CASSANDRA:

An Automated Software Engineering Coach

Markus Schacher
KnowGravity Inc.
Badenerstrasse 808
8048 Zürich
Switzerland
Phone: ++41-(0)1/434'20'00
Fax: ++41-(0)1/434'20'09
Email: markus.schacher@knowgravity.com

Version: 1.1.0 / 1. November 2001

Abstract

This document describes the software engineering research project CASSANDRA. CASSANDRA is an automated assistant that guides software developers through the software development process. It analyzes project information held in one of many familiar UML-based CASE tools and derives issues to be clarified, creates new models or suggests the next steps to be done in a project.

The usage of CASSANDRA is very simple, since it is designed as a replacement of a human coach, i.e. it provides a very simple, human-like user interface. CASSANDRA is the result of synergies between complementary disciplines such as software engineering, artificial intelligence, psychology, research and real world project work.

Keywords

Agents, Artificial Intelligence, Automated Coaching, CASE, Information Systems, Meta Modeling, Methodologies, Model- and System-Validation, Object-Oriented Technology, Ontologies, Prolog, Requirements Engineering, UML.

1. Introduction

Today, advanced IT skills become more and more relevant to all business areas. This is basically caused by the technological progress that takes place in a faster pace than ever before. Even non-IT specialists are increasingly forced to adapt their IT systems to constantly changing business needs.

On the other side, all the world is short of professional IT specialists. This results in an increasing number of ad hoc projects that are developed in an uncoordinated manner. However, the resulting IT systems usually hardly fit together and must be thrown away or fundamentally redesigned after they have been in operation for a few month. This is where CASSANDRA comes in.

CASSANDRA stands for (**C**assandra - an **A**ssistant for **S**ystem **S**pecification **AND** **R**equirements **A**nalysis). It is an experimental research platform that provides specialized software engineering know how to IT people not familiar with a particular engineering technique or even to non-IT people. Based on the Unified Modeling Language (UML) standard [OMG99], CASSANDRA provides support for the software engineering process starting at the initial requirements gathering phase. This support is provided in a style that resembles a human coach: CASSANDRA is proactive, asks the user questions, makes suggestions and explains her thoughts if the users requests.

2. A Software Development Process

Before CASSANDRA has been initially designed, the software development process has been analyzed. Based on the resulting “business process model”, the following main processes have been identified, that are potential candidates for being supported by a tool like CASSANDRA: Analysis, Design, Construction, and Project Management.

2.1 The Analysis Process

The primary goal of the *Analysis Process* is to elaborate a precise specification of all business relevant facts for an IT solution. It mainly consists of the following iterative activities:

- **Fact Finding**
  identification of relevant business facts in collaboration with the domain specialists;
- **Fact Representation**
  documentation of the relevant business facts in a precise way;
- **Fact Verification**
  validation and verification of the documented business facts in collaboration with the domain specialists.

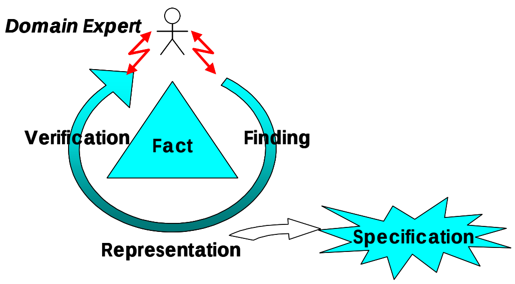


Figure 1: The Analysis Process

The *Specification* is the main deliverable of the analysis processes and is usually heavily documented using UML. It also may be supplemented by user interface and functional prototypes developed in collaboration with users and domain experts.

2.2 The Design Process

The primary goal of the *Design Process* is to elaborate a set of mechanisms and design patterns for mapping elements in the specification to the chosen implementation technology. It mainly consists of the following iterative activities:

- **Technology Evaluation**
  selection of appropriate technological components such as programming language, DBMS, middleware, frameworks, etc.;
- **Solution Elaboration**
  elaboration or adaptation as well as documentation of mechanisms and design patterns such as GUI architecture, object persistency, transaction handling, security, etc. required for implementing the specification;
- **Solution Mapping**
  definition and documentation of the mapping between specification elements and architecture solutions to finally produce the implementation.

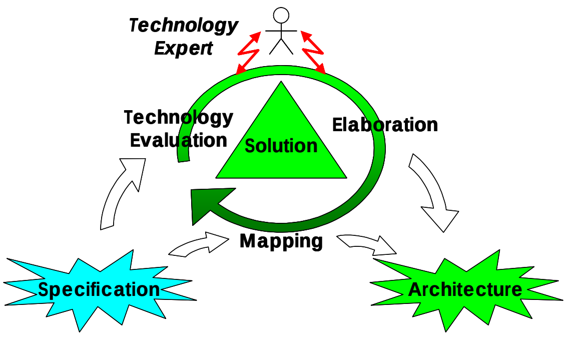


Figure 2: The Design Process

The main deliverable of the design processes is the *Architecture* which is also usually heavily documented using UML. This approach for the transition from specification to implementation is also known as *Recursive Design* or *Design by Mapping* [Me01].

2.3 The Construction Process

Based on the specification and the architecture elaborated as explained above, the *Construction Process* actually consists of the following iterative activities:

- **Component Selection**
  from the specification, i.e. choosing use cases and/or domain objects form implementation;
- **Component Building**
  according to the guidelines defined in the architecture;
- **Component Testing**
  comparing the behavior of the actual implementation with the specification and fixing any differences.

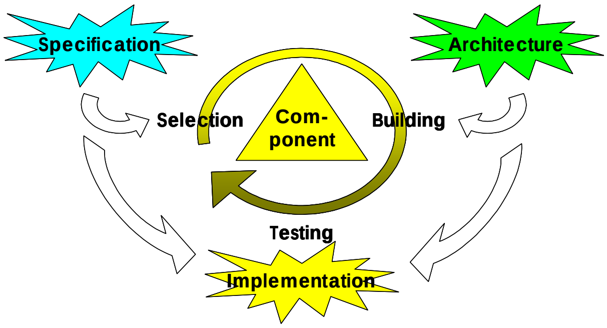


Figure 3: The Construction Process

The main deliverable of the construction process is the actual *Implementation* which basically consists of a tested and running application as well as all elements necessary to rebuild the application. The adherence to the given architecture guarantees a consistent, maintainable and harmonic implementation.

2.4 The Project Management Process

Finally, the overall development process is controlled by the *Project Management Process*, which basically consists of the following iterative activities:

- **Deliverable Definition**
  identification of the results to be produced and delivered;

- **Planning and Delegation**
  identification of dependencies between deliverables and assignment of deliverables to available resources;
- **Progress Monitoring**
  watching the actual progress and quality of deliverables and comparison with the original plan;
- **Control**
  taking corrective actions on deliverables and resources, if necessary.

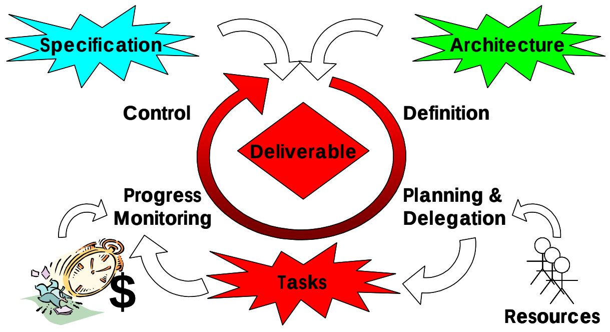


Figure 4: The Project Management Process

3. Requirements for an Automated Coach

Based on the activities described above, one can ask for services that would be helpful to the developer in charge and that could be provided by an automated tool. Such services could either support or simplify an activity or even completely automate it. Among others, the following services have been identified as potential *functional requirements* for CASSANDRA:

- **Activity *Fact Finding*:**
  Asking domain level questions based on analysis patterns in the use case, domain object and behavior models. Building initial use case model fragments from business process model. Building initial domain object model fragments from a use case model.
- **Activity *Fact Representation*:**
  Update the System Model according to a answers given on domain level questions.
- **Activity *Fact Verification*:**
  Simulation of behavior based on stimuli sent to the system model as a black box. Providing a summary of domain facts about the System Model.
- **Activity *Solution Elaboration*:**
  Recommendation of solutions (design patterns) to solve a certain design problem.
- **Activity *Solution Mapping*:**
  Selection of one or more optimal design patterns to a certain area of the specification.

- **Activities *Component Selection & Building*:**
  Automatic code generation based on design patterns.
- **Activity *Component Testing*:**
  Generation of test and integration plans for the overall system. Generation of test specifications for use cases and domain objects.
- **Activity *Deliverable Definition*:**
  Partitioning of a system model into independent subprojects. Identification of results to be delivered and tasks to be done.
- **Activity *Planning & Delegation*:**
  Identification of the next steps to be done in the project. Effort and cost estimation for the tasks to be done.
- **Activity *Progress Monitoring*:**
  Determination of the completion degree of a given system model.

In addition to the functional requirements listed above, the following main *non-functional requirements* have been defined for CASSANDRA:

- **Human touch**
  Working with CASSANDRA should be similar to working with a human coach. This means that it must be very easy to use CASSANDRA (there is usually no user manual available for a human coach), the interaction should be based on natural speech and mimics, i.e. we should be able to discuss with CASSANDRA and she must be responsive.
- **Integration with various CASE and other tools**
  Like a human coach, CASSANDRA should not be limited to a particular CASE tool but she should be able to cope with various CASE and other tools.
- **Extendable architecture**
  Since CASSANDRA is intended as a research platform to explore various new ideas and approaches, it should be easily extendable and provide the highest level of services to the developer of CASSANDRA.

4. CASSANDRA's Anatomy

In order to speed development and reduce development efforts and costs, CASSANDRA has been implemented using the declarative programming language Prolog [CM94], originally developed for artificial intelligence applications. Specifically, CASSANDRA has been built using the WIN-Prolog implementation of Logic Programming Associates Ltd. [LPA00].

The following picture gives a quick overview of the main technological components in CASSANDRA's anatomy:

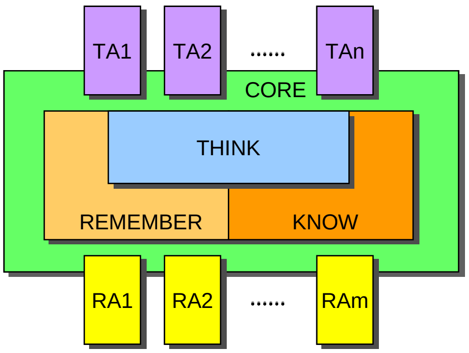


Figure 5: CASSANDRA’s Anatomy

- **CORE**
  Provides basic functionality such as I/O (including GUI, XML- and multi language support), speech output and agent support via MS Agent technology [Mi200], configurability and personalization, persistency, licensing, etc.
- **REMEMBER**
  An active declarative database as a common repository for all project information based on the UML meta model plus some extensions (a software engineering ontology).
- **KNOW**
  Common sense in form of a “domain object model of the world” based on WordNet [Pr98], consisting of about 66’000 classes including their definitions, inheritance hierarchies and associations.
- **THINK**
  A rule-based inference engine to generate suggestions based on project information with the capability of explaining those suggestions and findings.
- **RA1…RAn: Multiple Resource Agents**
  Interface components to adapt various external applications such as CASE tools for bi-directional information exchange and “remote control” of those tools.
- **TA1…TAn: Multiple Task Agents**
  Service components that actually provide CASSANDRA’s (hopefully) valuable services to the users by applying dedicated software engineering know how.

CASSANDRA’s anatomy may be compared to an operating system architecture that provides basic services to various applications (CASSANDRA’s Task Agents) and provides adaptors to various hardware devices (CASSANDRA’s Resource Agents).

5. CASSANDRA’s Current State

CASSANDRA is stable and operational since about one year and has been used by our own consultants in real projects. So far, the following Task Agents have been developed:

- **Review of Use Case Models**
  Reviews a given use case model and spots areas that need clarification or areas that might be better modeled in a different way. It also suggests modeling alternatives and is able to explain these alternatives to the user.
- **Review of Domain Object Models**
  Reviews a given domain object model based on common analysis patterns and raises domain-level questions that need to be clarified. It is not only able to identify obsolete classes and associations, but also missing classes and associations. Furthermore, it may suggests modeling alternatives and is able to explain these alternatives to the user.
- **Building of Domain Object Models**
  Creates an initial domain object model based on a given use case model plus associated descriptions. This is achieved by reading use case names and descriptions and recognizing class names that are common in a given business area as defined in CASSANDRA’s class model of the world (KNOW).
- **Building Data Warehouse Models**
  Creates a multi-dimensional data model (cube design for a data warehouse) from the domain object model of a given operational application. It generates fact tables as well as dimension hierarchies by various levels of denormalization of the domain object model.
- **Effort and Cost Estimation**
  Creates a list of tasks for an IT project based on a specification in a CASE tool plus a series of questions about project characteristics, team characteristics, and the intended implementation technology. Finally, the effort and costs for these tasks are estimated and a rough project plan and increment plan is produced.

Currently, the following Resource Agents are available for CASSANDRA:

- **Rational Rose**
  Supports the CASE tool Rational Rose 98 and Rational Rose 2000.
- **Artisan RTS**
  Supports the CASE tool Artisan Real Time StudioV3.x and V4.x
- **Princeton Select Enterprise**
  Supports the CASE tool Select Enterprise V5.x and V6.x

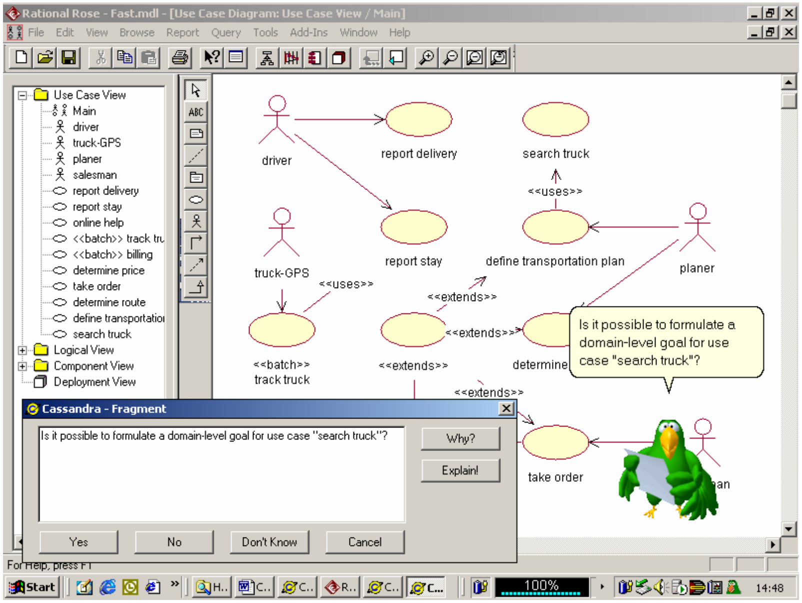
Using these Resource Agents CASSANDRA is able to access information stored in these tools (usually models of the specification) as well as to update this information.

6. Example Usage

To illustrate the style of working with CASSANDRA, in this chapter two short example usage scenarios are shown.

6.1 Scenario 1: Performing a Use Case Model Review

After initially creating a use case model in Rational Rose, an analyst may wish a review of that model. Thus he starts CASSANDRA, selects his project and requests the CASSANDRA service “Review your use case model…”. After a short while, CASSANDRA comes back to the user asking specific questions about the model. Such a question is asked using a conventional dialog plus optionally by a *Microsoft Agent* character such as *Peedy* [Mi00] as shown in the (figure below).


Cassandra - Fragment

Is it possible to formulate a domain-level goal for use case "search truck"?

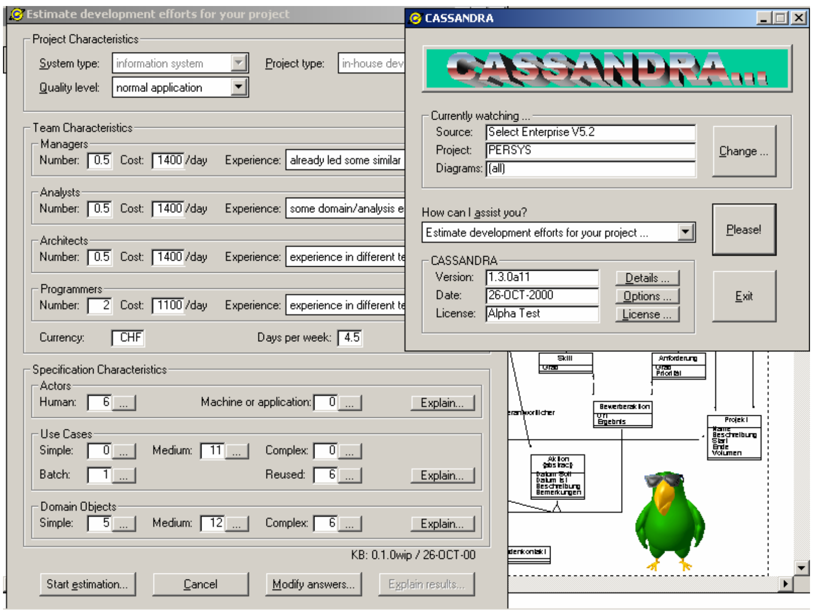
Is it possible to formulate a domain-level goal for use case "search truck"?

Figure 6: CASSANDRA discussing about a Use Case Model

The analyst may then either answer that question or request further explanations. These interactions between the analyst and CASSANDRA continue for the whole use case model. Finally, at the end of this discussion, CASSANDRA suggests a To-Do list depending on the answers given by the analyst.

6.2 Scenario 2: Performing an Effort Estimation

In the second usage scenario it is assumed that an initial version of a specification for an IT solution has been defined and that it has been modeled using the CASE tool Select Enterprise. The project manager starts CASSANDRA, selects the project and requests the CASSANDRA service “Estimate development effort for your project…”. CASSANDRA then reads all project information from the CASE tool and presents the estimation main dialog as shown below.



Estimate development efforts for your project

Project Characteristics
System type: information system   Project type: in-house dev
Quality level: normal application

Team Characteristics
Managers
Number: 0.5 Cost: 1400 /day Experience: already led some similar
Analysts
Number: 0.5 Cost: 1400 /day Experience: some domain/analysis e
Architects
Number: 0.5 Cost: 1400 /day Experience: experience in different te
Programmers
Number: 2 Cost: 1100 /day Experience: experience in different te
Currency: CHF   Days per week: 4.5

Specification Characteristics
Actors
Human: 6   Machine or application: 0
Use Cases
Simple: 0   Medium: 11   Complex: 0
Batch: 1   Reused: 6
Domain Objects
Simple: 5   Medium: 12   Complex: 6

KB: 0.1.0wip / 26-OCT-00

CASSANDRA

Currently watching ...
Source: Select Enterprise V5.2
Project: PERSYS
Diagrams: (all)

How can I assist you?
Estimate development efforts for your project ...

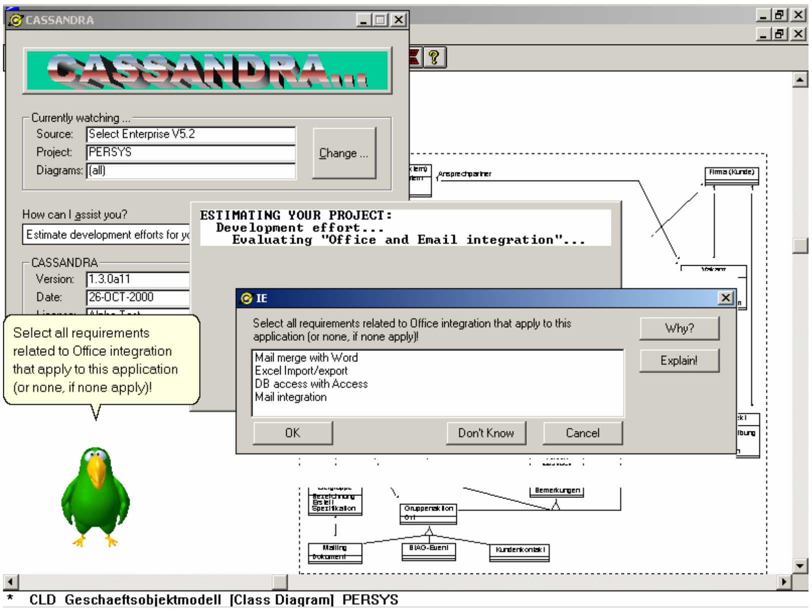
CASSANDRA
Version: 1.3.0a11
Date: 26-OCT-2000
License: Alpha Test

Figure 7: CASSANDRA’s Main Dialog for Effort Estimation

The *Specification Characteristics* are already filled out based on the specification read from the CASE tool. The project manager may now complete the dialog by specifying the *Team Characteristics* and other information. Finally he starts the main estimation process which causes CASSANDRA to ask additional questions as necessary (see figure below). Again, the user has the possibility to answer those questions directly or to request further explanations.

During the estimation process, CASSANDRA asks a whole series of questions about the project situation, special requirements or the intended implementation technology. These questions occur not in a strict sequence, but depend on answers previously given. For example, when the user said that the basic architecture is a 2-Tier architecture, CASSANDRA never asks for separate programming languages on Client and Server side.

Furthermore, if the project manager wants to estimate the same project some days or weeks later, for example because some additional requirements where discovered, CASSANDRA remembers the answers given by the user in the previous session and usually doesn’t ask the same questions again.



Currently watching ...
Source: Select Enterprise V5.2
Project: PERSYS
Diagrams: (all)

How can I assist you?
Estimate development efforts for yo

CASSANDRA
Version: 1.3.0a11
Date: 26-OCT-2000

```
ESTIMATING YOUR PROJECT:
  Development effort...
    Evaluating "Office and Email integration"...
```

Select all requirements related to Office integration that apply to this application (or none, if none apply)!

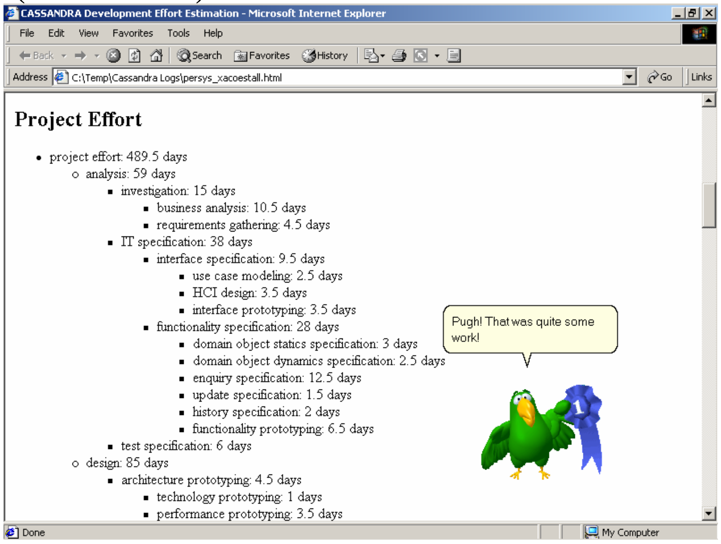
Mail merge with Word
Excel Import/export
DB access with Access
Mail integration

Select all requirements related to Office integration that apply to this application (or none, if none apply)!

* CLD Geschaeftsobjektmodell [Class Diagram] PERSYS

Figure 8: An additional Question during the Estimation Process

Finally, CASSANDRA presents a detailed estimation of individual activities. This result can be generated in various formats (plain text, HTML, or XML) and displayed in various viewers such as CASSANDRA's internal viewer, an external text editor, or a WEB browser (as shown below).


CASSANDRA Development Effort Estimation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\Temp\Cassandra Logs\persys_xacoestall.html
Go Links
Done
My Computer

Project Effort

- project effort: 489.5 days
  - analysis: 59 days
    - investigation: 15 days
      - business analysis: 10.5 days
      - requirements gathering: 4.5 days
    - IT specification: 38 days
      - interface specification: 9.5 days
        - use case modeling: 2.5 days
        - HCI design: 3.5 days
        - interface prototyping: 3.5 days
      - functionality specification: 28 days
        - domain object statics specification: 3 days
        - domain object dynamics specification: 2.5 days
        - enquiry specification: 12.5 days
        - update specification: 1.5 days
        - history specification: 2 days
        - functionality prototyping: 6.5 days
    - test specification: 6 days
  - design: 85 days
    - architecture prototyping: 4.5 days
      - technology prototyping: 1 days
      - performance prototyping: 3.5 days

Pugh! That was quite some work!

Figure 9: Results of an Estimation in HTML

7. Summary and Next Steps

CASSANDRA is an innovative software engineering application that tries to emulate the capabilities of a human software engineering consultant. Specifically, CASSANDRA

- provides various advanced software engineering services to the user,
- is able to cope with various CASE tools,
- (inter-)acts with the user using domain-level discussions like a human consultant.

Currently, CASSANDRA focuses on the early requirements gathering phases in the development process. This is an area not very well covered by any other computer-based tool. Some of the next Task Agents that will be implemented are *Project Risk Analysis* and *Specification Simulation* (i.e. providing an executable specification in UML). Later, it is intended to provide services that also support design and construction activities. On the Resource Agent side, it is planned to provide interfaces to the CASE tool *Rhapsody* from I-Logix and to *Microsoft Word* and *Microsoft Project* in the immediate future.

However, CASSANDRA is still a software engineering research project. Currently, it is mainly used by our own consultants to simplify their jobs working on real projects. CASSANDRA is published as a public domain product on our web site [Kn00]. Another application of CASSANDRA that turned out to be very useful is software engineering education. We provided CASSANDRA as an aid during our UML/CASE courses. The feedback from participants of the modeling insights they gained when using CASSANDRA where really encouraging.

The ultimate vision for CASSANDRA is that it may some day replace our human consultants completely. We could then send CASSANDRA to help our customers with their projects instead of going by ourselves. The only question that remains then is: How do <u>we</u> earn our money?

8. References

[CM94] W. F. Clocksin, C. S. Mellish: Programming in Prolog, Springer Verlag, 1994
[Kn97] KnowGravity Inc.: Software Engineering course material, 1997-2000
[Kn01] Web site of KnowGravity Inc: www.knowgravity.com
[Pr98] WordNet - a Lexical Database for English, Cognitive Science Laboratory, Princeton University, www.cogsci.princeton.edu/~wn/ , 1998
[LPA00] Logic Programming Associates Ltd.: WIN-PROLOG (user documentation) , V4.040, 2000
[Mi00] Microsoft Corp.: Microsoft Agent, http://msdn.microsoft.com/msagent, 2000
[OMG99] OMG: Unified Modeling Language Specification, Version 1.3, June 1999
[Me01] Stephen Mellor: Recursive Design, Prentice Hall, 2001