



# ***Guidelines for the Specification and Design of Large-Scale Semantic Applications***

ASWC 2009



Oscar Muñoz-García - [omunoz@fi.upm.es](mailto:omunoz@fi.upm.es)

Raúl García-Castro - [rgarcia@fi.upm.es](mailto:rgarcia@fi.upm.es)

How can I **specify** the **semantic** functionalities of my **application**?

How can I carry out the **development process** of my **semantic application**?

Non-expert  
Software Engineer



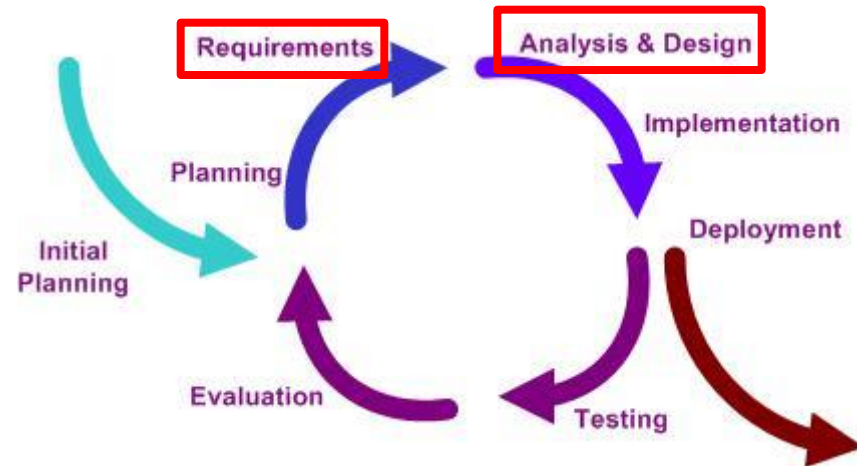
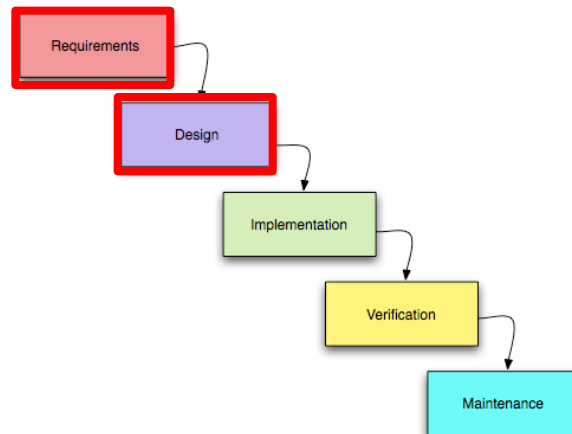
Semantic applications require different software development techniques.

- They manipulate huge quantities of heterogeneous decentralized information.
- They integrate semantic and non-semantic data.
- They explore different sources at run-time.

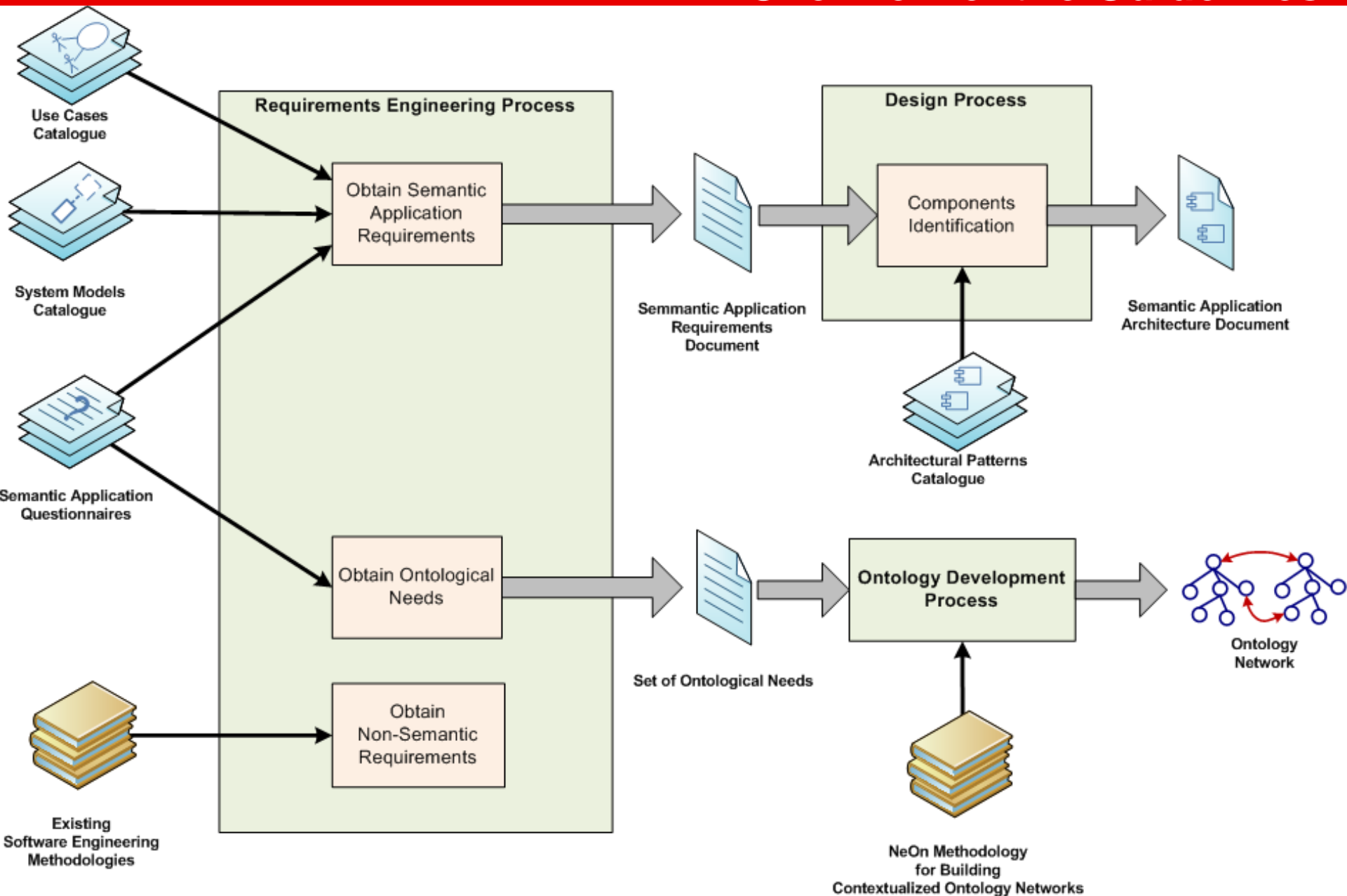
Need **guidelines** that can be easily adapted and **integrated** in the software **development process**



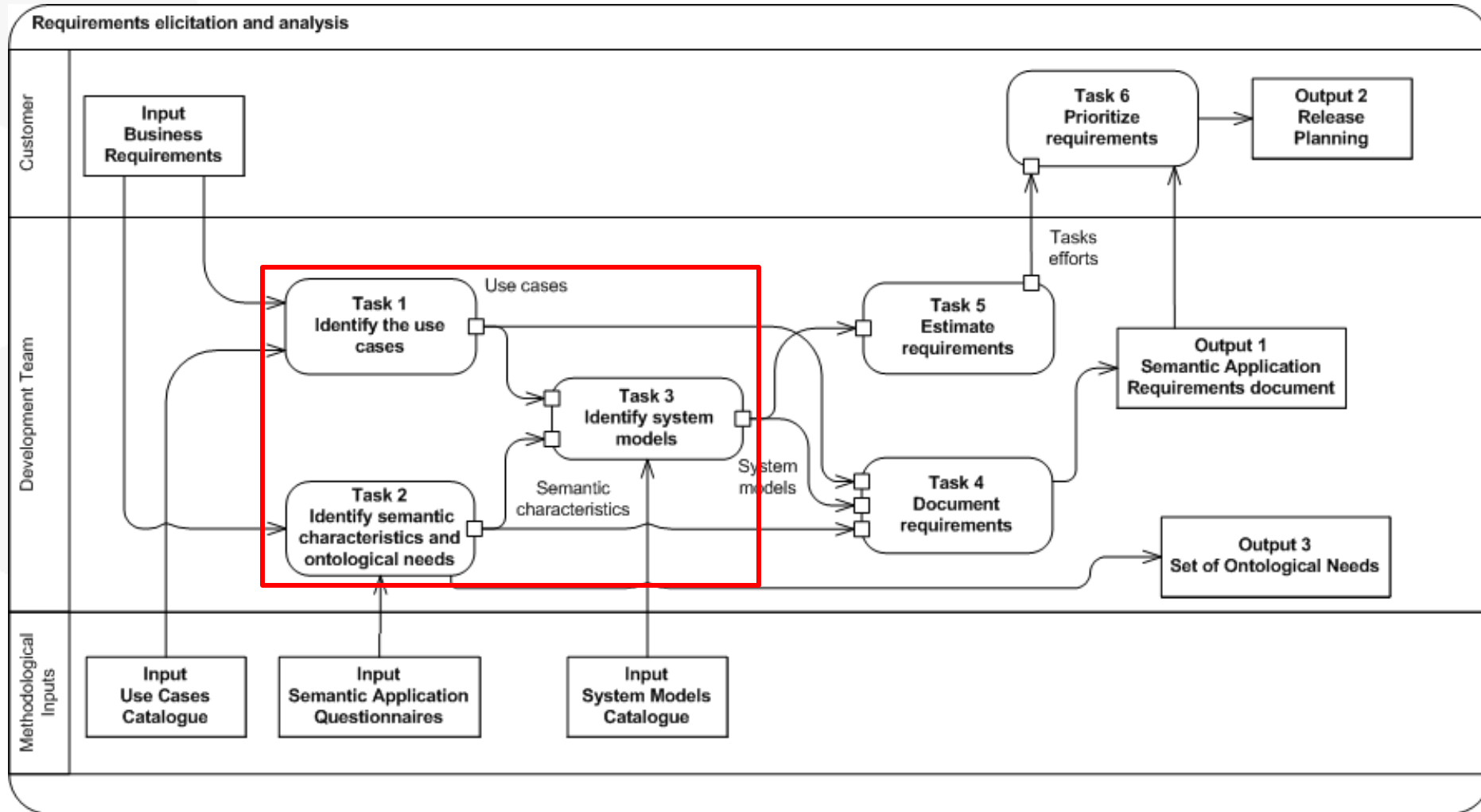
1<sup>st</sup> stage: **Requirements Engineering** of  
Semantic Applications  
2<sup>st</sup> stage: **Design** of Semantic Applications



# Overview of the Guidelines



# Requirements Elicitation Activity



# Task 1. Identify the use cases

## Objective

To gather information about the application from the business requirements and to distil scenarios from this information

## Technique

To select and adapt semantic-related scenarios from a catalogue of 5 use cases

## Use Cases Catalogue

### •Query Information

To obtain integrated information from several resources given a query

### •Search Resources

To obtain resources (annotated with the corresponding metadata) related to a given query

### •Browse Resources

To navigate through categorised resources utilizing ontologies as indexes

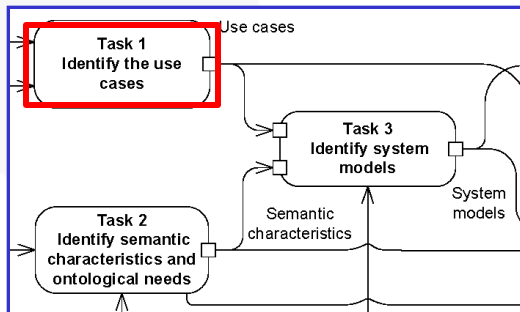
### •Extract Information

To extract meaningful information from a set of resources obtaining after performing a search

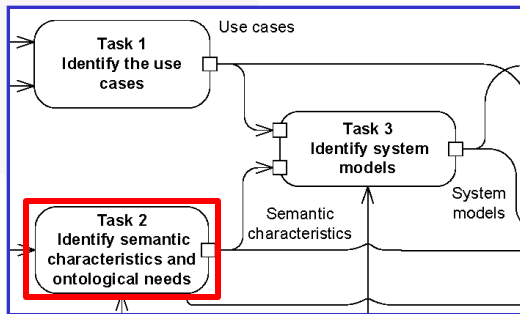
### •Manage Knowledge

To collaboratively construct and evolve shared knowledge by

- Managing classes, properties, instances, mappings, annotations...
- Learning or populating an ontology, ...



## Task 2. Identify Semantic Characteristics and Ontological Needs



## Objective

To collect semantic application requirements and ontological needs

## Technique

To apply 4 questionnaires oriented to obtain the values of 32 characteristics and ontological needs

## Characteristics

<p><b>Ontologies</b></p> <ul style="list-style-type: none"> <li>(1) Use of a single ontology or a network of ontologies</li> <li>(2) Design-time or run-time ontology selection</li> <li>(3) Use of generic or domain-specific ontologies</li> <li>(4) Generation of new ontologies</li> <li>(5) Use of internal or external ontologies</li> <li>(6) Ontologies reuse and reengineering</li> <li>(7) Non-ontological resources reengineering</li> <li>(8) Ontologies dynamicity</li> <li>(9) Centralization or distribution of ontologies</li> <li>(10) Scalability regarding the number of ontologies</li> <li>(11) Scalability regarding the number of ontology elements</li> <li>(12) Ontologies encoding heterogeneity</li> <li>(13) Ability to resolve conceptual heterogeneity in ontologies</li> </ul>	<p><b>Data</b></p> <ul style="list-style-type: none"> <li>(14) Data domain dependence</li> <li>(15) Data Generation</li> <li>(16) Use of internal or external data sources</li> <li>(17) Use of linked data</li> <li>(18) Data distribution</li> <li>(19) Data dynamicity</li> <li>(20) Design-time or run-time data selection</li> <li>(21) Data scalability</li> <li>(22) Use of non-semantic data</li> <li>(23) Data encoding heterogeneity</li> </ul>
<p><b>Reasoning</b></p> <ul style="list-style-type: none"> <li>(24) Kind of semantic reasoning</li> <li>(25) Sound reasoning</li> <li>(26) Complete reasoning</li> <li>(27) Hybrid reasoning</li> <li>(28) Reasoning with contradictory data</li> <li>(29) Reasoning with incomplete data</li> <li>(30) Reasoning with uncertainty</li> <li>(31) Distributed reasoning</li> </ul>	<p><b>Non-functional</b></p> <ul style="list-style-type: none"> <li>(32) Interoperability with other applications</li> </ul>

# Task 3. Identify System Models

## Objective

To preliminary specify the application from

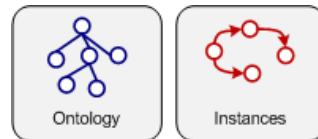
- An external perspective  
Systems involved and their limits
- A structural perspective  
By modelling the structure of the information processed

## Technique

To build system model using a set of templates taking into account

- Identified use cases
- Application characteristics

## Basic Symbols



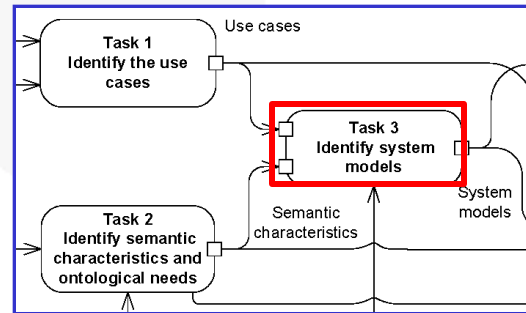
## Relationships



...

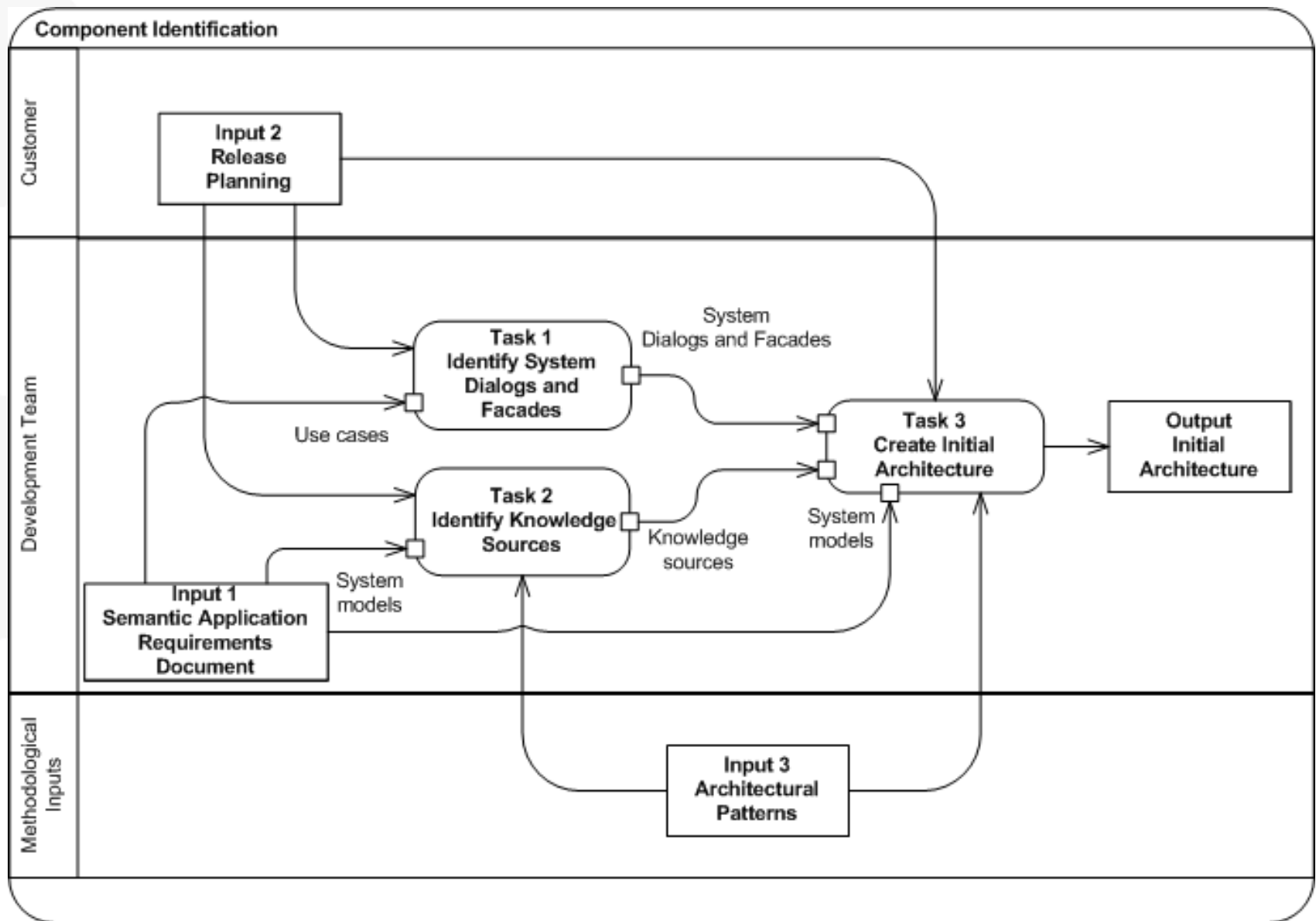
## System Models Templates Catalogue

- Includes 11 system models templates



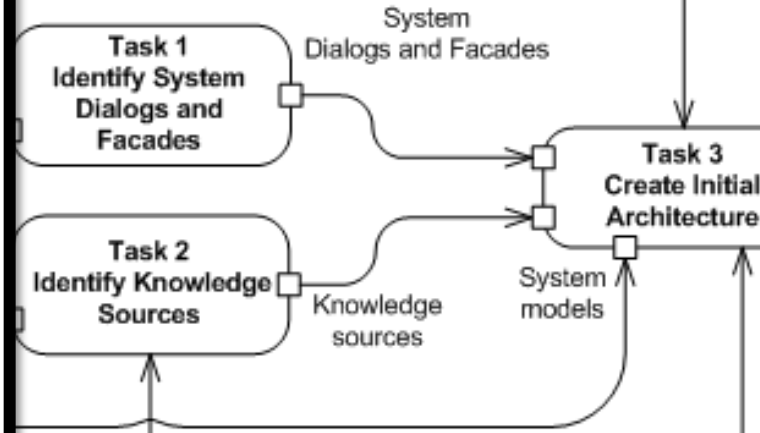
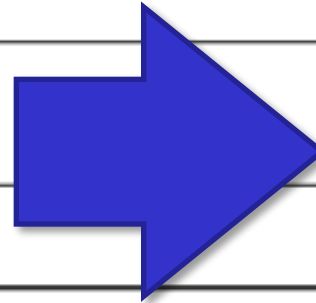
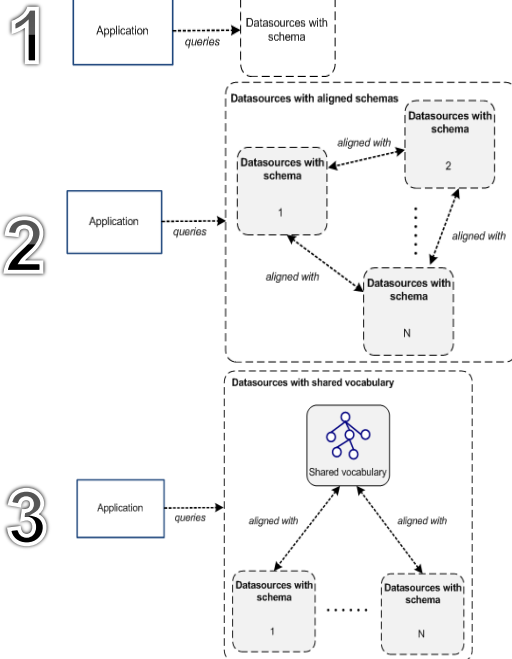


# Component Identification Activity

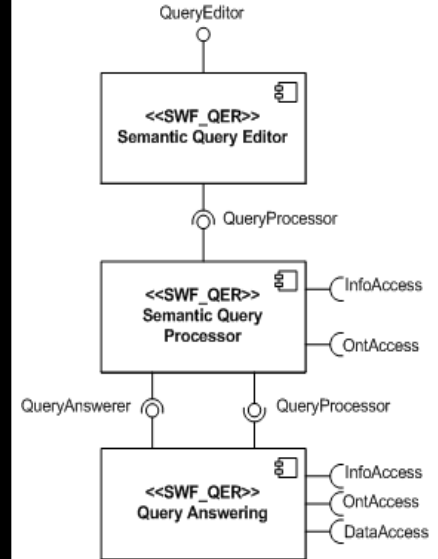


## Component Identification

### System Model Templates



### Pattern



### Objective

To draw the application architecture that form an idea of how the components might fit together

### Technique

To draw the architecture by applying 28 architectural patterns associated to the previously identified System Models

- Our main goal is to facilitate the development of semantic applications for non-expert software engineers.
  - Adapting the Requirements Engineering and Design processes from SE.
  - Allowing to design the architecture of SA from scratch and to include semantic components into traditional information systems.
- The techniques described are novel and specially oriented to the specification and design of the semantic functionalities of an application.
- The semantic **characteristics** of an application will influence the whole development process.
- The **use cases** are useful to start analysing the application from concrete functional user needs.
- The **system models** are graphical representations of the macro-structure and environment of the application. We have provided **architectural patterns** that aim to facilitate the Architectural Design of the Semantic Application.
  - These architectural patterns are not bound to a particular implementation.