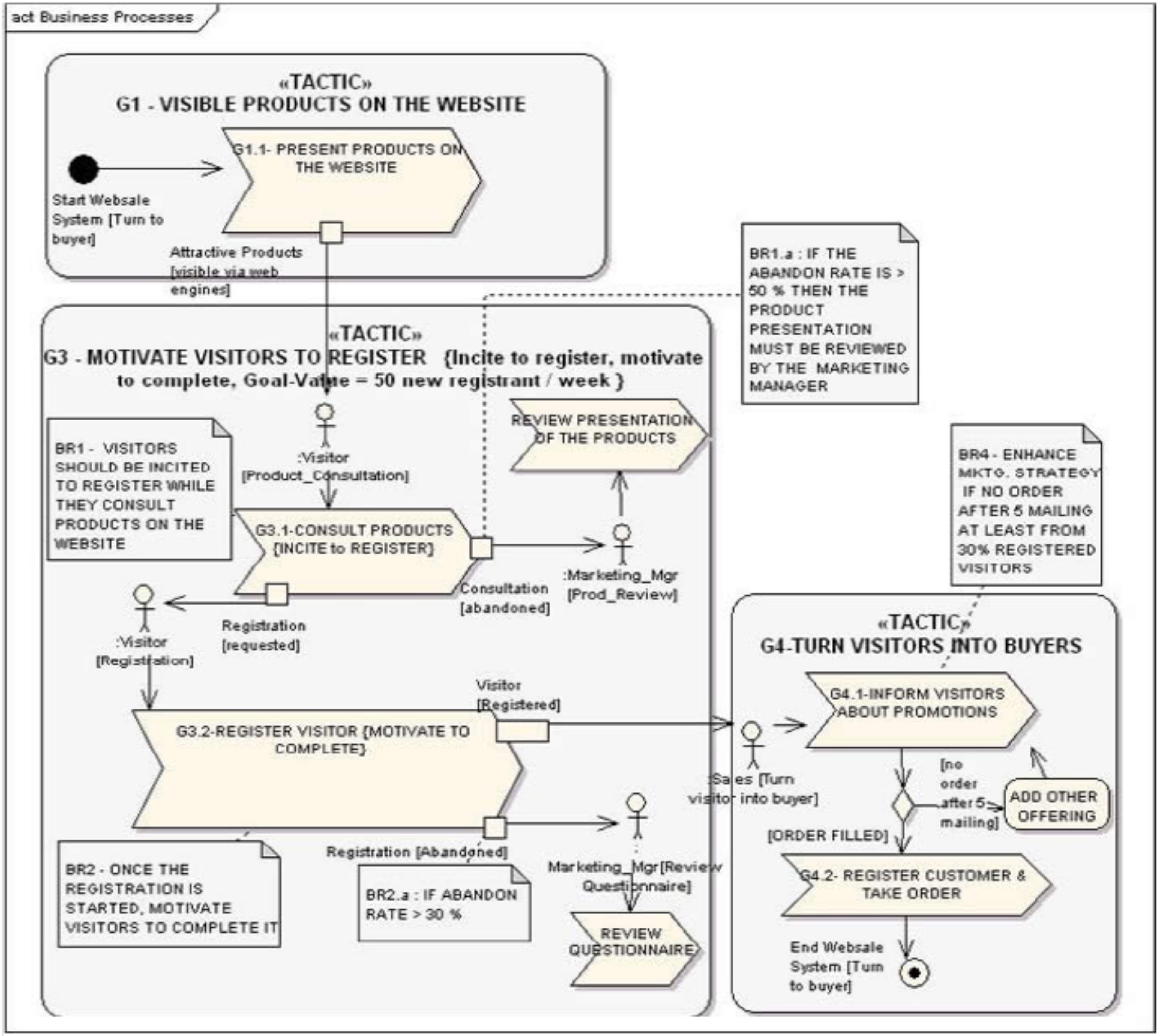
- In order to bridge business decisions toward IT system behaviors, the SOA framework focus especially on the Business Motivation (the "Why"), the Service Definition (the "What"), and on the Service Realization (the "How") layers.
- The **Business Motivation Layer**: the main focused elements are the Goal (End), Strategy, Tactic (Means), Business Rule (Directive), Business Processes and Resources.
- The **Service Definition Layer** focuses on discovering services of the system and use cases that are candidate to invoke them as well as resources that are to be handled by services at this abstraction level.
- The **Service Description and Realization Layer** focuses first on detailed specifications of services and use cases. Then it focuses on actions that realize behaviors of these components from the User Interface toward Entity Objects.
- Requirements that are discovered on the basis of the Business Motivation Layer are also used and enriched by the Service Definition and Realization layers.


pkg LAYERS OVERVIEW
BUSINESS MOTIVATION (THE WHY)
END
Goal
MEANS
Strategy
+deploys
Tactic
+is realised by
DIRECTIVE
support achievement of
Business Rule
+guide
PROCESSES
+guided by
Business Process
+manages
ASSET
Resource
«trace»
Requirement
SERVICES (THE WHAT)
SERVICE USAGE
Actor
Use Case
+is realised by
«use»
SERVICE DEFINITION
Goal Driven Service
+is realised by
«use»
RESOURCES
Entity Object
«trace»
SERVICE DESCRIPTION AND REALISATION (THE HOW)
SERVICE DESCRIPTION
Use Case Description
+realises
«process» Goal Driven Service Description
«operate on»
«refine»
SERVICE REALISATION
Actor Action
«invoke»
Use Case Action
«invoke»
Service Action

Example- BMM Layer

- Business Vision: “To be a profitable customer focused web-sale company“.
- Main Goal: “To build a Beneficial WebSite System“.
- A strategy: "Turning visitors into Buyers“.
- Two tactics (for strategy):
  - G3 - “Motivate internet visitors to Register using a Bonus System“
  - G2 - “Encourage Sales Employees to turn Visitors into Buyers“.
- Business rules:
  - "Marketing Staff to keep Visitors on the WebSite ...“ aggregates:
  - BR1. Incite Visitors to Register during their product consultation on the website
  - BR2. Motivate them to complete their registration process


pkg The Business Motivation Layer - The WHY
END
Goal
Objective
- Beneficial WebSite System...
+has achievement supported by
+supported by
+quantified by
+quantifies
- Increase Sales by 15 % this year
MEANS
Strategy
Tactic
+channel efforts toward
+deploys
+implemented by
+implements
+effects enforcement level of
+is realised by
- Turn Visitors into Buyers
G3. - Motivate Internet Visitors to Register
G2. - Encourage Sales Employees to turn Visitors into Buyers
DIRECTIVE
Business Rule
Composite Business Rule
Leaf Business Rule
support achievement of
+has enforcement level effected by
+guide
+elements
- Marketing Staff to Keep Visitors on the WebSite
PROCESSES
Business Process
Actor Responsibility
+realises
+is guided by
+source
+target
+manages
ASSET
Resource
PRODUCT, VISITOR, QUESTIONNAIRE, MARKETING, SALE EMPLOYEE, PROMOTION,...
BR1 - Visitors to be incited to register while they consult products on the website,...
BR1.a - If the abandon rate during the first 2 mn. connection is over than 50%, then the presentation of the products must be reviewed
BR2 - Once the registration process is started, visitors should be motivated to complete their registration
BR2.a - If the abandon rate is over than 30%, then the registration process is to be reviewed
- Visitor [to Register],
- Marketing [Questionnaire to review],
- Sale Employee [to inform registered visitors...],
-Present Products, Consult Products, Register Visitor, Review Questionnaire, Take Order, ...

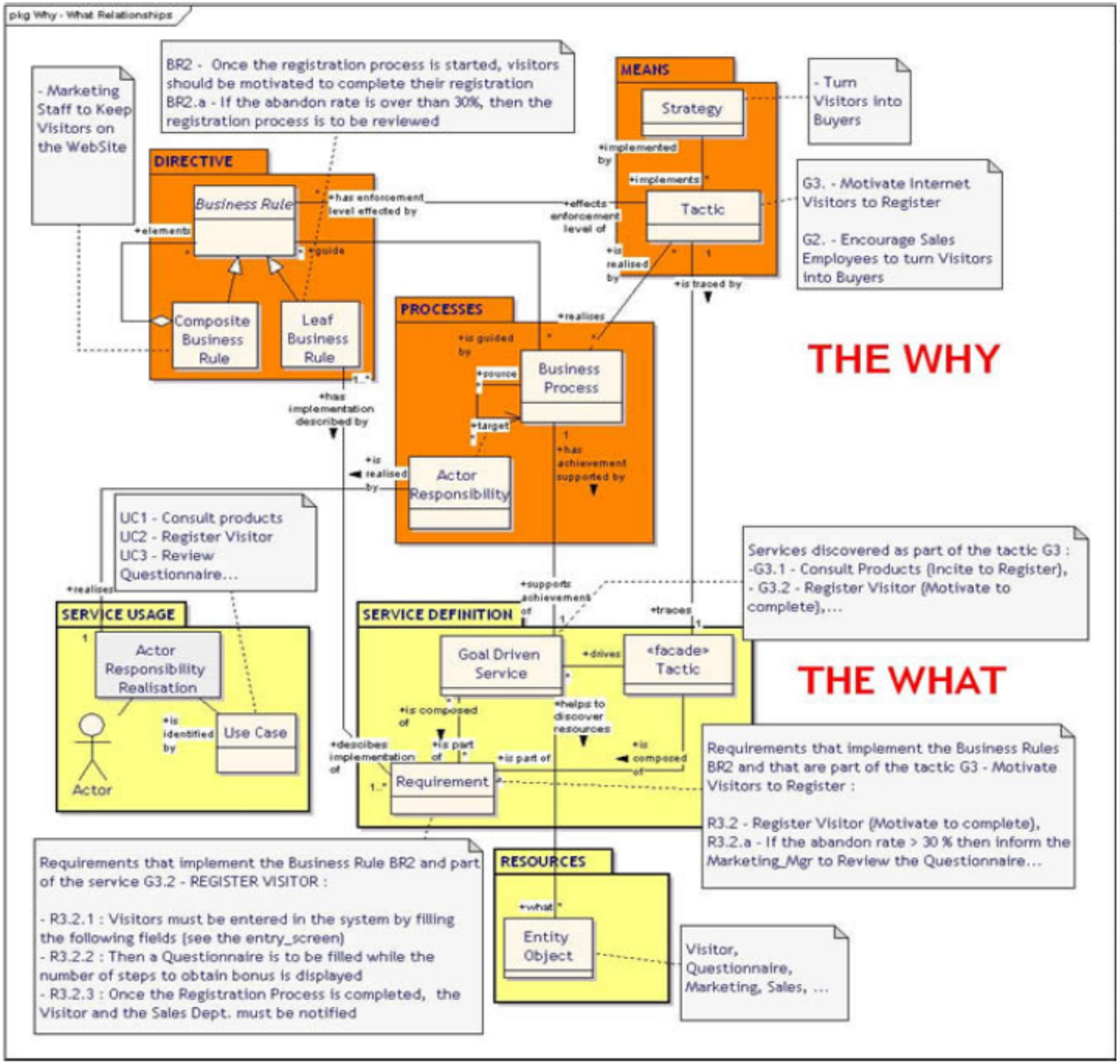

act Business Processes
«TACTIC»
G1 - VISIBLE PRODUCTS ON THE WEBSITE
G1.1- PRESENT PRODUCTS ON THE WEBSITE
Start Websale System [Turn to buyer]
Attractive Products [visible via web engines]
«TACTIC»
G3 - MOTIVATE VISITORS TO REGISTER {Incite to register, motivate to complete, Goal-Value = 50 new registrant / week }
BR1 - VISITORS SHOULD BE INCITED TO REGISTER WHILE THEY CONSULT PRODUCTS ON THE WEBSITE
:Visitor [Product_Consultation]
G3.1-CONSULT PRODUCTS {INCITE to REGISTER}
Consultation [abandoned]
:Marketing_Mgr [Prod_Review]
REVIEW PRESENTATION OF THE PRODUCTS
BR1.a : IF THE ABANDON RATE IS > 50 % THEN THE PRODUCT PRESENTATION MUST BE REVIEWED BY THE MARKETING MANAGER
:Visitor [Registration]
Registration [requested]
G3.2-REGISTER VISITOR {MOTIVATE TO COMPLETE}
Visitor [Registered]
Registration [Abandoned]
Marketing_Mgr[Review Questionnaire]
REVIEW QUESTIONNAIRE
BR2 - ONCE THE REGISTRATION IS STARTED, MOTIVATE VISITORS TO COMPLETE IT
BR2.a : IF ABANDON RATE > 30 %
BR4 - ENHANCE MKTG. STRATEGY IF NO ORDER AFTER 5 MAILING AT LEAST FROM 30% REGISTERED VISITORS
«TACTIC»
G4-TURN VISITORS INTO BUYERS
:Sales [Turn visitor into buyer]
G4.1-INFORM VISITORS ABOUT PROMOTIONS
[no order after 5 mailing]
ADD OTHER OFFERING
[ORDER FILLED]
G4.2- REGISTER CUSTOMER & TAKE ORDER
End Websale System [Turn to buyer]

From "Why" to "What"

- Tactics and business rules determined in the Business Motivation Layer are traced toward the Service Layer to drive services and to derive Requirements that compose them.
- A tactic acts as a facade to "drive" a set of goal-driven services and is composed of requirements derived from a related business rule.
- Eg. For tactic G3 - Motivate visitors to Register, two goal-driven services G3.1 Incite visitors to register during product consultation and G3.2 Motivate to complete his registration once started.
- A **goal-driven service** is defined for each business process of the business motivation layer and is composed of a cohesive set of requirements that implement business rules that guide this process.
- Grouping requirements first on the basis of tactics then by services help to design stable and loosely coupled services in order to increase their adaptation to changes and better control their evolution on the basis of changing tactics.

From "Why" to "What"

- Goal-Driven Services also permit to identify use cases using a one-to-one traceability relationship as well as entity objects they have to handle on the basis of requirements that compose them.
- Eg the service G3.2- Register Visitor permits to identify the use case Register Visitor and help to discover entity objects Visitor, Marketing, Sales,... on the basis of requirements that are part of this service.



BR2 - Once the registration process is started, visitors should be motivated to complete their registration
BR2.a - If the abandon rate is over than 30%, then the registration process is to be reviewed

- Marketing Staff to Keep Visitors on the WebSite

- Turn Visitors into Buyers

G3. - Motivate Internet Visitors to Register

G2. - Encourage Sales Employees to turn Visitors into Buyers

UC1 - Consult products
UC2 - Register Visitor
UC3 - Review Questionnaire...

Services discovered as part of the tactic G3 :
-G3.1 - Consult Products (Incite to Register),
- G3.2 - Register Visitor (Motivate to complete),...

Requirements that implement the Business Rules BR2 and that are part of the tactic G3 - Motivate Visitors to Register :

R3.2 - Register Visitor (Motivate to complete),
R3.2.a - If the abandon rate > 30 % then inform the Marketing_Mgr to Review the Questionnaire...

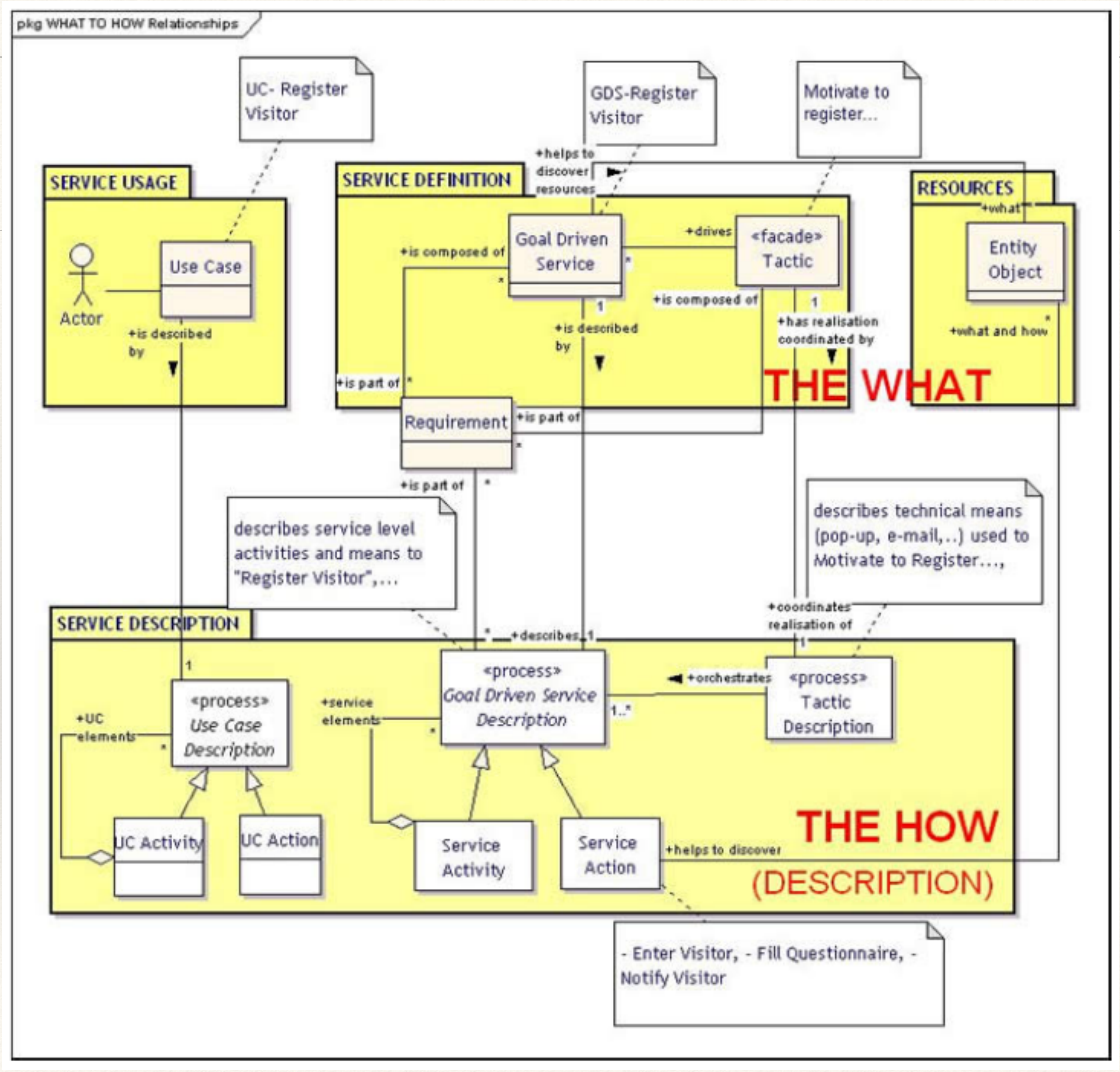
Requirements that implement the Business Rule BR2 and part of the service G3.2 - REGISTER VISITOR :

- R3.2.1 : Visitors must be entered in the system by filling the following fields (see the entry_screen)
- R3.2.2 : Then a Questionnaire is to be filled while the number of steps to obtain bonus is displayed
- R3.2.3 : Once the Registration Process is completed, the Visitor and the Sales Dept. must be notified

Visitor, Questionnaire, Marketing, Sales, ...

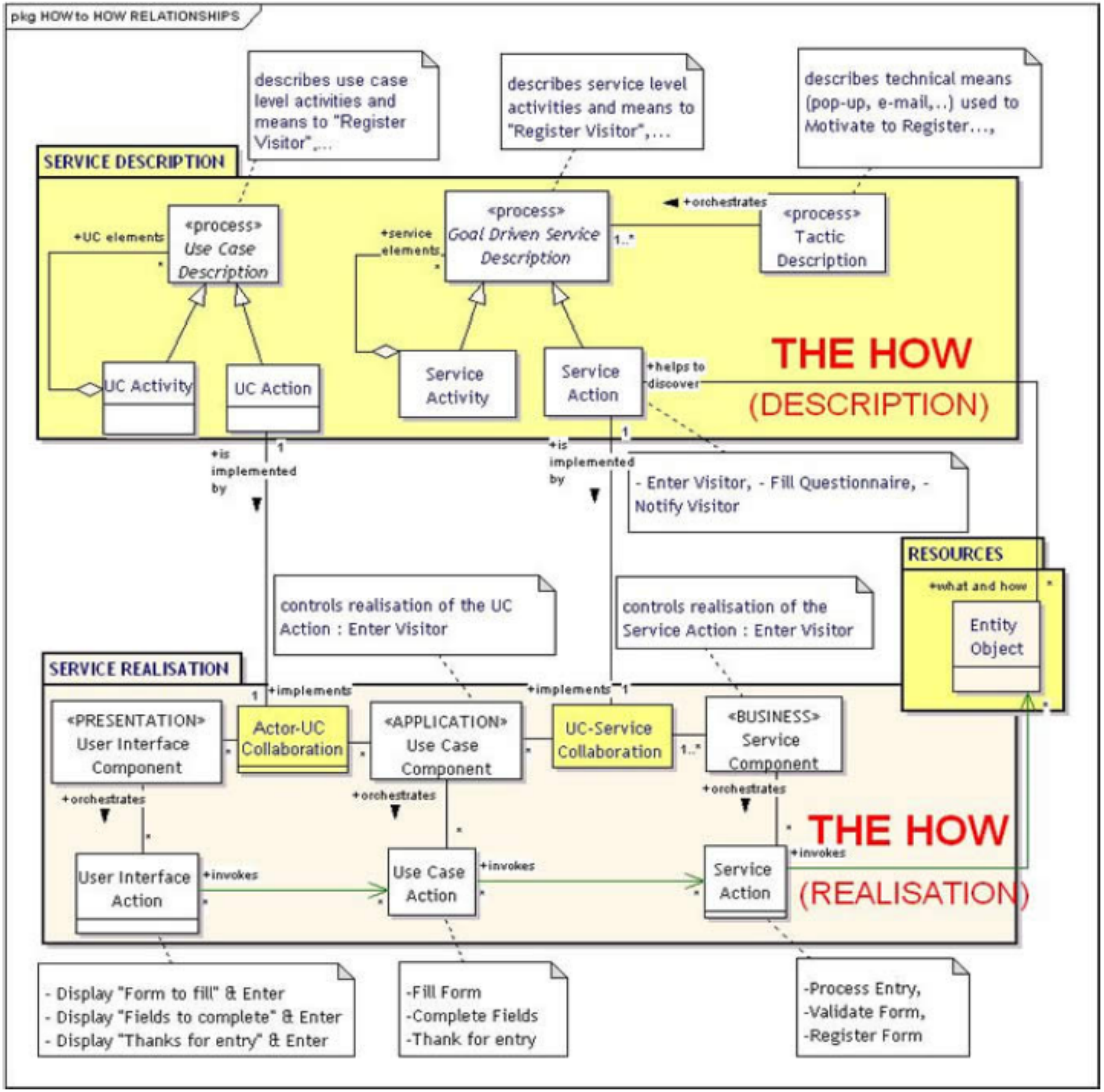
From "What" To "How"

- The main role of the service description and realization layer is to give a description of appropriate goal-driven services and use cases identified at the previous layer then explain what means to use to realize them.
- Tactics, Services and Use Cases that are determined at the Service Definition Layer (The What) are traced and described at the Service Description Layer (the How) respectively by Tactic, Service and Use Case Descriptions.
- A goal-driven service description is a process that can be materialized by an activity (a composite structure) or an action.
- An activity contains underlying activities or actions.
- Eg. the service Register Visitor may be described at this layer using actions Enter Visitor, Fill Questionnaire and Notify Visitor...


pkg WHAT TO HOW Relationships
UC- Register Visitor
GDS-Register Visitor
Motivate to register...
SERVICE USAGE
Actor
Use Case
+is described by
SERVICE DEFINITION
+helps to discover resources
Goal Driven Service
+drives
«facade» Tactic
+is composed of
+is described by
+has realisation coordinated by
+is part of
Requirement
RESOURCES
+what
Entity Object
+what and how
THE WHAT
describes service level activities and means to "Register Visitor",...
describes technical means (pop-up, e-mail,..) used to Motivate to Register...,
+coordinates realisation of
SERVICE DESCRIPTION
+describes
«process» Use Case Description
+UC elements
«process» Goal Driven Service Description
+service elements
+orchestrates
«process» Tactic Description
UC Activity
UC Action
Service Activity
Service Action
+helps to discover
THE HOW
(DESCRIPTION)
- Enter Visitor, - Fill Questionnaire, - Notify Visitor

“How" To "How"

- The purpose of the realization sub-layer is to describe how to achieve service actions and use case actions of the Service Description Layer using components of a logical IT architecture (Presentation, Application, Business Service and Domain Object Layers).
- Actions of the service description layer are described using collaborations between these components that form their underlying actions.


pkg HOW to HOW RELATIONSHIPS
describes use case level activities and means to "Register Visitor",...
describes service level activities and means to "Register Visitor",...
describes technical means (pop-up, e-mail,..) used to Motivate to Register...,
SERVICE DESCRIPTION
«process» Use Case Description
+UC elements
UC Activity
UC Action
«process» Goal Driven Service Description
+service elements
Service Activity
Service Action
+orchestrates
«process» Tactic Description
1..*
+helps to discover
THE HOW
(DESCRIPTION)
+is implemented by
- Enter Visitor, - Fill Questionnaire, - Notify Visitor
RESOURCES
+what and how
Entity Object
controls realisation of the UC Action : Enter Visitor
controls realisation of the Service Action : Enter Visitor
SERVICE REALISATION
+implements
«PRESENTATION» User Interface Component
Actor-UC Collaboration
«APPLICATION» Use Case Component
UC-Service Collaboration
«BUSINESS» Service Component
+orchestrates
User Interface Action
Use Case Action
Service Action
+invokes
THE HOW
(REALISATION)
- Display "Form to fill" & Enter
- Display "Fields to complete" & Enter
- Display "Thanks for entry" & Enter
-Fill Form
-Complete Fields
-Thank for entry
-Process Entry,
-Validate Form,
-Register Form

From "How" To "Where"

- Components that are described in the previous layer are affected to organizational locations.
- Eg., components related to the presentation and use case layers may be assigned to agencies in front and back-offices whereas service components and entities may be hosted in the headquarter area.

Agile Methods

- Agile Methods (AMs) aim to deliver products faster, with high quality, and satisfy customer needs through the application of the principles of lean production to software development.
- The principle of lean production is the constant identification and removal of waste, that is, anything that does not add value for the customer to the final product.
- AMs focus on:
  - Delivering value for the customer
  - Ensuring that the customer understands such value and be satisfied by the project
- To ensure customer satisfaction, a close collaboration between the development team and the customer is sought, so that:
  - Requirements are fully identified and correctly understood
  - Final products reflect what the customer needs, no more and no less
- Requirement engineering is very important for AMs.

The Agile Manifesto

- Contains a set of values common across all AMs:
  - **Individuals and interactions over process and tools**: The Agile approach emphasizes the importance of people and their interactions rather than focusing on structured processes and tools.
  - **Customer collaboration over contracts**: The relationship between the development team and the customer is regulated through the involvement of the customer in the development process rather than through detailed and fixed contracts.
  - **Working software over documentation**: The goal of the development team is delivering working code, which is the artifact that provides value to the customer. Well-written code is self-documented and formal documentation is reduced to the minimum.
  - **Responding to change over planning**: The development team has to react quickly to requirements variability. Binding decisions affecting this ability are delayed as long as possible and the time spent in the planning activity is limited to what the customer needs. Any attempts to forecast future needs are forbidden.

Agile Methods

- Common practices:
  - **Adaptability**: Practices have to be adapted to the specific needs of both the development team and the customer. There is no one size fits all solution.
  - **Incremental development**: The different phases of software development (analysis, design, code, and testing) are compressed in very short iterations (from 2 weeks to 2 months) in order to focus on a few, well-defined problems that provide real value to the customer.
  - **Frequent releases**: At the end of every iteration, the application is released to the customer that tests it and provides feedback. This approach produces several benefits such as:
    - the customer can use the application very early, allowing the identification of potential problems in time for improving the product, limiting the effect on the schedule;
    - the customer feels in control of the development process, since progresses are always visible;
    - the trust between the customer and the development team increases, since the team is considered reliable because is able to deliver working versions of the application early.

Agile Methods

- Common practices:
  - **Requirements prioritization before every iteration**: Before every iteration, the customer and the development team identify new requirements and reassign priorities to the old ones on the base of the customer actual needs.
  - **High customer involvement**: The customer is involved in the development process through a continuous request of feedback in order to identify potential problems early in the development. In some cases, the customer is even a member of the development team (customer on site practice) and is always available to interact with the team and clarify requirements-related issues.

Traditional RE

- Five of the eight main factors for project failure deal with requirements:
  - incomplete requirements, low customer involvement, unrealistic expectations, changes in the requirements and useless requirements.
- Traditional development processes have elaborated several standards:
  - IEEE Standard 830: Recommended Practice for Software Requirements Specifications
  - IEEE Standard 1233: Guide for Developing System Requirements Specifications
  - IEEE Standard 1362: Guide for Information Technology – System Definition – Concept of Operations Document
- AMs do not rely on these standards for requirements elicitation and management but they have adapted many of the basic ideas to the new environment:
  - The whole development team is involved in requirements elicitation and management, while in traditional approaches often only a subset of the development team is involved.

Agile Methods and RE

- Agile practices also deal with requirements in order to implement them correctly and satisfy the needs of the customer.
- These practices focus on a continuous interaction with the customer to address the requirements evolution over time, prioritize them, and deliver the most valuable functionalities first.

Agile Approaches to RE

- AMs include practices focused on the key factors to reduce the risk of failure.
- The aim of incremental development, frequent releases, requirements prioritization before every iteration, and customer involvement is to address the main risk factors.
- The Customer:
  - The customer assumes a very important role.
  - In AMs, the problem of multiple stakeholders is solved reducing their number to one, a single person that represents all the stakeholders involved in the project.
  - This customer should be a domain expert and able to make important decisions such as accepting the product, prioritize requirements, etc.
  - This approach is feasible only if the size of the problem is limited and a single person can act as customer, representing all the stakeholders.

Agile Approaches to RE

- The Customer:
  - In some AMs, the customer on site practice is common.
  - The customer is a member of the development team, is co-located with the team, and is always available to discuss issues related to the project with any team member.
  - The customer-on-site practice defines some specific requirements for the customer: availability, complete knowledge, decision power.
  - It is difficult to find a person satisfying all these requirements, since he has to be a very valuable member of the staff.
  - The availability of this kind of customer is very important for AMs, since most of their benefits (e.g., reduction of documentation, incremental delivery, etc.) are tightly coupled with the customer involvement.

Agile Approaches to RE

- Waste in Requirements:
  - AMs focus on the identification and reduction of waste in the development process. In particular, identifying and reducing the waste from requirements assume a paramount role to avoid the creation of waste later in the process.
  - Requirements engineering in AMs focuses on:
    - Reduction of waste from requirements
    - Managing the requirements evolution
  - The main effects of waste in requirements include:
    - More source code to write and higher cost
    - Increased complexity of the source code
    - Delayed delivery of the final version of the application with all functionalities
    - More complex and costly maintenance

Agile Approaches to RE

- **Waste in Requirements**:
  - Waste in requirements includes both wrong and useless requirements.
  - A misunderstanding between the customer and the development team causes wrong requirements. In order to reduce the probability of such misunderstanding, AMs adopt several techniques focused on the interaction between the customer and the development team:
  - *The whole development team collects requirements from the customer*: Requirements elicitation is an activity in which the whole team is involved. In this way, the usage of documents to share the knowledge is reduced to a minimum and the probability of misunderstandings decreases.
  - *Requirements are collected using a common language*: Requirements are collected using the language of the customer, not a formal language for requirements specification. This means that developers have to be introduced to the domain of the customer in order to understand him/her.

Agile Approaches to RE

- **Waste in Requirements**:
  - *Direct interaction between the development team and the customer*: There are no intermediaries between the development team and the customer. This approach reduces both the number of documents required and the probability of misunderstanding due to unnecessary communication layers.
  - *Requirements splitting*: If the development team considers a requirement too complex, this technique helps the customer to split it in simpler ones. This splitting helps developers to understand better the functionalities requested by the customer.

Agile Approaches to RE

- In order to reduce useless requirements, AMs use the following technique:
  - **Requirements Prioritization**: The customer and the development team assign priorities to each requirement in order to identify more important features that have to be implemented first.
  - **Incremental Releases**: Functionalities are released in small but frequent bunches (from 2 weeks to 2 months), in order to collect feedback from the customer.
  - The prioritization activity is performed in four steps:
    - The development team estimates the time required to implement each functionality. If the effort required is too high, the requirement is split into simpler ones that can be implemented with less effort.
    - The customer specifies business priorities for each functionality.
    - According to the business priorities, the development team assign a risk factor to the functionalities.
    - The customer and the development team identify the functionalities to implement in the iteration.
  - Requirements elicitation and prioritization are repeated at the beginning of every iteration.

Agile Approaches to RE

- **Requirements Evolution:**
  - AMs assume that it is very hard to elicit all the requirements from the user upfront, at the beginning of a development project.
  - They also assume that such requirements evolve in time as the customer may change its mind or the overall technical and socio-economical environment may evolve.
  - Agile companies are aware that changes are inevitable and they include the management of variability into the development process.
  - AMs base the requirements collection and management on three main hypotheses:
    - Requirements are not well known at the beginning of the project
    - Requirements change
    - Making changes is not expensive

Agile Approaches to RE

- **Requirements Evolution**:
  - In order to manage requirements evolution, AMs use variable scope-variable price contracts.
  - Requirements are not specified in details at contract level but defined step by step during the project through a negotiation process between the customer and the development team.
  - AMs approach requirements change in two ways:
    - **Decoupling Requirements:** Requirements have to be as independent as possible in order to clearly identify what to implement and make the order of their implementation irrelevant.

Agile Approaches to RE

- **Requirements Evolution**:
  - **Requirement Elicitation and Prioritization**:
    - At the beginning of every iteration, there is a requirements collection and prioritization activity where new requirements are identified and prioritized.
    - This approach helps to identify the most important features inside the project.
    - If a requirement is very important it is scheduled for the implementation in the upcoming iteration, otherwise it is kept on hold.
    - At the following iteration, the requirements on hold are evaluated and, if they are still valid, they are included in the list of the candidate requirements together with the new ones.
    - Then, the new list is prioritized to identify the features that will be implemented. If a requirement is not important enough, it is kept on hold indefinitely.

Agile Approaches to RE

- **Requirements Evolution:**
  - This approach is able to identify the most important requirements during the whole project, not just at the beginning.
  - Requirements that are not considered very important at the beginning may become relevant at some stage of the project.
  - The decoupling of the requirements allows the implementation of the features in nearly any order; therefore, features are implemented mainly according to their prioritization, not to their functional dependencies.

Agile Approaches to RE

- **Non-Functional Requirements**:
  - AMs do not provide any widely accepted technique for eliciting and managing non-functional requirements.
  - Such requirements are collected implicitly during the requirements collection activity.
  - The need of specifying non-functional requirements is less important than in other context due to the continuous interaction with the customer.
  - After every iteration, the product is released and the customer is able to test the product.
  - If he/she identifies problems related to non-functional qualities, the team can adapt the system to meet such requirements in the subsequent iteration without affecting the schedule too much.
  - Often, the customer does not perceive as high impact many non-functional requirements (e.g., scalability, security, etc.).
  - This may affect the release of the final version of the application, therefore the development team has to guide the customer in order to identify such hidden needs.