Bridging the Gap between Design and Implementation for Multi-agent Systems

Mohamed M. Elammari¹, Bilal A. Aljabour²,
¹Faculty of Information Technology, University of Benghazi, Benghazi, Libya
²Delta Apps, Benghazi, Libya
¹mohamed.elammari@benghazi.edu.ly, ²bilal@deltapps.com

Abstract. Although of the existence of many multi agent systems (MAS) development methodologies, the MAS field still suffering from some problems, those represent serious obstacles for the success and propagation of such kind of systems. In this paper, one of these problems is presented, that is the gap between MAS design methodologies and implementation frameworks, and a solution for such problem is proposed. This is done by selecting some agent-oriented methodologies and extending each of them towards an implementation framework. The extension processes are accomplished using a proposed criteria for multi agent systems development. By analyzing the commonalities between the extension processes a common bridge between design and implementation is being created. The bridge is expected to be helpful for anyone who wants an easy and straightforward development from the first stage of design using any of the selected methodologies to the implementation using the selected framework. Also, this work opens the door for future work towards generalizing the created bridge to include more design methodologies and more implementation frameworks.

Keywords: Agent Oriented Software Engineering Methodologies, HLIM, MaSE, Prometheus, Jadex

1. Introduction

The growth of interest in multi agent system has been very recognizable in the last years. Several software implementation frameworks to build agent systems (for example FIPA-OS [8], JADE [4] and BDI [12] Jadex [12] [3]) have been proposed. These implementation frameworks demanded that, system analysis and design methodologies should exist to support a complete multi agent systems development lifecycle. So, many of design methodologies were proposed (for example HLIM [9], Gaia [14] [15], Prometheus [10], Tropos [5], MASD [1] and MaSE [56] [7]). However, despite the possibilities provided by the frameworks and the methodologies, there is a gap between them. The reason of this gap is most methodologies do not really consider detailed design and implementation environment issues. Anyway, this paper presents an attempt to propose a solution for this problem by selecting three design methodologies, those are (HLIM, MaSE and Prometheus), and an implementation framework, which is BDI Jadex. By studying BDI Jadex, specifying the specifications of BDI Jadex agents, and specifying what is required from the designers to submit indetailed design to be ready for implementation using Jadex. Then, apply the proposed (Agent Systems Development Criteria) (ASDC) on the methodologies. The proposed criteria is assumed to cover the main aspects of multi agent systems. While of evaluation of the different aspects in the methodologies the additions towards a detailed design which supports implementation using BDI Jadex being made. Depending on the results of applying the ASDC on the three methodologies, the main commonalities being extracted and a bridging framework between design and implementation was being created.

2. The Agent Systems Development Criteria (ASDC):

As known, any integrated multi-agent system development methodology should provide support for many aspects; those represent together the spirit of agency. Here, the proposed criteria presents these aspects keeping in mind the target, which is BDI Jadex:
1. **System view:** Understanding agent system requires a high-level view of how the system works as a whole to accomplish some application related purpose. Also, system view explains the system requirements in a high level. The bridging process already works on the design and detailed design phases of the methodology under study, but presents how the system view is covered by the methodology to assure that, the philosophy of the methodology will not be altered. Also, by doing this, any extension made will be clearer for the designer(s).

2. **Agents’ Internal structure:** Agents’ internal structure has to be expressed by aspects such as goals, plans, and beliefs for the support of BDI Jadex. A development methodology should facilitate the discovery of agents needed along with their internal structure.

3. **Agents Relationships:** In multi-agent systems, the system works as a one entity to achieve the different goals of each agent, and the systems overall goals as well. To accomplish such harmony of coordination, many kinds of interactions between system agents are required. An agent might be dependent on another agent to achieve a goal, perform a task, or supply a resource. Also, a conversation between two agents may be needed for the accomplishment of a specific task.

4. **Detailed design:** This is the most important phase for the implementers, where entities are implemented are those produced in the detailed design phase. For Jadex implementation, the following elements must be provided by the methodology detailed design phase for an easy and direct transition towards implementation:
   1. A list of the system agents, the goals of each agent with all individual goal settings, the appropriate plans for each goal with all individual plan settings, and the beliefs will be used by each plan while trying to achieve some goal. These will be used to create the Jadex Agent Definition Files (ADF) for each agent in the multi agent system.
   2. Development methodology must represent the system behavior, by representing the tasks and settings of each plan in the system and all the system interactions as well.

5. **Implementation (using Jadex):** Agents in Jadex are defined by the so called Agent Definition Files (ADF), these are XML descriptions, and a set of java classes (referenced in the ADF) to implement the desired behavior. The creation of ADF’s and Java coding will be straight forward if the detailed design phase submits the elements as mentioned in points 1, 2, 3 and 4 above.

In the following sections III, IV and V, the ASDC is applied on HLIM, MaSE, and Prometheus methodologies. Keeping in consider that, the philosophy of themethodologies will not be altered.

Due to the lack of space, in this paper we present almost the complete appliance of ASDC on HLIM methodology and only the major points of the applying ASDC on MaSE and Prometheus methodologies:

### 3. Applying ASDC on HLIM

From its name, HLIM is concerned to present a strong foundation of an agent oriented methodology, based on a clear understanding of system requirements starting from the Use Case Maps (UCM’s) [11] scenarios of the system. The detailed design phase is not covered, because HLIM stops in the intermediate design level.

1. **System view:**
   In HLIM, system view is covered by the so called High Level Model, it captures the high level behavior of the agent system being developed. System designer's use the UCM's for the representation of the different system scenarios to make a clear system overview and bridge the gap between requirements and analysis.
   By the end of this level designers get the following: Use case maps explaining all system scenarios, the behavior of each system agent in a high level, the external interactions with the system (operators, sensors ...etc.), the information affects with or affected by different system scenarios (preconditions, post conditions, data sources, etc.), a high level view of agents interactions with each other within the different scenarios, and (Agent Interactions).

2. **Internal structure:**
   HLIM describes the internal structures of the agents discovered in the High Level Model with the so called Internal Agent Model which derived directly from the High Level Model.
Although the HLIM methodology is not explicitly representing the DBI Paradigm, the foundations of such paradigm is already exist. The concept of Belief is identified using the Pre Condition and Post Condition derived from the high level model's Use Case Maps (UCMS). The concept of Goal is identified using the Paths segments traverses the agent, and also the dynamic stubs in the path. Plan concept is identified using the combination of (Goal, pre & post conditions, Tasks). By this, HLIM relates the plan with its beliefs, and the goals which the plan is applicable for. HLIM Internal Agent Model delivers the agents’ internal structure described in terms of their goals, beliefs, plans, and tasks described in a tabular model (a table for each agent). Figure 1 represents the HLIM Internal Agent Model.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Precondition</th>
<th>Postcondition</th>
<th>Task</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: The Internal Agent Model

For a clearer representation of the internal agent model, it is refined it in terms of BDI as demonstrated in Figure 2.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Belief</th>
<th>Plan</th>
<th>Task</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>g1</td>
<td>p1</td>
<td>t1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>t2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tn</td>
<td></td>
</tr>
<tr>
<td>g1</td>
<td>b1 = ..</td>
<td>p2</td>
<td>t1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b5 = ..</td>
<td></td>
<td>t2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bn = ..</td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tn</td>
<td></td>
</tr>
<tr>
<td>g1</td>
<td>b7 = ..</td>
<td>pn</td>
<td>t1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>t2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tn</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Refined internal agent model

From figure 2 it is noticed that, for the same goal $g1$, there are many alternative plans $p1, p2, ..pn$, and each plan $pn$, consists of many tasks $t1, t2, ..tn$. HLIM stops in here for the internal structure. Now, a study for each internal structure aspect will be done, such as Beliefs, Goals, and Plans in order to evaluate and extend for new models, when required.

- **Plan details:**

  Plans represent the agent’s means to act within the system. Therefore, the plans have to be predefined by the developer to compose the library of tasks the agent can perform within its lifecycle. In Jadex, plans consist of two parts: A plan head and a corresponding plan body. The plan head is declared in the ADF whereas the plan body is realized in a concrete Java class.

  The head of the plan requires specific fields, like plan priority between the other alternative plans associated to achieve the same goal. Also, the plan triggers, i.e. which goal will trigger this specific plan. Plan triggers can be specified from the Refined Internal Agent Model. Plan priority can be specified later by the designer during the detailed design. Plan body can be represented using UML Activity Diagram, one for each plan, for a graphical description of plan tasks or functionality described textually in HLIM’s Internal Agent Model, as shown in Figure 3.
The plan body for each agent plan can be represented using UML class diagram, which can be derived directly from the Refined Internal Agent Model, as shown in Figure 4.

### Figure 3: activity diagram explains plan logic

The plan body for each agent plan can be represented using UML class diagram, which can be derived directly from the Refined Internal Agent Model, as shown in Figure 4.

### Figure 4: Deriving the class diagram from the Refined Internal Agent Model

<table>
<thead>
<tr>
<th>Goal</th>
<th>Belief</th>
<th>Plan</th>
<th>Task</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>g1</td>
<td>p1</td>
<td>t1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>t2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tn</td>
<td></td>
</tr>
<tr>
<td>g1</td>
<td>b1 = ..</td>
<td>p2</td>
<td>t1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b5 = ..</td>
<td></td>
<td>t2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bn = ..</td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tn</td>
<td></td>
</tr>
<tr>
<td>g1</td>
<td>b7 = ..</td>
<td>pn</td>
<td>t1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>t2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tn</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>Gn</td>
<td>pz</td>
<td>t2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tn</td>
<td></td>
</tr>
</tbody>
</table>

**Goal details:**

Goals make up the agent's motivational stance and are the driving forces for its actions. The representation and handling of goals is one of the main features of Jadex. For each agent goal, system designers have to prepare answers for many questions like, what is the goal type, in what cases the goal can be considered as achieved?, in what cases the goal can be considered as failed to achieve, does the agent can adopt more than one instance at a time?, what is/are the condition(s) that makes the agent stop or resume this goal pursuing?, can this goal be dropped under some conditions?, etc

**Belief details:**

Beliefs represent the agent's knowledge about its environment and itself. For all system beliefs, a list of them and their classes (data types) is created, as shown in Table 1.

<table>
<thead>
<tr>
<th>No.</th>
<th>Belief name:</th>
<th>Belief Class:</th>
<th>Plan name</th>
<th>Attributes</th>
<th>Operations</th>
<th>Plan tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

105
3. Relationships

Relationships represented in HLIM by Agent Relationship Model (dependency, jurisdictional), the Conversational Model, and the Contract Model. They are all representing various kinds of agents relationships, keeping in consider the managerial and organizational point of view for each one. They are all organizing the relation among the different agents within the multi agent system being developed.

The conversational model (Figure 5) derived directly from the Internal Agent Model is the most important relationship model in HLIM, because it represents the messaging exchange among agents. It will be represented using AUML interactions protocols, (Figure 6) to compliment the existing tabular representation.

The Agent Relationship Model (dependency, jurisdictional) and the Contract Model do not need a representation of AUML interactions protocols, where there is no conversation occur between the involved agents in such relations. So, the existing tabular representation is clear and enough.

<table>
<thead>
<tr>
<th>Sender</th>
<th>Parameters</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: HLIM Interagent conversational model

Figure 6: Extended Interagent conversational model using AUML interaction protocols

4. Detailed Design

This is the closest phase to the implementation. It is not covered by original HLIM. aDetailed Design Phase will be created, which submits the following Jadex implementation framework details:

1. Models for agents combining the system, there goals, plans, and beliefs.
2. Detailed design for each individual agent goal, the settings and attributes of each goal, those will be straight forward moved towards Jadex ADF's.
3. Detailed design of each individual agent plan, the head details and a skeleton of java body for each plan, those will be straight forward moved towards Jadex ADF's.
4. List of all beliefs used by agents' plans.
5. Detailed design of all agents interaction, translated to Jadex message events.

At a moment, the designers get closer to implementation. Now, in more details, models created in previous phases will be represented using Jadex terms. Those models are:

1. Refined internal agent model
2. Class diagram for plan body representation.
4. Extended Interagent conversational model.

In the Following, the detailed design is represented for them all:
a. From refined internal agent model to detailed goal model:

According to goals information in the refined internal agent model and the designer prepared answers about goal detail, setting, attributes, and other technical details, the detailed goal model (Table 2) will be created. That will help to transit goals directly to Jadex XML ADF. Answers about goal details give the designer full information about the goal and its settings, and how will be represented in Jadex XML ADF.

Here, some settings, and their means in Jadex terminology is represented to be used later for goal detailed design:

<table>
<thead>
<tr>
<th>Question</th>
<th>Mapping to BDI goals terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the goal type? <em>(Achieve, Perform, Query, or Maintain).</em></td>
<td>One of the four goal types supported by Jadex <em>(Achieve, Perform, Query, or Maintain).</em></td>
</tr>
<tr>
<td>5. In what cases a goal can be considered as achieved? Name the beliefs and conditions.</td>
<td>Target condition</td>
</tr>
<tr>
<td>6. In what cases a goal can be considered as failed to achieve? Name the beliefs and conditions.</td>
<td>Failure condition</td>
</tr>
<tr>
<td>Does the agent can adopt two instances of this goal at a time?</td>
<td>The goal is Unique</td>
</tr>
</tbody>
</table>

Table 2: The translation of goal terms from the goal details table

**Goal Types:** Achieve, Perform, Query, Maintain.

**Goal Settings:** Unique, Exclude, Target condition, Failure condition, Context condition, Drop condition, Deliberation, Inhibits.

**Attributes:** Retry, Retry delay

The table above is summarized in the following goal detailed design model, (Figure 7):

![Figure 7: Goal detailed design model](image)

b. From refined internal agent model to plan detailed design model:

From Refined internal agent model, for each plan a list of goals or events those trigger the plan is made, and a list of tasks combining the plan, (Figure 8):

**Trigger Types:** Message, Goal, Condition *(Belief change).*

**Body:** Plan Body Constructor…e.g. new RepairPlan()
c. From activity diagram and class diagram to Java body skeleton:
Plan task mapped directly to methods in java code, and attributes mapped to beliefs:

```java
public void body() {
    method1() {
        ....
    }
    method2() {
        ....
    }
    ....
    methodn() {
        ....
    }
}
```

d. From Extended interagent conversational model to Agent Messaging Model:
All interagent interactions will be translated into message events in Jadex. (Figure 9), represents the Agent Message Model details, that will be straightforward converted into Jadex ADF message events in the implementation phase.

```
Figure 9: The Agent Messaging Model
```
5. Implementation with Jadex

Here, the creation of Jadex Agent Definition Files (ADF) is presented:

* Goals to XML ADF:

To show the move from detailed goal model to Jadex XML ADF, an example XML snippets code for some kind of goals, that is a Performgoal, is presented.

**XML representation for perform goal:**

```xml
<agent ...>
  ...
  <performgoal name="..." retry="..." exclude="...">
    <contextcondition>
      ...
    </contextcondition>
  </performgoal>
  ...
</agent>
```

* Beliefs to XML ADF:

A belief that contains more than one value called a beliefset.

**XML representation for beliefs:**

```xml
<agent ...>
  ...
  <beliefs>
    <belief name="my_location" class="String">
      <fact>"Libya"</fact>
    </belief>
    <beliefset name="my_friends" class="String">
      <fact>"Ali"</fact>
      <fact>"Adam"</fact>
    </beliefset>
    ...
  </beliefs>
  ...
</agent>
```

* Plan head to XML ADF:

A sample code for ADF that shows the declaration of a plan is given below:

```xml
<agent ...>
  ...
  <plans>
    <plan name="">
      <body>new ...()</body>
      <trigger>
        <messageevent ref="..."/>
      </trigger>
    </plan>
    ...
  </plans>
  ...
</agent>
```

* Summary of the appliance of ASDC on HLIM methodology:
Applying ASDC on MaSE

MaSE internal architecture must support BDI and covers the detailed design phase, which affords all the implementation details required by Jadex. MaSE takes an initial system specification, and produces a set of formal design documents in a graphically based style [13]. MaSE combines several preexisting models into a single structured methodology.

Figure 10: From HLIM to Jadex implementation - detailed

Figure 11: From HLIM to Jadex implementation

4. Applying ASDC on MaSE

MaSE internal architecture must support BDI and covers the detailed design phase, which affords all the implementation details required by Jadex. MaSE takes an initial system specification, and produces a set of formal design documents in a graphically based style [13]. MaSE combines several preexisting models into a single structured methodology.
• **System View:** In MaSE, the system view is covered using the system requirements and use cases. The system requirements may include detailed technical documents, user stories, or formalized government specifications. Use cases are drawn from the system requirements. Use cases are narrative descriptions of sequence of events that defines system behavior.

**Agents' Internal structure:** From the Assembling Agent Classes step in MaSE, it is independent of a particular multi agent system architecture, where it leaves the choice to the details designers and implementers either to design their own architecture or use a predefined architecture. Therefore, the work here will be focused on how to make an internal structure for MaSE, which is the BDI architecture to serve the main objectives of this research. Now, the MaSE concepts and/or models those can be mapped or transformed to BDI concepts will be specified.

• **Making a BDI internal Architecture for MaSE**

  • **Goal concept:** The concept of goal represented in MaSE Capturing Goals step is a system goal, where the transition towards BDI here requires agent goals. From the MaSE Refining Roles "the related goals may be combined into single roles", actually they must be combined into a single role. By doing that some kind of container will be found, that is the role. By definition, role represents an entity in the system that performs some function within the system and responsible for achieving, or helping to achieve specific system goals or sub-goals, and more important that the role is transformed into an agent class in the design phase. So, a role that will be transformed later to an agent has been created, and such role has its own goals.

  Now, and to be ready for detailed design phase, For each agent goal, system designers have to prepare answers for many questions like, what is the goal type, in what cases the goal can be considered as achieved?, in what cases the goal can be considered as failed to achieve, does the agent can adopt more than one instance at a time,…etc.

  • **Plan concept:** Plan concept represents the agent's means to act within the system, and represent the main functionality of agents. Also, it describes the agent behavior to achieve its goals. In MaSE, the plan concept can be extracted from the Role Model, where Tasks associated with each role, in Role Model, describe the behavior that the role must exhibit to successfully achieve its goals. The detailed description of a role task that is represented in more details in Concurrent Task Diagram is very similar to the plan concept. So, each role task will be mapped into a Plan.

  • **Belief concept:** The MaSE Concurrent Task model represents the roles' tasks using Finite State Automation, which consist of states and transitions. Sometimes, the start of task execution requires a change of some system knowledge component, which is called a Transient Task. The concept of states those associated with tasks is very similar to the concept of belief. So, each variable that represents an informational state will be assumed as a belief.

  • **The relation between goal, plans, and beliefs concepts extended from MaSE:**

    Now, the BDI internal architecture for MaSE has been made. But, the integration of such architecture has to be assured. To summarize what is exist now:

    1. System divided into roles (Agents) combining the system.
    2. Each role has its own (Goals).
    3. Each role has its own set of tasks (Plans).
    4. Each task has a set of states (Beliefs).

    For integration, some things must be insured, those are, there is a clear definition for each agent goal, all alternative plans with their tasks those must be defined, and all beliefs required by each plan while of the execution must be listed. To document and organize all these information, an extension to the Internal Agent Model has to be made, which is the same model used with HLIM. In the following, (Figure 12) demonstrates the extended MaSE methodology that supports BDI implementation.
V. Applying ASDC on Prometheus

- **System View:**
  System view is covered by Prometheus using the system Specification Phase, which focuses on the identification of actors and their interactions with the system, developing Use Case scenarios illustrating the system’s operation, identifying the system goals and sub-goals, identifying any external data used, and grouping goals and other items into basic roles (functionality) of the system.

- **Agents' Internal Structure:**
  Prometheus uses the BDI paradigm in the representation of agents' internal structure. Prometheus Detailed Design Phase focuses in developing the internal structure of each agent.

Analyzing how Prometheus represents each BDI concept, and how to move toward implementation:

- **Goal concept:**
  Although, Prometheus claiming that they are supporting the BDI paradigm the system goals and sub-goals identified in the system specification phase are not further mentioned. Even while they group goals in system specification phase into system functionality of the system, no agent goals are identified. The methodology represents many descriptors in the Detailed Design Phase, the Plan descriptors, the Data Descriptors, and the event descriptors, but not representation for Goal Descriptors.
Actually, they admit that at the end of the Discussion and Conclusion section by saying "Other areas for future work include: clearer integration of goals as a first class concept (currently goals are implicit in functionalities), extensions to the graphical notation to allow percepts, actions, goals, and (some) sequencing information to be specified". The proposal for goal integration is to recognize functionalities included in the Agent Descriptor as agents' goals. Where functionalities originally extracted from the overall system goals, so, it is very reasonable to remap them to goal, especially after they are decomposed and related with agents. Prometheus uses a consistent notation to depict agents, events, plans, capabilities, and data (beliefs), (Figure 13). A notation for goal representation was added to be used for goals representation in the different methodology diagrams.

![Figure 13: Prometheus notation and the added Goal notation](image)

- **Plan concept:**
  Plan concept represents the agent's means to act within the system, and represent the main functionality of agents. The plan is how the goal will be achieved. There should be one or more alternative plans to achieve each goal. Prometheus Agent Descriptor Diagram was already added by mapping agent functionalities into goals. Now, the same model will be refined to encompass goals and plans in norms of Goal n (Plan m,...). Also, a list of main plan actions is added to the model in norms of Plan n (Task m,...).

- **Detailed design:**
  At this stage of the design, the methodology becomes more specific to agents these use user-defined plans, triggered by goals or events, such as the various implementations of Belief, Desire, Intention (BDI) systems. Prometheus presents plan descriptors and data descriptors those can be transformed into plan details model and beliefs details. Here, a goal descriptor depending on the goal concept is presented, to include:

  - Goal identifier, goal description, and answers about goal settings like what is the goal type? In what cases the goal can be considered as achieved? In what cases the goal can be considered as failed to achieve.

To summarize, Prometheus should submit the following details for Jadex:

1. Models for agents combining the system, and there goals, plans, and beliefs.
2. Detailed design for each individual agent goal, the settings and attributes of each goal, those will be straight forward moved towards Jadex ADF's.
3. Detailed design of each individual agent plan, the head details and a skeleton of java body for each plan, those will be straight forward moved towards Jadex ADF's.
4. List of all beliefs used by agents' plans.
5. Detailed design of all agents interactions, translated into Jadex message events.

In more details, and using Jadex terms, Created models are:

1. Refined Agent Descriptor Model.
2. Internal Agent Model.
3. Plan Activity Model.
4. Agent Conversations Model.

Depending on Prometheus models and the new added models, the same steps done with HLIM can be done here, to get a detailed design formed as required for implantation. (Figure 14), demonstrates the extended Prometheus that supports BDI Jadex.
6. The Bridge

While of the appliance of the ASDC on the HLIM, MaSE, and Prometheus methodologies, three methodologies have been extended from design to implementation. New models, new phases, and new concepts have been added to make the methodologies support the BDI paradigm with all design details required for the implementation using BDI Jadex framework. Depending on the extensions processes for each methodology by analyzing the extensions done, there are some concepts which require the designers to extract themselves from the methodology; those are often the conceptual concept, which is called here the "Bridge Flexible Part". Other design details built upon the flexible part, and are straightforward mapping towards BDI Jadex implementation. Which are called here the "Bridge Fixed Part". The bridge parts with details are represented in the following:

- **Bridge Flexible part**: It is called flexible, because it depends on the designer who wants to extend the methodology and/or implement using BDI Jadex. In the proposed bridge, depending on the
appliance of ASDC on HLIM and MaSE design methodologies, guidelines on how to extract the BDI concepts from a design methodology are pointed. Anyone who wants to add more guidelines depending on their expedience of applying the ASDC on any other methodology will enrich this bridging framework.

Where each agent oriented methodology has its own style on how to represent the different agent system aspects, general definitions and guidelines to guide the designer in how to extract the main concepts required by BDI Jadex framework are sated. These concepts are (Goal, Plan, and Belief), as following:

- **Goal extraction:**
  1. Goals are the agents' motivational stance and are the driving forces for its actions.
  2. The designer has to ensure that the goals are agent goals not system goals, and if they are system goals, he has to find some way on how to associate related goals with some system agent.
  3. The alternative plans to achieve the goal have to be defined.

- **Plan extraction:**
  1. The concrete actions an agent may carry out to reach its goals are the plans of the agent.
  2. The combination of Goal, Preconditions, Postconditions, and Tasks is mapped into plan, in Use Case Maps based methodologies.
  3. The task associated with a role is straightforward mapped into a plan, in role based methodologies.
  4. The plan has to be related with its different triggers.
  5. The plan behavior, including parameters, has to be graphically represented using a design technique such as UML activity diagram, or Finite State Machine.

- **Belief extraction:**
  1. The agent's knowledge about its environment and itself are recognized as beliefs.
  2. The pre & post conditions governing the scenarios are mapped into beliefs in the use case maps based methodologies.
  3. The different states in the role task (plan) representation are recognized as beliefs.

  4. The result of the flexible part of the bridge will be a set of defined agents, the goals of each agent, the alternative plans to achieve the goals, and the beliefs required by each plan while of execution. The result is summarized in the "BDI Internal Agent Model". Where, a model for each agent in the system being developed has to be created.

This Flexible part is concerned in how to make the methodology support BDI. So, If the methodology explicitly support BDI concepts, the designers can skip the previous (Goal, Plan, Belief) extraction steps and create the BDI Internal Agent Model directly.

- **Agents Conversations:** all kinds of agents interactions extracted from the different methodology models can be summarized in Agents conversations model.

- **The Bridge Fixed part:** Now, a catalogue of agents and their goals, plans, and beliefs has been created. Also, all the conversations have been captured in the Agent Conversations Model. Depending on these details designers are ready now to move towards Detailed Design. This is accomplished using, the Goal Detailed Design Model, the Plan detailed design model, the Belief Details Model, and the Agent Messaging Model, as following:

- **Bridge summary:**
  1. By analyzing the commonalities between the extensions made on HLIM and MaSE methodologies, there is a flexible part depends on the designers experience and their conceptual thinking, and a fixed part depends on the flexible part and take the development process directly towards BDI implementation.
  2. If the methodology already supports the BDI paradigm, the designers can skip the BDI concepts extraction step.
  3. As shown in the methodologies extensions, the move towards Jadex BDI implementation will be straightforward.(Figure 15), illustrating the Bridging framework.
**Figure 15: Illustrating the Bridging framework**

7. **Conclusion**

- The original HLIM methodology is a well formed methodology and was built carefully according to agency theory. Where, although it is a high level and intermediate, it provided a strong foundation for extension towards detailed design and implementation, and the extended HLIM methodology is a BDI methodology.
- Original Prometheus methodology partially supports the BDI paradigm.
- After this research, anyone who wants to implement using BDI Jadex can move from Prometheus through the proposed bridge towards BDI Jadex implementation framework.
- Original MaSE methodology does not support the BDI paradigm. After this research, anyone who wants to implement using BDI Jadex can move from MaSE analysis to through the extended internal architecture, the detailed design towards Jadex implementation framework.
- The Agent Systems Development Criteria (ASDC) succeeded to evaluate the methodologies, and helped in the extension process. It can be applied on other methodologies for the purpose of extension and transition towards BDI implementation.

**Future work:**

This research leaves open doors for multi agent systems development future research. The future work is expected to be built upon this research is summarized in the following points:

- The flexibility in Flexible Part of the bridge opens doors for anyone who wants to add guidelines on how to extract BDI concepts from any other methodology rather than the three ones presented in this research.
- This research makes a (3:1) relation from methodology to implementation framework. In future, this relation may be improved for (n:1) and (n:m).

**References:**