UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

SOFTWARE AGENTS FOR PROJECTS IDENTIFICATION – The case of CDF Projects

BY

MWANGI JOSEPHINE WAMBUI

P56/8486/2004

Supervisor

Dr. Wanjiku Ng'ang'a

July 2009

Submitted in partial fulfillment of the requirements of the Master of Science in Applied Computer Science

DECLARATION

This research project, as presented in this report, is my original work and has not been presented for any other university award.

Name: **Josephine W. Mwangi**

Reg-no: P56/8486/2004

Signature: .....

Date: 11th August 2009

Supervised by:

This research project has been submitted as part of fulfillment of the requirements for Masters of Science in Applied Computer Science of the University of Nairobi with my approval as the University supervisor.

Name: **Dr. Wanjiku Ng'ang'a.**

Signature: .....

Date: 11th August, 2009

ABSTRACT

This paper describes and reports on a prototypical implementation of a Multi-agent system for Constituency Development Fund (CDF) Projects Identification. Agents as representatives of various stakeholders have the task of availing the projects requirements/preferences and offers and performing the necessary updates of the system data.

The approach proposed uses the semantic web languages and tools; Rule Markup Language (RuleML) to specify and publish projects requirements and preferences as per the CDF Act 2003, as the basis for agent knowledge base and Resource Description Framework (RDF) to represent the projects offerings. The system also contains a persistent storage of the Constituencies data and the yearly CDF Allocations in a MySQL Database. The use of semantic language is motivated by the need to increase openness and interoperability among agents.

DEDICATION

Special dedication goes to my beloved son, Caleb Ndung'u and my wonderful husband, Paul Ndung'u.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my supervisors, Dr. Wanjiku Ng'ang'a and Dr. Katherine Getao, for their invaluable guidance without which this research project would not be accomplished.

Special thanks to the SCI Lecturers who encouraged me to work on during the entire MSc Programme period. I saw your sincere concern to see me through, I'm forever grateful.

To my sponsors, VLIR scholarship programme, University of Nairobi and specifically Prof. Kenneth Mavuti, many thanks for your financial support.

Special mention to Paul Ndung'u Kariuki, my loving husband, for the company and moral support he gave as we worked together on our research projects and the persistent pressure to complete the work. My son Caleb, you understood my absence from home, as I worked for long hours.

I am indebted to my friends, colleagues, and students for their help, support, and teamwork over the past five years.

Finally to the Almighty God, he gave me the strength and hope when I felt like giving up.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

- ACCs - AIDS Control Committees
- ACLs - Agent Communication Languages
- AIP - Agent Interaction Protocol
- AMS - Agent Management System
- CDC - Constituencies Development Committee
- CDF - Constituency Development Funds
- CDF Board - Constituencies Development Fund Board
- CFC - Constituencies Fund Committee
- DAML - DARPA Agent Markup Language
- DF - Directory Facilitator
- DOM- Document Object Model
- DPC- District Projects Committee
- DTD - Document Type Definition
- FIPA - Foundation for intelligent Physical Agents
- FPE - Free Primary Education
- GUI - Graphical user interface
- JADE -Java Agent Development Framework
- JSP - JavaServer Pages
- KIPPRA - Kenya Institute for Public Policy Research and Analysis
- KQML- Knowledge Query Manipulation Languages
- KSE - Knowledge Sharing Effort
- LAs - Local authorities
- LATF - Local Authority Transfer Fund
- LDL - Logical Deduction Language
- MTS - Message Transport System
- NACC - National Aids Control Council
- NAF – Negation as Failure
- NASSEP - National Sample Survey and Evaluation Program
- RDF - Resource Description Framework
- RDFS - RDF Schema

- RMI - Remote Method Invocation
- REPLF -Rural Electrification Programme Levy Fund
- RMA- Remote Monitoring Agent
- RMLF - Road Maintenance Levy Fund
- OWL - Web Ontology Language
- SEBF - Secondary School Education Bursary Fund
- TILAB - Telecom Italia Labs
- UML - Unified Modeling Language
- URI – Universal Resource Indicator
- XML - Extensible Markup Language
- XSL - Extensible Stylesheet Language
- XSLT - XSL Transformations

1 INTRODUCTION

1.1 Background

A project is a series of activities aimed at bringing about clearly specified objectives within a defined time-period and with a defined budget. Projects typically have identifiable phases and each phase has a unique set of challenges for the project manager. All projects whether lasting for a few months or spreading for many years, need proper management in order to achieve their objectives.

The first of these phases is the Identification or initiation stage, where the need is identified. Need identification phase characterizes the stakeholders becoming aware of some unmet need. It results when there is a difference between a desired state and an actual state. It is also called Problem Recognition according to J. Engel and R. Blackwell model (1982). It is in this phase where issues of feasibility and justification are addressed. Achieving this calls for projects clearly identified stakeholders, including the final beneficiaries among other things.

A good project analysis ensures that;

- The project is relevant to an agreed strategy and to the real problems of target groups/beneficiaries
- Projects are feasible, meaning that objectives can be realistically achieved within the constraints of the operating environment and capabilities of the implementing agencies
- Benefits generated by the projects are likely to be sustainable.

Achieving the above-mentioned goals in analyzing projects requires that a number of people collaborate with one another in order to share their knowledge, resources and capabilities.

It is from this analytical stage that cost, schedule, functionality, and quality targets are set. A project manager is thence required to run the project efficiently; act as an arbiter of the differing objectives that will inevitably exist within and across the team (Heerkens, 2002). These vary from project to project and team to team.

Managing this collaborative environment is a complex task and it is the most important role that a project manager plays, which depends on many parameters.

The Constituency Development Fund (CDF) projects are not an exception to these above mentioned challenges. To understand these in the CDF projects perspective, a little background suffices.

According to CDF Act 2003 [35], Section 50 states that Constituency Development Fund (CDF) projects are complementary to other normal government or any other agency's development initiatives. There is therefore a need to harmonize CDF projects with other development initiatives at the district level and the country at large. An appropriate balance is therefore needed between the Government's policy priorities and those of the beneficiaries. This requires proper project identification. Identification strives to identify project ideas that are consistent with implementers' development priorities and assess the relevance and likely feasibility of these project ideas.

There exist almost identical definitions to the term constituency. According to [42] a constituency is defined as a group of citizens or voters in a geographic district who may elect a representative to a legislative or other body of government. [45] Defines a Constituency as a division of a larger election area into several, approximately equal large subsections. Another definition states a constituency as a geographical area of the country represented by a Member of Parliament. In this write up, a constituency refers to all the three terms.

Constituency Development Fund, in the Kenyan context, which is the basis for this research, is a fund that aims to control imbalances in regional development brought about by partisan politics.

- It targets all constituency-level development projects, particularly those aiming to combat poverty at the grassroots.

According to the Constituency Development Fund (CDF) Act (2003) [21], the Constituencies Development Fund Board[1] requires that Identified Projects be submitted in a list according to priority. These projects proposals are submitted from the various locations in a constituency. Each and every location (grassroots' communities) strongly believes that their projects interest should be given priority even when the funds are ceiled year by year. This is unachievable and hence only a few projects are approved for implementation.

---

[1] *A national committee that receives/reviews project proposals submitted from various constituencies and approves them for funding.*

Well structured and justifiable ways of prioritizing to promote transparency and accountability are needed by the Constituency Development Committee[2]. This Committee is faced with the dilemma of choosing between multiple projects, some of seemingly equal economic value, a choice which can only be facilitated through systematic and realistic project studies to clearly establish priorities. These studies should help in identifying and compiling limited lists of the probable projects likely to be selected.

It must then be ensured, through the mechanism of project evaluation that projects marked out for implementation are of the highest priority or urgency from both implementers and beneficiaries' viewpoint. This confirms that identification is a critical phase and for good results, a lot of time and effort has to be put into it.

The focus of this thesis is to investigate how software agents can be used to develop a model that can intelligently support CDF projects identification. It uses logic-based, agent-based and semantic web-based technologies. What follows is a description of CDF projects, the motivation for and contribution of this work.

1.2 Project Specification and Justification

According to baseline survey [25] findings gathered through focus group discussions by the Kenya Institute for Public Policy Research and Analysis (KIPPRA[3]) between February and April 2006, quite a number of challenges are being faced by this decentralized fund. This survey covered 35 constituencies in eight (8) pilot districts (Bondo, Bungoma, Nakuru, Machakos, Kirinyaga, Wajir, Mombasa and Nairobi).

The baseline survey comprised 3 components:-

- household surveys- KIPPRA interviewed a total of 4,423 households drawn from the NASSEP IV sampling frame of the Central Bureau of Statistics)
- Key informant surveys
- Focus group discussions

---

[2] *Committee constituted and convened by the elected Member of Parliament for every constituency that deliberate on project proposals from all the locations in the constituency*

[3] *Kenya Institute for Public Policy Research and Analysis (KIPPRA) is an autonomous public institute whose primary mission is to provide quality public policy advice to the government of Kenya and to the private sector in order to contribute to achievement of national development goals.*

This survey aimed at:-

- Ascertaining the level of public awareness of the decentralized funds as outlined earlier in this write up
- investigating the degree of community involvement in administration of each fund;
- Obtaining suggestions on how the coordination and effectiveness of the various the funds can be improved.

1.2.1 Survey Findings

The survey focused on the 7 decentralized funds in Kenya, *funds detailed in chapter 2*, but of interest to this thesis are the findings on CDF funds.

Impact

Respondents indicated:-

Table 1: CDF Funds Impact Survey

Negative impact	No impact.	Positive impact.
12.3%	48.8%	38.9 %

Participation

Statistics on participation rating particularly in decision-making processes are:-

Table 2: CDF Participation Survey

Level of Participation	**Percentages**		
	Not involved	**Not stated**	**Involved**
Receiving information from Government officials, MPs and other sources	62.6%	4.5%	32.8%
Listening and getting information at Meetings, Writings	75.9%	4.7%	19.5%

Working Groups and Meetings to discuss issues	85.9%	4.7%	9.4%
Through analysis and setting agenda via Multi-stakeholders groups	91.8%	4.6%	3.7%
Attending and actively engaging at district or constituency level meetings	91.9%	4.6%	3.5%
Involved in making decisions or matters relating to the funds	91.0%	4.8%	4.2%

Accountability and Performance

Respondents were asked to agree or disagree with the statement that decision for the CDF funds are taken within the funds mandate; in other words, whether the fund managers are using the funds for the purpose intended.

Table 3: Accountability and Performance survey

No opinion	Do not know	Agree	Disagree
15%	35%	15%	35%

Awareness regarding whether decisions taken are within the mandates of the CDF funds is relatively low with most of the respondents stating that they do not know. Significantly, more than double the number of respondents disagrees than agree that the CDF funds operate within their mandate, indicating the generally high levels of distrust in fund managers.

Decisions Taken are sufficiently Justified

Table 4: Justification Level survey

No opinion	Disagree	Do not know	Agree	Not stated
12%	46%	25%	15%	1%

Those that agree that the decisions are sufficiently justified are only 15%. This data shows large dissatisfaction in the probity of decision-making with 46% indicating that the fund decisions are not sufficiently justified.

1.2.2 Summary of the Key Public Perceptions

What follows is a summary of some of the key public perceptions in relation to the various funds that motivate research in this area with a specific focus on CDF projects.

Governance

Political loyalties have led to unfair sharing of resources across constituencies/wards. In addition, there is a general lack of transparency and accountability probably due to the blending of supervisory and implementing roles.

Implementation

Poor awareness by community members and fund managers of their roles and responsibilities in the governance of funds has contributed to poor performance and in some cases a complete failure of the funds utilization. Poor participation, particularly by marginalized groups, results in poor prioritization of projects and exclusion. No mechanisms exist to deal with projects such as roads, water systems, and schools that may cut across constituencies entailing shared benefits. No clear mechanisms exist to avert duplication of functions. Both CDF and the Ministry of Education offer education bursaries. There are also reported instances of a single project claiming support from different funds, with no checks to prevent 'double' accounting. Finally, there are challenges ensuring that all decentralized funds reach all parts of the district or constituency in adequate quantities, and that all funds allocated are actually utilized instead of being returned to the source.

In the conclusion made by the KIPPRA survey, they mention a need to train the managers of the fund and community organizations on the procedures for utilization of the fund. Secondly, new regulations and restructuring of the current funds are necessary to ensure that the funds meet the needs of the targeted beneficiaries.

The CDF Act being one of the most credible and critical legislations passed by the Kenyan Parliament in the recent past, a solution therefore is needed in aid of projects identification. It must address the resolution of the various stakeholders' diverse interests; benefits/conflicts appropriately and hence justify the criteria for distributing funds. It has enabled Kenyans to

experience the value of government money and the common man and woman can now directly take part in deciding on development matters for their area.

In line with the KIPPRA conclusion, this work proposes an agent-based platform that serves to integrate the different views, interest and even conflicts, help in problem and strategy analysis. This is done by a community of interacting agents. Each Agent represents a distinct target group in the project and is capable of exchanging messages and taking roles of the entities they represent.

The main roles identified in this work are Project Requesters (CDF Committee), Projects Providers (Local Community) and brokers all represented by software agents. The preferences/ requirements of the requesters are represented in a logical language using rules and priorities. The Project proposals are represented in a certain semi-structured format using the Semantic Web standard language. The broker Agent has a special knowledge both for the declarative language and the advertisement format

This will promote participation, effective coordination between these existing groups/options and hence proper utilization procedures.

1.3 Study Objectives and Key Questions

The focus of this thesis is to:-

- Study challenges that exist with participatory identification of projects
- Investigate what engineering methodologies can be used for developing applications that intelligently support project identification; in particular the software agents
  - Define how a participatory stake holder's interface can be modeled
  - Experiment with different semantic web standards in the architecture
  - Develop a semantically rich model

1.3.1 Key Questions

- Which technologies, protocols and systems are available and accurate for Project Identification?
- How can we provide submission and access to different project definitions in a semantically rich information space?
- How can we use intelligent agents to implement project identification framework specifically brokering?
- What semantic web standards are applicable to this model?

1.4 Organization/Structure of the Thesis

This work is articulated as follows:

Chapter 2 provides a background theory and related work. More specifically, definitions of the terms "intelligent agent" and "semantic web" are given each with a provision to gain better insight into the issues involved. Consequently, a detailed presentation of *defeasible logic*, which is the knowledge representation framework of this work, regarding brokering. A definition of the term "brokering" with its basic techniques and requirements then follows. Subsequently, a presentation of some selected work in this field is done. The next part of the chapter presents a justification of the eligibility of defeasible logic for expressing user preferences in brokering cases. Then, the semantic web tower is discussed placing it in the agent technology currently. A justification of the use of software agents is then made in comparison with other engineering methodologies.

Chapter 3 is a presentation of the methodology used for this work. This entails the analysis and design steps adopted together with the objects/diagrams produced in every step.

Chapter 4 flows from *chapter 3* since the deliverables are directly applied for implementation as obtained without any alterations. The artifacts produced previously are tailored for JADE platform implementation which is well described in chapter 4. The implementation is discussed as follows:-the overall architecture of the system and description of all the modules with their individual specifications.

Chapter 5 reports on the testing verification and results of the system. It details the testing approaches and criteria, the performance measures all used in this work. Sample results are also illustrated that are produced with various test cases. Chapter 6 is a discussion of the conclusions made and the proposals thereof for possible future works. A few limitations were experienced which are also described in this chapter.

Finally, Samples of codes used both for project offers and rules are included in the appendices.

2 LITERATURE REVIEW

2.1 Background Information

The Kenya government has a number of decentralized funds established based on the belief that government at the local level has a better understanding of community needs and is more capable of delivering improved, responsive and relevant services. Specifically, decentralized funds, also known as social or public funds, are established to:

- Increase community participation in decision-making where local affairs are concerned;
- Enhance government transparency as more people become aware of (and involved with) these funds; and
- Speed up government's responsiveness and improve the quality of its service delivery.

There are currently 7 operational decentralized funds, which generally aim at reducing socio-economic disparities and improving the well being of citizens.

They include:-

1. **Secondary School Education Bursary Fund (SEBF)**

The Secondary School Education Bursary Fund (SEBF) was established in 1993/4 through a Presidential pronouncement. SEBF aims to cushion the country's poor and vulnerable groups against the high and increasing cost of secondary education, therefore reducing inequalities.

2. **Road Maintenance Levy Fund (RMLF)**

The Road Maintenance Levy Fund (RMLF) was established in 1993 through the Road Maintenance Levy Fund Act. RMLF caters for the maintenance of public roads, including local authority unclassified roads. The fund is made up from a fuel levy on petroleum products and transit toll collections. It is administered by the Kenya Roads Board, which was established in 1999.

3. **Rural Electrification Programme Levy Fund (REPLF)**

The Rural Electrification Programme Levy Fund (REPLF) was established in 1998 through sections 129 & 130 of the Electric Power Act (1997). The fund aims to finance electrification of rural and other underserved areas. It is used for programmes relating to

the design, construction, equipping and operation & maintenance of rural electrification projects identified by communities.

4. Local Authority Transfer Fund (LATF)

The Local Authority Transfer Fund (LATF) was established in 1999 through the LATF Act No. 8 of 1998, with the objective of improving service delivery, improving financial management, and reducing the outstanding debt of local authorities (LAs). LATF, which comprises 5% of the national income tax collection in any year, currently makes up approximately 24% of local authority revenues.

5. HIV/AIDS Fund

The HIV/AIDS fund was established in 1999 through a Presidential order contained in Legal Notice No. 170. Its establishment coincided with the declaration of HIV/AIDS as a national disaster, formation of the National Aids Control Council (NACC) and the AIDS Control Committees (ACCs). The fund targets individuals infected with and affected by HIV/AIDS, with the focus being on long-term care and support.

6. Free Primary Education (FPE)

FPE was established in January 2003 through the government's National Alliance Rainbow Coalition (NARC) manifesto. The fund aims to address financing and quality challenges in primary schooling. It targets all Kenyan children attending formal and non-formal public schools. Emphasis is however directed towards children from poor households.

7. Constituency Development Fund (CDF)

The Constituencies Development Fund (CDF) Act was passed into law in 2003 by National Assembly. CDF is a fund that aims to control imbalances in regional development brought about by partisan politics. It targets all constituency-level development projects, particularly those aiming to combat poverty at the grassroots.

2.1.1 Justification of the CDF Case Study

The focus of this thesis work is on Constituency Development Fund for the following reasons:-

- The fund is not established on any particular development area and hence gives a broad base of testing the model.
- It is well established and its guidelines and structures are well documented in the CDF Act 2003.
- According to KIPPRA survey (2006) its awareness is relatively high compared to others and ranks third after the educational funds (Free Primary Education and Secondary School Education Bursary Fund). This is a precursor to effective public participation and hence better project identification

Key Highlights of the CDF Act 2003

The CDF initiative is intended to uplift the living standards of the Kenyan people at the grass root level. The fund was designed to enable balanced development across the country. The CDF fund has three prime objectives:

- To fund projects with immediate social and economic impact in order to uplift the lives of the people.
- To alleviate poverty
- For purposes of development and in particular in the fight against poverty at the constituency level.

The CDF Act is one of the most credible and critical legislations passed by the Kenyan Parliament in the recent past. It has enabled Kenyans to experience the value of government money and the common man can now directly take part in deciding on development matters for the area. The criterion for distributing funds is on the basis of poverty index and population size.

CDF is managed through 4 committees, 2 of which are at the national level and 2 at the grassroots level.

Legal Framework of the CDF

Established in 2003, the CDF Act provides that at least 2.5% of all collected ordinary government revenue in every financial year shall be paid into the Fund. This amount shall be disbursed under the direction of Constituencies Development Fund Board (CDF Board)

constituted as per Section 5 of the Act. Seventy-five percent of the amount is disbursed equally across all the 210 constituencies while the remaining 25% is disbursed on the basis of the poverty index (Section 19 of the CDF Act). At the constituency level, the CDF Act specifies that up to a maximum of 3% of the total annual allocation may be used on office running expenses, 5% shall be set aside for emergency while not more than 10% shall be allocated to the education bursary scheme annually.

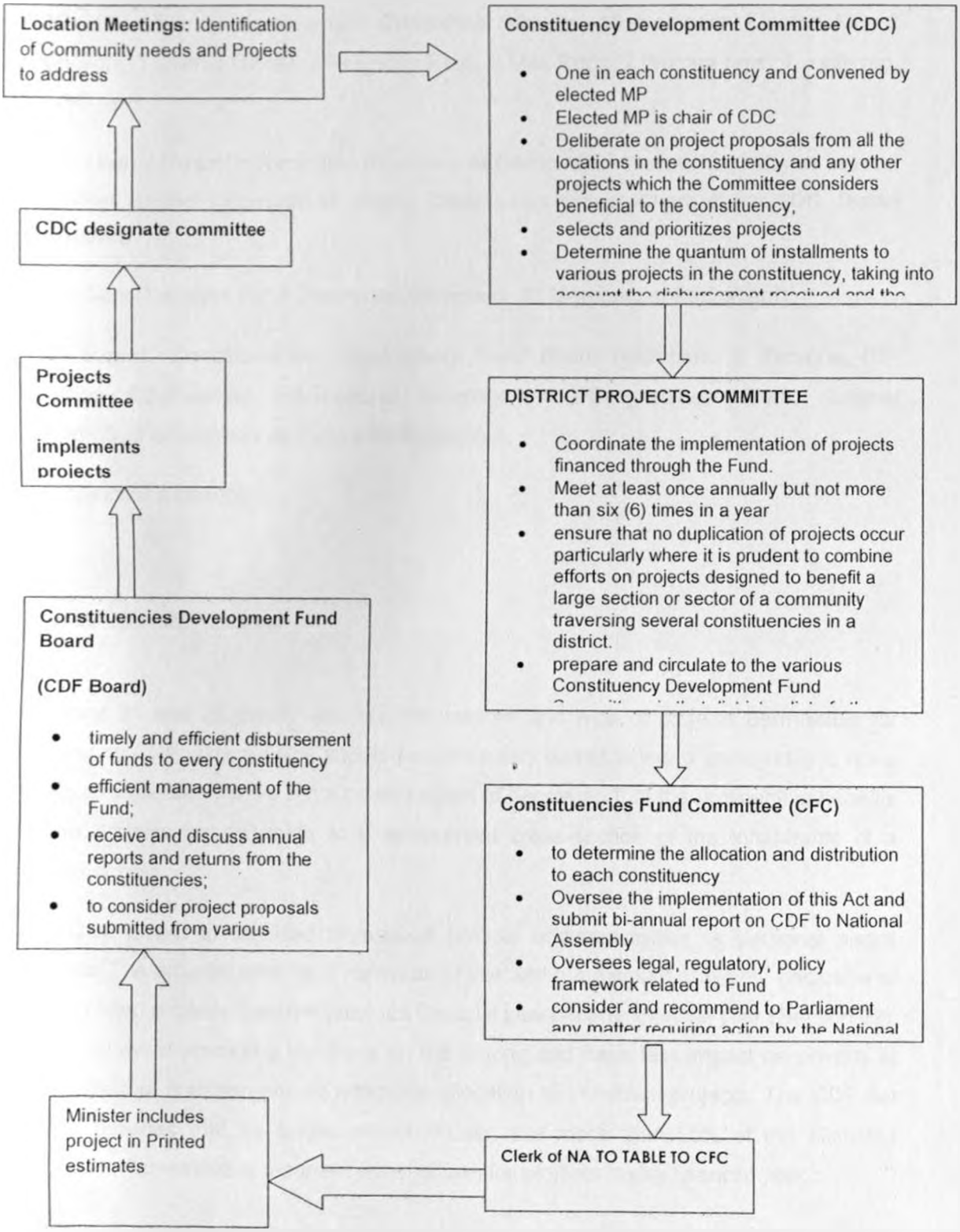
A sitting MP is not a signatory to the CDF bank account but convenes the CDF Committee in her/his constituency. CDF project proposals are submitted to MPs who in turn forward them to the Clerk of the National Assembly. The approved project list is reviewed by the National CDF committee, which presents final recommendation to the Finance Minister. All unutilized funds shall remain in the constituency account.

The Act further provides for the formation procedure and the operational structures to oversee the implementation of the fund. The Act also provides for how the CDF projects shall be identified, the number and type of projects to be funded.

Operational Structures in the Fund

The CDF Act provides for management structures to be put in place to oversee the implementation of CDF projects and more importantly to ensure prudent utilization of the funds. These structures include the Constituencies Development Fund Committee (Section 27 of the CDF Act), Constituencies Development Fund Board (CDF Board) - Section 5, District Projects Committee (Section 39 of the CDF Act) and the Constituencies Development Committee (CDC) – Section 23.1.

Figure 1: CDF Project Cycle



CDF Project Cycle

Key:

CDC: *Constituency Development Committee* (Max. of 15 members: Elected MP, 2 councilors, 1 District Officer, 2 Religious Reps, 2 Men Reps, 2 Women reps, 1 youth rep, 1 NGO rep).

DPC: *District Projects Committee* (Members: All District MPs, Chairmen and mayors of Local Authorities, District Commissioner, District Development Officer, Chairs of the CDC, District Accountant)

CFC: *Constituencies Fund Committee* (Members: 11 Members of Parliament)

CDF Board: *Constituencies Development Fund Board* (Members: 8 Persons, PS-Finance, PS-Planning, PS-Regional Development, PS-Agriculture, Clerk, National Assembly, 8 appointees as Fund Administrators)

NA: *National Assembly*

Project Eligibility Criteria

Sections 21 and 22 clearly sets out the number and type of projects permissible for funding by CDF. The projects should be community based where a community is not a group of organized people but a cross-section of people so that the prospective benefits of the projects are available to a widespread cross-section of the inhabitants of a particular area.

The CDF is not to be used to support political bodies/activities or personal award projects. The projects shall be a minimum of five and a maximum of twenty (inclusive of uncompleted projects from the previous financial year) in any financial year (Section 22). This is to avoid spreading too thinly on the ground and have less impact on poverty at the grass root and also ensure adequate allocation to individual projects. The CDF Act further provides that no single project should take more than 50% of the allocated amount (which excludes recurrent expenditure) for projects in any financial year.

Section 26 provides for counter part funding if the co-financier has no objection and the part funding goes to a defined unit or phase of the project in order to ensure that the

particular portion defined in the allocation is completed with the funds allocated under the Act.

Projects Identification

Section 23(2, 3 & 4) provides for how to identify projects. The Act requires that location meetings be held and other forums used to select projects before submitting to the CDC onward transmission for funding.

Section 50 states that CDF projects are complementary to other normal government or any other agency's development initiatives. Hence, the need to harmonize CDF projects with other development initiatives at the district level.

2.2 Current Methods being used for project identification

2.2.1 Overview

During Project Identification, the first stage of project design, the project concept is analyzed and elaborated. The practitioners ask themselves the following four questions:

1. What is the overall rationale and objectives for the project, and do the stakeholders pursuing or affected by the results of this effort truly "own" the project concept?
2. Is the project relevant? If so, for whom? The Government? The potential beneficiaries?
3. Can the project be effective, efficient and implementable? What potential pre-feasibility impediments present themselves, and how do these factors influence the potential for success of the effort?
4. All things considered, is it advisable to allocate additional resources in order to move forward with a more detailed project design effort – specifically to proceed to the project preparation stage?

Answering these questions encourages the practitioner to rigorously define and clearly state the intended "end results" of the project under consideration. Measures of ownership, relevance, effectiveness, efficiency and sustainability should be designed into the identified project in order to ensure that the project can accomplish its intended

results within acceptable levels of risks to the public, the sponsors, and the project manager.

2.2.2 Tools in Perspective

Detailed below is a subset of tools used in the Project Identification stage.

Problem Tree

A properly planned project addresses the needs of multiple stakeholders or constituencies. Project identification commences with the delineation of the key problems that exist in a given context. A Problem Tree is a diagram that summarizes project problems in hierarchical order, establishing "cause and effect" relationships between them.

The first step in developing a Problem Tree is to identify perceived problems related to a project concept. Then, "cause and effect" logical thinking is used to identify and diagram the relationships between the various problems. The diagramming methodology is as follows:

- If the problem is a cause, it goes on the level below;
- If the problem is an effect, it goes on the level above;
- If it is neither a cause, nor an effect, it goes at the same level.

Once problems are arranged in a hierarchical order, arrows are added to show the cause and effect relationships. Where multiple problems cause convergence around one main problematic effect in the middle of the chart, it is possible to identify a "core problem" related to the project concept. The core problem becomes the focal problem for the project, and can be addressed by various alternative solutions as demonstrated in the Alternatives Tree tool.

Alternatives Tree

Through the Problem Tree process, key problems are identified while ancillary or spurious concerns are left to the side. There is typically more than one way to address a project's "core problem" and achieve a desired outcome. Carefully delineating alternative approaches to a problem can often result in finding a more timely and cost effective solution while still delivering the same or higher impact results. An appropriate tool for analyzing various options for addressing a project's core problem is the Alternatives Tree. The Alternatives Tree is a tool that:

- Is targeted at providing alternative solutions to "core problem" identified in a problem tree;

- Relates alternative project "Outputs" to a core problem solution or the "Project Purpose" in the form of "if-then" hypothesis statements;
- Relates the "Project Purpose" to stakeholder interests or "Project Goals;"
- Is based on strategic considerations of a project including stakeholder interests and problems; and,
- Provides a basis for comparing, ranking and combining alternatives helping to choose the "most fundable" project.

An easy way to identify a project's purpose is to convert the core problem identified with the Problem Tree into a positive objective statement. This objective statement then forms the focal point for generating alternative solutions (or combinations of solutions) for accomplishing the purpose. In this manner, the Alternatives Tree identifies the range of potential solutions available to the project manager for addressing stakeholder problems and issues.

Stakeholder Analysis

Stakeholder Analysis is employed to identify the interests or "stakes" of key constituents and to gain the ownership and commitment of stakeholders groups for a project concept. The purpose of stakeholder analysis is to indicate whose interests should be taken into account when making a decision (Crosby B.-1992). Stakeholders are organizations, groups, or individuals who exhibit qualities as follows:

- Have an interest in short and long term success (or failure) of a project;
- Contribute to or are affected by some part of a project as gainers or losers; or
- Can directly or indirectly influence the design and implementation of a project.

During the Project Identification stage, stakeholders can play an important role by:

- Providing guidance on the problems and needs that a project should address;
- Providing support for required information gathering and analysis;
- Making resources available for proposal preparation and presentations; and,

By ignoring stakeholder interests and needs, opposition can build against the proposed project resulting in project design and implementation delays, in cost overruns and in sub-standard performance. Satisfaction of stakeholders' needs is paramount to gaining their commitment, both in terms of rhetoric and resources.

2.3 *Intelligent agents, their architectures, frameworks and any other related information*

2.3.1 *Intelligent Agent*

Agent-based technology provides a perfect means for automating the process of brokering. Quite a number of Agent definitions exist. Roberto A. [*Towards a Standardization of Multi-Agent System Frameworks*] gives a nice definition of an agent. "Agent is an interactive entity that exists as part of an environment shared with other agents, which can perceive and act in a proactive or reactive manner, based on shared knowledge of communication and representation".

Some of the most important attributes, which differentiate agents from other programs, are among others:

- Adaptivity is the ability to learn and improve with experience.
- Autonomy is goal-directedness, proactive and self-starting behaviour.
- Collaborative Behaviour is the ability to work with other agents to achieve a common goal.
- Mobility is the ability to migrate in a self-directed way from one platform host to another.
- Temporal Continuity describes the persistence of identity and state for long periods of time.
- Knowledge-level communication ability characterizes the ability to communicate with language more resembling human-like "speech acts" than typical symbol-level program-to-program protocols.
- Reactivity is the ability to selectively sense and act.
- Personality is the adaptability to different preferences and special requirement of users and lastly.
- Inferential Capability-Intelligence is the ability to act, based on abstract task specifications.

2.3.2 *Agent Terminology*

In the last few years, a terminology relevant to agent technology has started to be used. According to the Roberto A.-1999, the following terms regarding agent technology are recognized: *Agent Architectures*, *Agent System Architectures*, *Agent Frameworks*

and *Agent Infrastructures*. *Agent Architectures* describe agents as separate entities, consisting of three basic modules named perception, reasoning and action module. *Agent System Architectures* define the interaction of agents under constraints in a common environment. *Agent Frameworks* are sets of tools and integrated environments for the development of agents and multi-agent systems: Agent Construction Tools [2] and Agent Platform Evaluation and Comparison [54]. Finally, *Agent Infrastructures* provide means for agent communication and common understandings of various concepts. Their basic components are *Ontologies, Interaction Protocols, Communication Languages* and *Communication Infrastructures.* Ontologies are formal specifications, which describe concepts, their properties and relations in a particular domain of interest. Agent Communication Languages (ACLs) [49] provide agents with means of exchanging information and knowledge, Yannis Labrou et al (1999) suggest this paradigm for software development.

Interaction Protocols are the rules, which enable agents to reason over the effects of their communications. Finally, Communication Infrastructures provide communication channels among agents.

2.4 Semantic Web

The aim of the Semantic Web initiative is to advance the state of the current Web through the use of semantics.

More specifically, it proposes to use *semantic annotations* to describe the meaning of certain parts of Web information. For example, Project definitions could be suitably annotated to distinguish between project name, implementation location, category, beneficiaries etc.

Such meta-data could facilitate the automated processing of the information on the Web site, thus making it accessible to machines and not primarily to human users, as it is the case today.

However, the question arises as to how the semantic annotations of different Web sites can be combined, if everyone uses terminologies of their own. The solution lies in the organization of vocabularies in so-called *ontologies*. References to such shared vocabularies allow interoperability between different Web resources and applications.

For example, ontology of project definitions could be used to determine that geographical information systems are categorized as software projects and Dam

construction is a development-related project. Such information would be crucial to establish the categories of projects suggestions and hence inform the decisions of the selections made.

The development of the Semantic Web proceeds in steps, each step building a layer on top of another. The layered design is shown in *Figure 2: The semantic Web Tower*, which is outlined below.

- At the bottom layer we find eXtensible Markup Language (*XML*), [15], a language that lets one write structured web documents with a user-defined vocabulary. XML is particularly suitable for sending documents across the Web, thus supporting syntactic interoperability.
- Resource Description Framework (RDF) is a basic data model, like the entity-relationship model, for writing simple statements about Web objects (resources). The RDF data model does not rely on XML, but RDF has an XML-based syntax. Therefore, it is located on top of the XML layer.
- RDF Schema provides modeling primitives, for organizing Web objects into hierarchies. RDF Schema is based on RDF. RDF Schema can be viewed as a primitive language for writing ontologies.
- But there is a need for more powerful ontology languages that expand RDF Schema and allow the representations of more complex relationships between Web objects. Ontology languages, such as Web Ontology Language (OWL), are built on the top of RDF and RDF Schema.
- The logic layer is used to enhance the ontology language further, and to allow writing application-specific declarative knowledge.
- The *proof layer* involves the actual deductive process, as well as the representation of proofs in Web languages and proof validation.
- Finally *trust* will emerge through the use of digital signatures, and other kind of knowledge, based on recommendations by agents, or rating and certification agencies and consumer bodies.

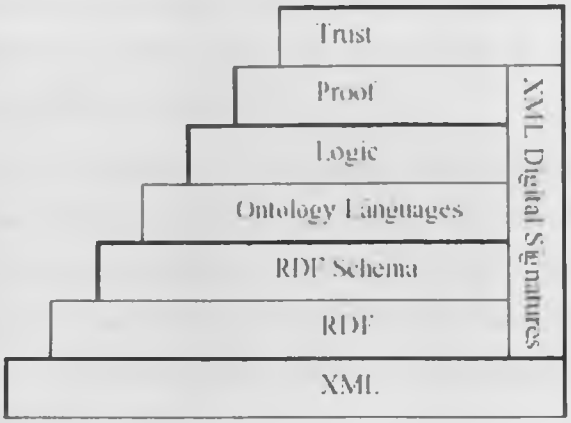


Figure 2: The semantic Web Tower

2.4.1 XML Basics

XML, Tim Bray et al (2000) and as documented by the World Wide Web Consortium stands for eXtensible Markup Language. It is a mark-up language much like HTML.

XML was designed to describe data and its tags are not predefined. The user must define his own tags by using a Document Type Definition (DTD) or an XML Schema to define the legal building blocks of an XML document, that is, define elements and attributes that can appear in a document, which elements are child elements, what is the order of child elements etc. XML with a DTD or XML Schema is self-descriptive. XML is not a replacement for HTML. They were designed with different goals: The former was designed to describe data and to focus on what data is and the latter was designed to display data and to focus on how data looks. HTML is about displaying information, while XML is about describing information. XML was created to structure, store and share information.

2.4.2 RDF Basic Features

RDF [11] stands for Resource Description Framework and its purpose is to describe resources on the Web. RDF is designed to be read by computers. The basic RDF data model consists of three fundamental concepts: *Resources*, *Properties* and *Statements*.

Resources are the central concept of RDF and are used to describe individual objects of any kind, for example Web pages, people, projects, flights etc. Every resource has a URI, a Universal Resource Identifier, which can be a Web address or some other kind of unique identifier.

Properties express specific aspects, characteristics, attributes, or relations between resources. For example, properties might be the number of people benefiting in a project, name of the benefiting location etc.

Finally statements are composed of a specific resource, together with a named property and the *value* of that property for that resource. The value can be a resource in turn; for example, the project category of School Construction Project proposal is education. Alternatively, the value can be a *literal*, a primitive term that is not evaluated by an RDF processor. For example, the number of households to be benefited by the school construction project is 1020.

A statement consists of three parts (subject, predicate, object) and is often referred to as an RDF triple. A triple of the form (x, P, y) corresponds to the logical formula P(x, y), where the binary predicate P relates the object x to the object y; this representation is used in the proposed system for translating RDF statements into a logical language ready to be processed automatically in conjunction with rules. Another possible representation is the graph based. The graph is directed with labeled nodes and arcs. The arcs are directed from the resource (the subject of the statement) to the value (the object of the statement); see *the illustration (Figure 3 : Graph Representation of RDF Statements) below.*

This kind of graph is known as a Semantic Net in the artificial intelligence community.

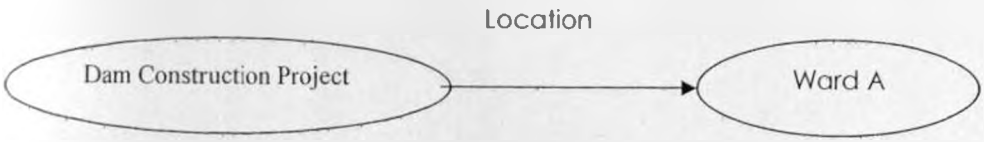


Figure 3 : Graph Representation of RDF Statements

Lastly, there is a third representation based on XML. This representation is compatible with the layered design of the Semantic Web, and facilitates exchange of RDF information among applications. Such a representation is depicted in *Figure 4: XML-Based Representation of an RDF Statement*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY cdf "http://10.2.21.44/cdf_projects/projects.rdf#">
<!ENTITY cdf_ex "http://10.2.21.44/cdf_projects/2007_2008/080_projects.rdf#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDF
xmlns:cdf_ex="http://10.2.21.44/cdf_projects/2007_2008/080_projects.rdf#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:cdf="http://10.2.21.44/cdf_projects/projects.rdf#"
>
<rdf:Description rdf:about="&cdf_ex;allocation">
<cdf:allocation>
     3921408
</cdf:allocation>
</rdf:Description>
<cdf:projects rdf:about="&cdf_ex;P1">
<cdf:name>Borehole Drilling</cdf:name>
<cdf:imp-cost rdf:datatype="&xsd;integer">2000000</cdf:imp-cost>
<cdf:op-cost rdf:datatype="&xsd;integer">50000</cdf:op-cost>
<cdf:target-size rdf:datatype="&xsd;integer">10000</cdf:target-size>
<cdf:project-area>environmental</cdf:project-area>
<cdf:geo-coverage rdf:datatype="&xsd;integer">3000</cdf:geo-coverage>
<cdf:location>LocationC</cdf:location>
<cdf:prev-imp>yes</cdf:prev-imp>
<cdf:project-status>different</cdf:project-status>
<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>
<cdf:complementary_proposal>no</cdf:complementary_proposal>
</cdf:projects>
```

Value

Resource

Figure 4: XML-Based Representation of an RDF Statement

2.4.3 RDF Schema Basic Features

RDF is domain-independent, in that no assumptions about a particular domain of use are made. It is up to the users to define their own terminology in a schema language called RDF Schema (RDFS) [16]. In doing so, they actually define a simple ontology, a conceptual model of the domain at hand. The basic features of RDF Schema are the following.

In RDF, Web resources are individual objects. In RDFS, objects sharing similar characteristics are put together to form *classes*. Examples for classes are constituencies, projects, constituents, annual allocations etc. Individuals belonging to a class are often referred to as *instances* of that class. For example, *Dam construction* is an instance of the class of projects of a particular constituency.

Binary *properties* are used to establish connections between classes. For example, a property *are_Allocated* establishes a connection between projects and Annual Allocations. Properties apply to individual objects (instances of the classes involved) to form RDF statements, as seen above.

The application of predicates can be restricted through the use of *domain and range restrictions*. For example, we can restrict the property *are_Allocated* to apply only to CDF Projects (domain restriction), and to have as value only CDF annual allocations (range restriction). This way, nonsensical statements due to user errors can be automatically detected.

Classes can be put together in hierarchies through the *subclass relationship*: a class C is a subclass of a class D if every instance of C is also an instance of D. For example, the class of school bursaries is a subclass of all projects: every instance of a CDF funded school bursary is also a CDF Projects.

The hierarchical organization of classes is important due to the notion of inheritance: once a class C has been declared a subclass of D, every known instance of C is automatically classified also as instance of D. This has far-reaching implications for matching requester preferences to service offerings. This information is inferred by the ontology automatically.

2.4.4 Owl Basic Features

OWL (Web Ontology Language) [51] comes to fill the missing features of RDF and RDFS. According to Grigoris Antoniou et al (2004), OWL deals with the following issues that RDF cannot express:

- Local scope of properties: *rdfs: range* defines the range of a property, say teaches, for all classes. Thus in RDF Schema we cannot declare range restrictions that apply to some classes only. For example, we cannot say that projects belonging to poverty reduction and health only, while other belonging to education, too.
- Disjointness of classes: Sometimes we wish to say that classes are disjoint. For example, male and female are disjoint. But in RDF Schema we can only state subclass relationships, e.g. female is a subclass of person.
- Boolean combinations of classes: Sometimes we wish to build new classes by combining other classes using union, intersection and complement. For example, we may wish to define the class person to be the disjoint union of the classes male and female. RDF Schema does not allow such definitions.
- Cardinality restrictions: Sometimes we wish to place restrictions on how many distinct values a property may or must take. For example, we would like to say that a project can only belong to category Poverty Reduction, education and Health, and that a project has at most one location. Again such restrictions are impossible to express in RDF Schema.
- Special characteristics of properties: Sometimes it is useful to say that a property is transitive (like "greater than"), unique (like "is mother of"), or the inverse of another property (like "avails" and "is availed by").

2.5 Defeasible Logic

2.5.1 Non-monotonic Reasoning

One of the issues that have recently attracted the concentration of the developers of the Semantic Web is the nature of the rule systems that should be employed in the logic layer of the Semantic Web tower. Monotonic rule systems have already been studied and accepted as an essential part of the layered development of the Semantic Web. Non-monotonic rule systems, on the other hand, seem also to be a good solution, especially due to their expressive capabilities.

Non-monotonic reasoning is a subfield of Artificial Intelligence trying to find more realistic formal models of reasoning than classical logic. In common sense reasoning, one often draws conclusions that have to be withdrawn, when further information is obtained.

Thus, the set of conclusions does not grow monotonically with the given information. Non-monotonic reasoning methods try to formalize the latter phenomenon.

In a monotonic logic system, given a collection of facts *D* that entail some sentence *S* (*S* is a logical conclusion of *D*), for any collection of facts *D'* such that *DD'*, *D'* also entails *S*. In other words: *S* is also a logical conclusion of any superset of *D*.

In a non-monotonic system, the addition of new facts can reduce the set of logical conclusions. So, if *S* is a logical conclusion of *D*, it is not necessarily a conclusion of any superset of *D*. Two of the basic characteristics of non-monotonic systems are: *adaptability* (ability to deal with a changing environment), and the ability to reason under conditions of *uncertainty*. In other words, such systems are capable of adding and retracting beliefs as new sets of information is available, and reasoning with an incomplete set of facts.

Defeasible logic, which was introduced by Donald Nute (1994), is a representative language of non-monotonic reasoning. In general, a defeasible theory (a knowledge base in defeasible logic) consists of five different kinds of knowledge: facts, strict rules, defeasible rules, defeaters, and a superiority relation.

Facts are indisputable statements, for example, "Dam construction is a development Project". Written formally, this would be expressed as

development_projects (*dam_construction*)

Strict Rules are rules in the classical sense: whenever the premises are indisputable (e.g., facts) then so is the conclusion. An example of a strict rule is "cattle dips projects are community projects". Written formally:

cattle_dips (X) → *community_projects*(X)

Defeasible rules are rules that can be defeated by contrary evidence. An example of such a rule is "All Projects are typically acceptable"; written formally:

projects (X) =>*acceptable* (X)

The idea is that if we know that something is a project, then we may conclude that it is acceptable, *unless there is other, not inferior, evidence suggesting that it may not* acceptable.

Defeaters are rules that cannot be used to draw any conclusions. Their only use is to prevent some conclusions. In other words, they are used to defeat some defeasible rules by producing evidence to the contrary. An example is "If a project does not span more than 5000 square Kilometers in geographical location then it might not be acceptable". Formally:

geo-coverage(X, Y), Y<5000=> ¬*acceptable*(X);

The main point is that the information that a project does not span 5000 square kilometers is not sufficient evidence to conclude that it is not acceptable. It is only evidence that the project may not be implemented. In other words, we do not wish to conclude ¬*acceptable* (X) if g*eo-coverage*(X, Y), Y<5000 we simply want to prevent a conclusion *acceptable* (X).

The ***superiority relation*** among rules is used to define priorities among rules, i.e., where one rule may override the conclusion of another rule. For example, given the *defeasible rules*

r: project (X) =>*acceptable* (X)

s: target-group(X, Y), Y<500 => ¬*acceptable*(X);

Which contradict one another, no conclusive decision can be made about whether a project that target less that 500 people can be implemented. But if we introduce a superiority relation > with *s* > *r*, with the intended meaning that *s* is strictly stronger than *r*, then we can indeed conclude that the project cannot be implemented.

Notice that a cycle in the superiority relation is counterintuitive. In the above example, it makes no sense to have both *r* > *s and s* > *r*. Consequently, we focus on cases where the superiority relation is acyclic.

Another point worth noting is that, in Defeasible Logic, priorities are *local* in the following sense: two rules are considered to be competing with one another only if they have complementary heads. Thus, since the superiority relation is used to resolve conflicts among competing rules, it is only used to compare rules with complementary heads; the information *r* > *s* for rules *r*, *s* without complementary heads may be part of the superiority relation, but has no effect on the proof theory. Hence, *Defeasible reasoning* is a simple rule-based approach to reasoning with incomplete and inconsistent information. It can represent facts, rules, and priorities among rules.

This reasoning family comprises defeasible logics [57], [4] and Courteous Logic Programs [4]. This approach has the following characteristics:

a) They are rule-based, without disjunction
b) Classical negation is used in the heads and bodies of rules, but negation-as-failure is not necessarily used in the object language (it can easily be simulated, if necessary [36])
c) Rules may support conflicting conclusions
d) The logics are skeptical in the sense that conflicting rules do not fire. Thus consistency is preserved
e) Priorities on rules may be used to resolve some conflicts among rules
f) Finally, the logics take a pragmatic view and have low computational complexity.

The above properties make this family a good candidate to be used on the Semantic Web, as well as to be used in applications where timely response is essential.

The choice of defeasible logic among various schemes for representing strategies and protocols in this work is justified as follows;

Firstly is *formal*, that is, its semantics and syntax are properly defined. This means that both humans and computers can interpret them the same way. Another characteristic of defeasible logic is that it is *conceptual* meaning that it offers a good level of abstraction. So anyone can focus only on protocol or strategy design, being indifferent to the implementation. Defeasible logic is also *comprehensible* and *expressive* as well. The latter is very important because enables us to describe a wide range of protocols and strategies. Lastly, if there is a defeasible logic inference engine available it is also *executable*.

2.5.2 DR-DEVICE ENGINE

Defeasible reasoning is a rule-based approach for efficient reasoning with incomplete and inconsistent information. Such reasoning is, among others, useful for ontology integration, where conflicting information arises naturally; and for the modeling of business rules and policies, where rules with exceptions are often used. DR-DEVICE is an implementation of a defeasible reasoning system for reasoning on the Web.

DR-DEVICE ([8], [9], [10]) is capable of reasoning about RDF metadata over multiple Web sources using defeasible logic rules. It is implemented on top of CLIPS production

rule system and builds upon R-DEVICE ([7], [9]), an earlier deductive rule system over RDF metadata that also supports derived attribute and aggregate attribute rules. Rules can be expressed either in a native CLIPS-like language, or in an extension of the OO-RuleML[4] syntax. The operational semantics of defeasible logic are implemented through compilation into the generic rule language of R-DEVICE.

DR-DEVICE's main characteristics are as follows:

- Its user interface is compatible with RuleML [62], the main standardization effort for rules on the Semantic Web.
- It is based on Prolog. The core of the system consists of a translation of defeasible knowledge into Prolog. However, the implementation is declarative because it interprets the *not* operator using Well-Founded Semantics [68].
- The main focus is flexibility. Strict and defeasible rules and priorities are part of the interface and the implementation. Also, a number of variants are implemented (ambiguity blocking, ambiguity propagating, conflicting literals).

The DR-DEVICE system consists of two major components: *the RDF loader/translator* and the *rule loader/translator*. The former accepts from the latter (or the user) requests for loading specific RDF documents. The RDF triple loader downloads the RDF document from the Internet and uses the ARP parser[5] [50] to translate it to triples in the N-triple[6] format. Both the RDF/XML and N-triple files are stored locally for future reference. Furthermore, the RDF document is recursively scanned for namespaces which are also parsed using the ARP parser. The rationale for translating namespaces is to obtain a complete RDF Schema in order to minimize the number of OO[7] schema redefinitions. Fetching multiple RDF schema files will aggregate multiple RDF-to-OO schema translations into a single OO schema redefinition.

Namespace resolution is not guaranteed to yield an RDF schema document; therefore, if the namespace Universal resource indicator (URI) is not an RDF document, then the ARP parser will not produce triples and DR-DEVICE will make assumptions, based on the RDF semantics [40], about non-resolved properties, resources, classes, etc. All N-

---

[4] http://www.ruleml.org/

[5] ARP Parser is an Xerces used for parsing the XML documents

[6] N-Triples is a line-based, plain text format for representing the correct answers for parsing RDF/XML test cases as part of the RDF Core working group. Parsing an N-Triple document results in a sequence of RDF statements formed from the subject, predicate and object terms

[7] is a schema language (or metagrammar) for defining the syntactic structure and partial semantics of XML document types

triples are loaded into memory, while the resources that have a *URI#anchorID* or *URI/anchorID* format are transformed into a *ns:anchored* format if URI belongs to the initially collected namespaces, in order to save memory space. The transformed RDF triples are fed to the RDF triple translator which maps them into COOL objects and then deletes them.

The rule loader accepts from the user a URI (or a local file name) that contains a defeasible logic rule program in RuleML notation [62] .

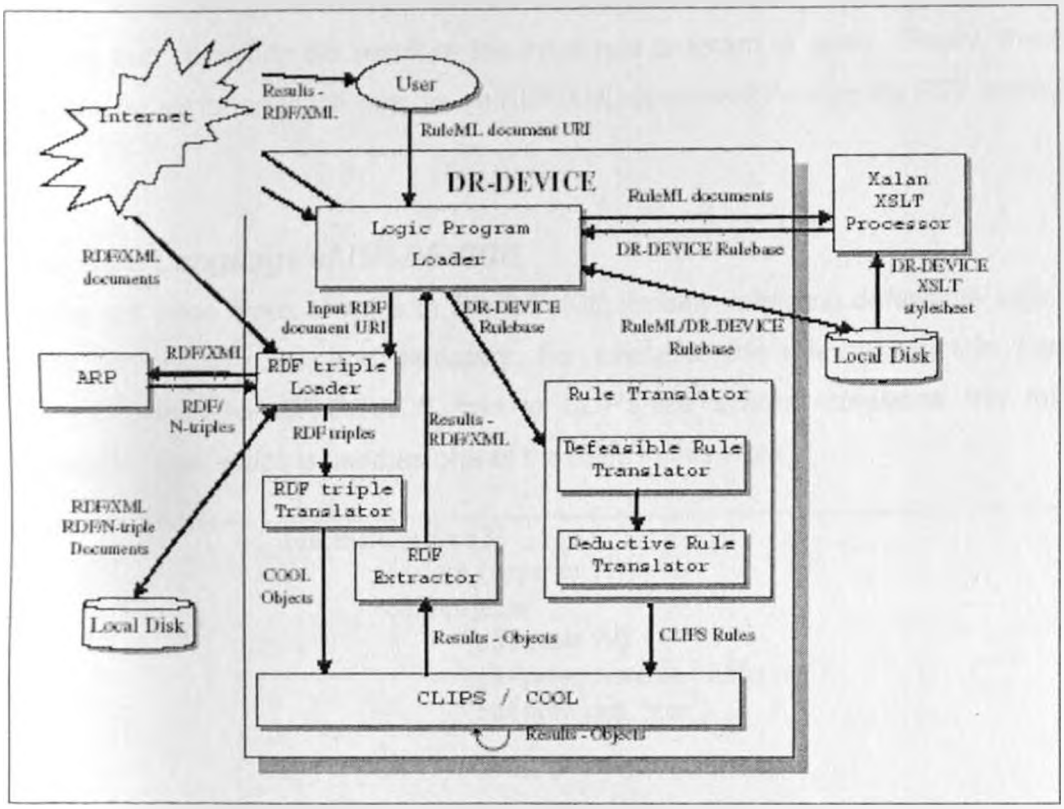


Figure 5: Architecture of the DR-DEVICE reasoning Module

The RuleML document may also contain the URI of the input RDF document on which the rule program will run, which is forwarded to the RDF loader. The RuleML program is translated into the native DR-DEVICE rule notation using the Xalan XSLT processor[8] [72] and an XSLT stylesheet. The DR-DEVICE rule program is then forwarded to the rule translator. The rule translator accepts from the rule loader (or directly from the user) a

[8] http://xml.apache.org/xalan-j/

set of rules in DR-DEVICE notation and translates them into a set of CLIPS production rules.

The translation of the defeasible logic rules is performed in two steps: first, the defeasible logic rules are translated into sets of deductive, derived attribute and aggregate attribute rules of the basic R-DEVICE rule language and then, all these rules are translated into CLIPS production rules. All compiled rule formats are kept into local files, so that the next time they are needed they can be directly loaded, increasing speed. When the translation ends, CLIPS runs the production rules and generates the objects that constitute the result of the initial rule program or query. Finally, the result-objects are exported to the user as an RDF/XML document through the RDF extractor.

The Rule Language of DR-DEVICE

There are three types of rules in DR-DEVICE, closely reflecting defeasible logic: strict rules, defeasible rules, and defeaters. For example, the rule construct in *Figure 6: Sample defeasible DR-DEVICE rule in CLIPS-like syntax* represents the following defeasible rule, which is used as one of the rules in this work.

```
(defeasiblerule r3
        (declare (superior r1))
        (cdf projects
                (cdf name ?x)
                (cdf project-status,"different")
                (cdf prev-imp,"yes")
        )
=>
   (not
                ( acceptable
                        (project ?x)
                )
        )
)
```

Figure 6: Sample defeasible DR-DEVICE rule in CLIPS-like syntax

Predicates have named arguments, called slots, since they represent CLIPS objects. DR-DEVICE has also a RuleML-like syntax. The same rule is represented in RuleML notation (version 0.85) as shown in

Figure 7. Several features of defeasible logic and its DR-DEVICE implementation could not be captured by the latest RuleML DTDs, so a new DTD was developed using the

modularization scheme of RuleML, extending the Datalog with negation DTD (both classical and NAF) with OO features.

Classes and objects (facts) can also be declared in DR-DEVICE; however, the focus of this work is the use of RDF data as facts. The input RDF file(s) are declared in the *rdf_import* attribute of the *rulebase (root)* element of the RuleML document. There exist two more attributes in the *rulebase* element: the *rdf_export* attribute that declares the address of the RDF file with the results of the rule program to be exported, and the

rdf_export_classes attribute that declares the derived classes whose instances will be exported in RDF/XML format. Further extensions to the RuleML syntax, include function calls that are used either as constraints in the rule body or as new value calculators at the rule head. Furthermore, multiple constraints in the rule body can be expressed through the logical operators: *_not*, *_and*, *_or*.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rulebase SYSTEM "http://10.2.21.44/cdf_projects/dr-device.dtd" [
<!ENTITY cdf "http://10.2.21.44/cdf_projects/cdf.rdf#">
<!ENTITY cdf_rb "http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#">
]>
<rulebase direction="forward" rdf_export="export-cdf.rdf" rdf_export_classes="largesttarget implement" rdf_import="http:/

        <_rbaselab>
                <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#" type="defeasible">cdf-rules</ind>
</_rbaselab>
        <competing_rules c_rules="r15 r16 r17">
                <_crlab>
                        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#cr1">cr1</ind>
                </_crlab>
</competing_rules>

        <imp>
                <_rlab ruleID="r1" ruletype="defeasiblerule">
                        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r1">r1</ind>
                </_rlab>

                <_head>
                        <atom>
                                <_opr>
                                        <rel>acceptable</rel>
                                </_opr>
                                <_slot name="projects">
                                        <var>x</var>
                                </_slot>
                        </atom>
                </_head>

                <_body>
                        <atom>
                                <_opr>
                                        <rel href="cdf:projects"></rel>
                                </_opr>
                                <_slot name="cdf:project-name">
                                        <var>x</var>
                                </_slot>
                        </atom>
                </_body>
</imp>
```

Figure 7: Sample defeasible DR-DEVICE rule in RuleML-like syntax.

The translation of defeasible rules into R-DEVICE rules is based on the translation of defeasible theories into logic programs through a meta-program. The meta-program is used to guide defeasible rule compilation. Each defeasible rule in DR-DEVICE is translated into a set of 5 R-DEVICE rules. Correct order of execution is guaranteed by predefined ordering among different R-DEVICE rule types and by stratification. For non-stratified programs the correct result is guaranteed through "truth maintenance" rules that undo (retract) the conclusions of rules when their condition is no longer met. In this way, even if rules are not executed in the correct order, the correct result will be eventually deduced because conclusions of rules that should have not been executed can be later undone.

DR-DEVICE Rule Syntax

```
<!ELEMENT rulebase (((_rbaselab, (imp | competing_rules)*) |
((imp | competing_rules)+, _rbaselab?))?)>
<!ATTLIST rulebase
xmlns CDATA #IMPLIED
xsi:schemaLocation CDATA #IMPLIED
xmlns:xsi CDATA #IMPLIED
<!-- The URLs of the RDF files to load -->
rdf_import CDATA #IMPLIED
<!-- The name of the file that the results of the reasoning process will be written to -->
rdf_export_classes NMTOKENS #IMPLIED
<!-- The names of the classes whose instances will constitute the results of the inference process
-->
rdf_export CDATA #IMPLIED
>
<!ELEMENT _rbaselab (ind)>
<!ELEMENT imp ((_rlab, ((_head, _body) | (_body, _head))) |
(_head, ((_rlab, _body) | (_body, _rlab?))) |
(_body, ((_rlab, _head) | (_head, _rlab?))))>
<!ELEMENT competing_rules (_crlab, _slots?)>
<!ATTLIST competing_rules
```

```
c_rules IDREFS #REQUIRED
>
<!ELEMENT _slots (slotname+)>
<!ELEMENT slotname (#PCDATA)>
<!ELEMENT _rlab (ind)>
<!ATTLIST _rlab
ruleID ID #REQUIRED
ruletype (strictrule | defeasiblerule | defeater) #REQUIRED
superior IDREFS #IMPLIED
>
<!ELEMENT _crlab (ind)>
<!ELEMENT _head (calc?, (atom | neg))>
<!ELEMENT _body (atom | neg | naf | and | or)>
<!ELEMENT atom ((_opr, (_slot)*, ((ind | var)+, (_slot)*)?) |
((((_slot)+, ((ind | var)+, (_slot)*)?) | ((ind | var)+, (_slot)*)), _opr))>
<!ELEMENT and ((atom | neg | naf | or)*)>
<!ELEMENT or ((atom | neg | naf | and)*)>
<!ELEMENT neg (atom)>
<!ELEMENT naf (atom | and)>
<!ELEMENT _opr (rel)>
<!ELEMENT rel (#PCDATA)>
<!ATTLIST rel
href CDATA #IMPLIED
>
<!ELEMENT _slot (ind | var | _not | _or | _and)>
<!ATTLIST _slot
name CDATA #REQUIRED
card CDATA #IMPLIED
weight CDATA #IMPLIED
```

```
>
<!ELEMENT ind (#PCDATA)>
<!ATTLIST ind
type CDATA #IMPLIED
href CDATA #IMPLIED
>
<!ELEMENT var (#PCDATA)>
<!ATTLIST var
type CDATA #IMPLIED
>
<!ELEMENT calc (function_call+)>
<!ELEMENT function_call (fname, (ind | var | function_call)*)>
<!ELEMENT fname (#PCDATA)>
<!ENTITY % term "(_not | ind | var | function_call)">
<!ELEMENT _not (ind | var)>
<!ELEMENT _or (%term;, (%term;)+)>
<!ELEMENT _and (%term;, (%term;)+)>
```

2.6 Brokering-Matchmaking

There is no consensus on the definition of terms brokering and matchmaking along with what functions they involve.

However, most researchers ascribe almost identical characteristics and functionality to these terms. Thus, according to D. Trastour et al (2001), brokering or matchmaking is the process by which parties that are interested in having exchange of economic value are put in contact with potential counterparts.

According to L. Li and I. Horrocks (2003), brokering or matchmaking is defined as a process that requires a repository host to take a query or advertisement's input and to return all advertisements that may potentially satisfy the requirements specified in the input query or advertisement.

M. Nodine et al (1998) defines that broker is someone who offers a matchmaking service to potential service providers and service requesters. In this work the term brokering is referring to both terms.

The main dimensions of brokering are advertisements, preferences & queries and brokering engines. An advertisement models the offers for a service of a potential service provider and encloses service's features, constraints and special characteristics.

It defines a space of possible realizations of a service. The query, which may also be a request advertisement, expresses constraints over aspects of advertised services that the submitter is interested in. It is a way for the requester to filter out existing advertisements that are not important to him. Lastly, Brokering engine is a mechanism, which employs a specific technique and does the act of brokering, which is the matching of the preferences of the service requesters and the advertisements of the service providers. A set of advertisements (subset of the total of advertisements) is returned to the service requester.

2.6.1 Brokering Techniques

After a review of the related literature, there seems not to be a complete classification of the brokering techniques. A classification scheme can be shaped as follows but it is possible that other techniques may exist.

Minghua He et al (2003) have identified three techniques for brokering. They are called Feature-based filtering, Collaborative Filtering and Constrained-based Filtering.

- Feature-based filtering involves selecting products based on feature keywords. For example, suppose the committee in a certain constituency wants to implement a poverty reduction project that benefits over 2000 people and it is not more KES.10,000,000 . The representative agent first select "*poverty_reduction*" categorized project first, then indicates "*>2000*" in the target group field and all projects under poverty reduction targeting over 2000 people are returned.

- Collaborative filtering involves giving an agent personalized recommendations based on the similarities between different users' preference profiles. Here, the project rating of group A is first compared with that of all the other groups in the system. Then, the "nearest neighbor" of A (i.e., the group whose profile is closest to that of A) is identified. Since groups with similar tastes and

preferences are likely to select similar projects, the profile of the identified group is used to pass recommendations onto A's agent.

- Constraint-based filtering involves an agent specifying constraints (e.g., the category range and target limit) to narrow down the projects. In this way, requesters' agents are guided through a large feature space of the product. In the end, a list of the desired projects that satisfy the user's constraints is returned. Some e-commerce systems use more than one kind of filtering technique (since sometimes users do not know exactly the constraints of the products they are looking for in advance)

In the last few years declarative approaches have started to emerge. In addition many of these approaches start to adopt semantic web technologies. There are many variations for brokering which are based on declarative approaches:

- A common technique uses a logic programming language for the modeling of both the advertisements and the queries. In this work, a rule-based inference engine is used for the brokering procedure.
- A more recent technique makes use of the data model of RDF for the description of both the advertisements and the requests. It uses a matching algorithm as the brokering engine; thereby matching of advertisements is reduced to matching of RDF graphs.
- Both advertisements and requests can also be expressed in a description logic-based web ontology language, such as OWL or DAML+OIL. In this case, advertisements and requests are converted to Aboxes and Tboxes and afterwards a description logic reasoner is used.
- A technique could make use of RDF for expressing advertisements and the use for a query language like those described by D. Trastour et al (2001) for expressing queries of service requesters.
- Finally, a hybrid approach is possible. The idea is that advertisements are expressed in a description logic based web ontology language or a data model such as RDF, but the queries and preferences of users are expressed through a logic programming language. In this case, the brokering engine must be able to convert description logic to declarative logic, in order to perform the reasoning process.

2.6.2 *Fundamental Requirements for a Brokering System*

The minimal functionalities that a brokering service provides are the features of a *Messaging Middleware*, *Advertising* a service, and *Browsing* or *Querying* a repository of advertised services according to D. Trastour et al (2001). More specifically:

- A messaging middleware must exist that allows message exchange among service requester, service provider and broker.
- There is a need for a language to express advertisements. This language must be flexible and highly expressive, support to express semi-structured data, types, subsumption and constraints.
- The existence of a language for expressing queries is indispensable for a brokering system. If queries are in the form of request advertisements, the characteristics of a language for advertisement also apply here. In addition the ability for the expression of complex queries is critical.

2.6.3 *Selected Work in Brokering*

Net Perception and CDNOW are two brokering systems, which use a collaborative technique. In Net Perceptions [59], users recommend the documents that their "knowledge neighbors" find valuable. In CDNOW [1], users are notified about the CDs or movies that are popular with other users with similar preferences. EBay, [27] guides a user agent to select the products by narrowing down the range of the possibilities based on the constraints the user gives (e.g. price range, item location, and so on). In the end, a list of the desired products that satisfy the user's constraints is returned. It uses a combination of constraint and feature-based techniques.

InfoSleuth as documented by M. Nodine et al (1998) is an agent-based information discovery and retrieval system, which performs a logic programming brokering. It adopts "broker agents" to perform the syntactic and semantic matchmaking. The broker agent matches agents that require services with other agents that can provide those services. By maintaining a repository, which contains up-to-date information about the operational agents and their services, the broker enables the querying agent to locate all available agents, which provide appropriate services. Syntactic brokering is the process of matching requests to agents on the basis of the syntax of the incoming messages which wrap the requests; semantic brokering is the process of matching requests to agents on the basis of the requested agent capabilities or services, with the agent capabilities and

services being described in a common shared ontology of attributes and constraints. This single domain-specific ontology is a shared vocabulary that all agents can use to specify advertisements and requests to the broker. In InfoSleuth, the service capability information that is regarded as the advertisement is written in LDL++, a logical deduction language. Agents use a set of LDL++ deductive rules to support inferences about whether an expression of requirements matches a set of advertised capabilities.

Chen et al. [19] propose a hybrid technique. The iAgent stores knowledge in the inference layer and makes inference by it, so iAgent can answer the queries of users or other agents. As they stress, a typical knowledge base makes inference according to the rules and facts, but they choose not to predefine rules and facts in the inference layer. Such design makes the inference layer as portable as possible, so different iAgents in different domains can use the same inference layer. The facts are extracted from semantic markup documents which are written in .DAML+OIL[9]. A fact translator converts all the DAML+OIL documents into Prolog formats. IAgent chooses a Prolog (Horn-logic based) engine, SWI-Prolog, as its inference engine.

2.7 *Justification of Software agents*

In this subsection, a comparison is done of a few other engineering methodologies that may be used in place of software agents in the brokering system proposed in this work. What follows is first a little background of each methodology.

2.7.1 *Game Theory*

What is game theory?

Game theory is the formal study of conflict and cooperation. Game theoretic concepts apply whenever the actions of several agents are interdependent. These agents may be individuals, groups, firms, or any combination of these. The concepts of game theory provide a language to formulate structure, analyze, and understand strategic scenarios.

History and impact of game theory

The earliest example of a formal game-theoretic analysis is the study of a duopoly by Antoine Cournot in 1838. The mathematician Emile Borel suggested a formal theory of games in 1921, which was furthered by the mathematician John von Neumann in 1928

---

[9] *DAML+OIL is a semantic markup language for Web resources. It builds on earlier W3C standards such as RDF and RDF Schema, and extends these languages with richer modelling primitives. DAML+OIL provides modelling primitives commonly found in frame-based languages.*

in a "theory of parlor games." Game theory was established as a field in its own right after the 1944 publication of the monumental volume Theory of Games and Economic Behavior by von Neumann and the economist Oskar Morgenstern. This book provided much of the basic terminology and problem setup that is still in use today.

In 1950, John Nash demonstrated that finite games have always have an equilibrium point, at which all players choose actions which are best for them given their opponents' choices. This central concept of non-cooperative game theory has been a focal point of analysis since then. In the 1950s and 1960s, game theory was broadened theoretically and applied to problems of war and politics. Since the 1970s, it has driven a revolution in economic theory. Additionally, it has found applications in sociology and psychology, and established links with evolution and biology. Game theory received special attention in 1994 with the awarding of the Nobel prize in economics to Nash, John Harsanyi, and Reinhard Selten.

At the end of the 1990s, a high-profile application of game theory had been the design of auctions. Prominent game theorists have been involved in the design of auctions for allocating rights to the use of bands of the electromagnetic spectrum to the mobile telecommunications industry. Most of these auctions were designed with the goal of allocating these resources more efficiently than traditional governmental practices, and additionally raised billions of dollars in the United States and Europe.

Definitions of games

The object of study in game theory is the game, which is a formal model of an interactive situation. It typically involves several players; a game with only one player is usually called a *decision problem*. The formal definition lays out the players, their preferences, their information, their strategic actions available to them, and how these influence the outcome.

Games can be described formally at various levels of detail. *A coalitional (or cooperative)* game is a high-level description, specifying only what payoffs each potential group, or coalition, can obtain by the cooperation of its members.

What is not made explicit is the process by which the coalition forms. As an example, the players may be several parties in parliament. Each party has a different strength, based upon the number of seats occupied by party members. The game describes which coalitions of parties can form a majority, but does not delineate, for example, the negotiation process through which an agreement to vote en bloc is achieved.

Cooperative game theory investigates such coalitional games with respect to the relative amounts of power held by various players, or how a successful coalition should divide its proceeds. This is most naturally applied to situations arising in political science or international relations, where concepts like power are most important.

For example, Nash proposed a solution for the division of gains from agreement in a bargaining problem which depends solely on the relative strengths of the two parties' bargaining position.

The amount of power a side has is determined by the usually inefficient outcome that results when negotiations break down. Nash's model fits within the cooperative framework in that it does not delineate a specific timeline of offers and counteroffers, but rather focuses solely on the outcome of the bargaining process.

In contrast, non-cooperative game theory is concerned with the analysis of strategic choices. The paradigm of non-cooperative game theory is that the details of the ordering and timing of players' choices are crucial to determining the outcome of a game.

In contrast to Nash's cooperative model, a *non-cooperative model* of bargaining would posit a specific process in which it is pre-specified who gets to make an offer at a given time. The term "non-cooperative" means this branch of game theory explicitly models the process of many players making choices out of their own interest.

Cooperation can, and often does, arise in non-cooperative models of games, when players find it in their own best interests.

Branches of game theory also differ in their assumptions. A central assumption in many variants of game theory is that the players are rational. A rational player is one who always chooses an action which gives the outcome he most prefers, given what he expects his opponents to do. The goal of game-theoretic analysis in these branches, then, is to predict how the game will be played by rational players, or, relatedly, to give advice on how best to play the game against opponents who are rational. This rationality assumption can be relaxed, and the resulting models have been more recently applied to the analysis of observed behavior (see Kagel and Roth, eds., Handbook of Experimental Economics, Princeton Univ. Press, 1997). This kind of game theory can be viewed as more "descriptive" than the prescriptive approach taken here.

Game theory and information systems

The internal consistency and mathematical foundations of game theory make it a prime tool for modeling and designing automated decision-making processes in interactive environments.

For example, one might like to have efficient bidding rules for an auction website, or tamper-proof automated negotiations for purchasing communication bandwidth. Research in these applications of game theory is the topic of recent conference and journal papers (see, for example, Binmore and Vulkan, "Applying game theory to automated negotiation," Netnomics Vol. 1, 1999, pages 1–9) but is still in a nascent stage. The automation of strategic choices enhances the need for these choices to be made efficiently, and to be robust against abuse. Game theory addresses these requirements.

As a mathematical tool for the decision-maker the strength of game theory is the methodology it provides for structuring and analyzing problems of strategic choice. The process of formally modeling a situation as a game requires the decision-maker to enumerate explicitly the players and their strategic options, and to consider their preferences and reactions. The discipline involved in constructing such a model already has the potential of providing the decision-maker with a clearer and broader view of the situation. This is a "prescriptive" application of game theory, with the goal of improved strategic decision making.

The dominance of process structuring that characterizes game theory makes it an inefficient tool as a stand alone methodology in this work. Particularly, considering the strength of non-cooperative branch of game theory, it is the timing and ordering of the player choices that determines the outcomes. The project submissions in the CDF case study are made over a period of time. Their submission timings and ordering are hence trivial and does not contribute much in acceptable project selections.

On the other hand, coalitional /cooperative game theory investigates more on the relative amount of power held by the various players. It is needless to say that where resources are scarce, then competition thrives. As much as all locations are competing for considerations in the fund allocations, it not how much they can bargain that yields result. This is evidenced by the fact that whatever grouping or locality get their projects implemented, equity must be seen to be done. This is ultimately one of the objectives that the committee strives to achieve.

2.7.2 *Expert Systems*

Knowledge-based expert systems, or simply expert systems, use human knowledge to solve problems that normally would require human intelligence. These expert systems represent the expertise knowledge as data or rules within the computer. These rules and data can be called upon when needed to solve problems. Books and manuals have a tremendous amount of knowledge but a human has to read and interpret the knowledge for it to be used. Conventional computer programs perform tasks using conventional decision-making logic -- containing little knowledge other than the basic algorithm for solving that specific problem and the necessary boundary conditions. This program knowledge is often embedded as part of the programming code, so that as the knowledge changes, the program has to be changed and then rebuilt. Knowledge-based systems collect the small fragments of human know-how into a knowledge-base which is used to reason through a problem, using the knowledge that is appropriate. A different problem, within the domain of the knowledge-base, can be solved using the same program without reprogramming. The ability of these systems to explain the reasoning process through back-traces and to handle levels of confidence and uncertainty provides an additional feature that conventional programming doesn't handle.

Despite the very obvious advantages of expert systems, however, there are a number of problems associated with their use when applied in a complex domain like the case study in question.

- They only interact with the users.
- Exhibits no cooperation with other systems. Most realistic and industrial problem domains are distributed and heterogeneous. It is hence difficult to build them in order to achieve this inter-operability.

Software agents are suitable for use in a wide variety of applications. They can make it much easier to build many kinds of complex systems.

Agents are well-suited for use in applications that involve distributed computation or communication between components. Agent technology is well-suited for use in applications that reason about the messages or objects received over a network.

This explains why agent-based approaches are so popular in applications that utilize the Internet. Multi-agent systems are also suited for applications that require distributed, concurrent processing capabilities.

The use of software agents in this work is most suitable since it incorporates the use of semantic web, a technology that is relatively new and now being much researched on. The project submitters are geographically separated and this makes the use of the web technology a suitable choice.

Since agents maintain a description of their own processing state and the state of the world around them, they are ideally suited to automation applications. Autonomous agents are capable of operating without user input or intervention unlike the expert systems. They are hence good in workflow management and process automation [69].

Additionally, CDF project identification problem has quite a number of dynamics. To highlight the Kenyan Case study, we have 210 constituencies each with different project submissions for every year. The project therefore requires the knowledge base to be a plug-in module which is not easy to achieve with the conventional expert systems.

Agents are not restricted to use in applications where the individual agents communicate with each other over a LAN or the Internet. In some applications it makes sense to utilize multiple agents executing on one machine and communicating with each other using some form of inter-process communication (e.g., RMI).

Again, agents are most suited to applications that require communications between components, sensing or monitoring of the environment, or autonomous operation. Since agents have the ability to reason (i.e. draw inferences), they can easily perform sequences of complex operations based on messages they receive, their own internal beliefs, and their overall goals and objectives.

In the CDF case study, drawing inference between project proposals and the selection criteria, represented as rules, is inevitable and hence a need to communicate with an inference engine which is a different component.

3 METHODOLOGY

There are many current methodologies that exist for the development of multi-agent systems including the Gaia [70], MESSAGE [17] and Cassiopera [20].

Most of these methodologies attempt to adapt object-oriented analysis and design methodologies to agent-based design [71], and in addition follow up a top-down approach.

According to M. Wooldridge 2002 in his publication, *Introduction to Multi-agents Systems*, adapting object-oriented analysis and design methodologies to Multi-agent system development has several disadvantages mainly arising from the fact that objects and agents provide different abstractions and as a result should be thought at different levels [44].

In addition, the wholly top-down approach assumed by many of the current methodologies is not sufficient for systems containing existing resources (interacting with other software applications) which need to be utilized within the multi-agent system. This chosen methodology for this work, *A methodology for the analysis and design of multi-agent systems using JADE* developed by Nikraz et al (2006) focuses on agents specifically and the abstractions provided by the agent paradigm. Furthermore, it combines a top-down and bottom-up approach so that both existing system capabilities (including those provided by software applications and people) and the application overall needs based on the requirements can be accounted for.

The design phase specifically focuses on the JADE Platform, and the concepts provided by it. Jade is an abbreviation for the Java Agent Development Framework and has been developed by the Telecom Italia Labs (TILAB) in Italy in compliance with the FIPA (Foundation for intelligent Physical Agents) specifications [31].

FIPA is a non-profit organization geared at producing standards for the interoperation of heterogeneous agents. Essentially JADE is a middle-ware (written entirely in java language using java technologies), which simplifies the implementation of Multi-agent systems by providing a set of physical tools that support the debugging and deployment phases. More details on FIPA specifications and JADE platform shall be covered in the following implementation chapter.

The choice of this methodology is justified because the system designers are able to move straight to implementation afterwards, without having to tediously adapt the results of the design phase to an agent platform of their choice.

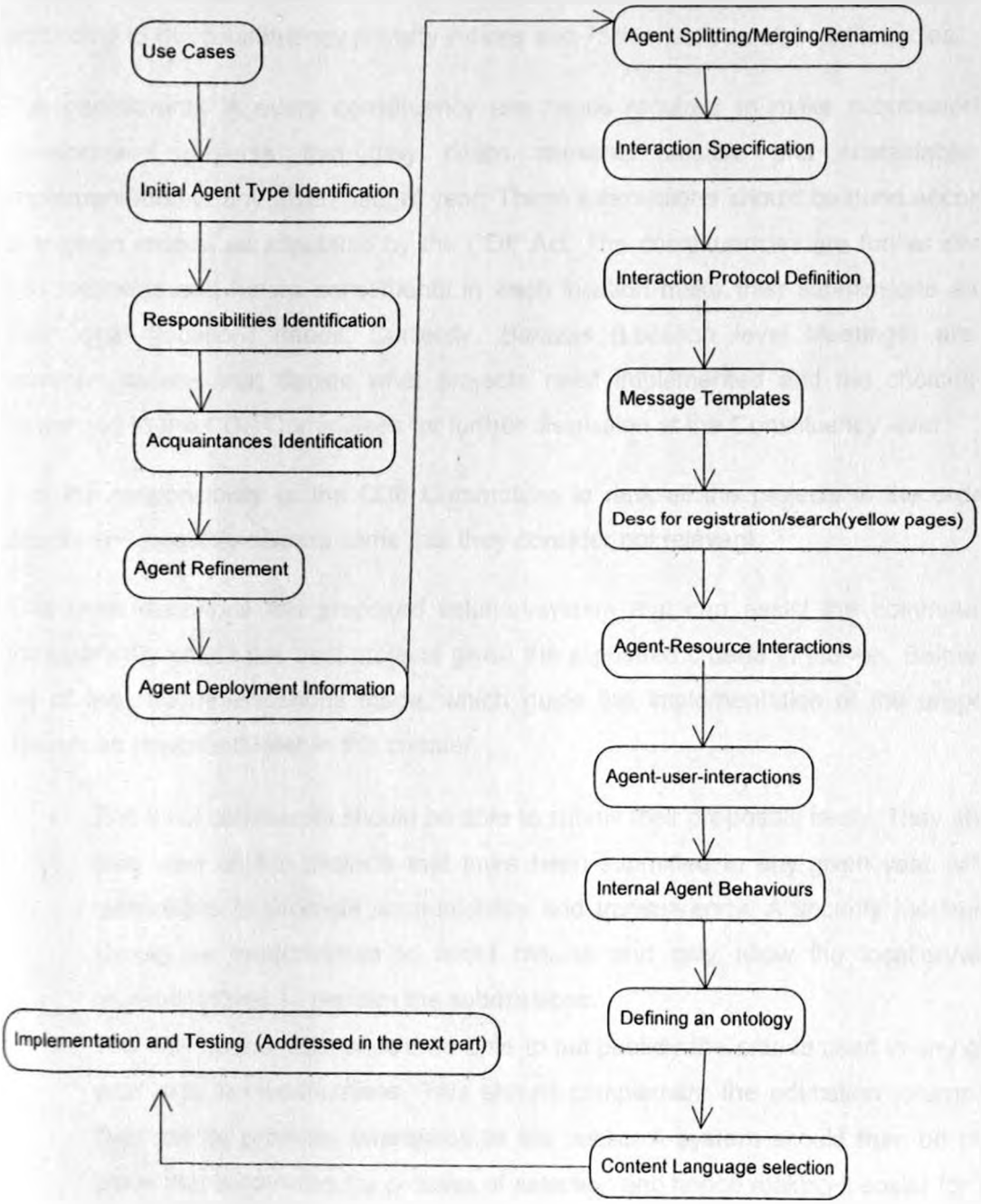


Figure 8: Overview of the Methodology

3.1 CDF CASE STUDY SCENARIO

Kenya government in the year 2003 passed a Constituencies Development Fund (CDF) Act that allocates at least 2.5% of all collected ordinary government revenue in every financial year to constituencies for development projects. This amount is allocated 25% according to the constituency poverty indices and 75% equally to all constituencies.

The constituents in every constituency are hence required to make submissions of development projects that they deem relevant, feasible and sustainable for implementation in any given budget year. These submissions should be done according to a given criteria as stipulated by the CDF Act. The constituencies are further divided into locations and hence constituents in each location make their submissions as per their local (location) needs. Currently, *Barazas* (Location level Meetings) are the common forums that decide what projects need implemented and the choices are forwarded to the CDF Committees for further discussion at the Constituency level.

It is the responsibility of the CDF Committees to rank all the projects in the order of priority and possibly discard some that they consider not relevant.

This work describes the proposed solution/system that can assist the committee to transparently select the best projects given the stipulated criteria in the Act. Below is a list of the recommendations made, which guide the implementation of the proposed system as described later in this chapter.

- The local community should be able to submit their proposals freely. They should also view all the projects that have been submitted in any given year without restrictions to promote accountability and transparency. A security mechanism should be implemented to avoid misuse and only allow the location/wards representatives to perform the submissions.
- The CDF Committee should be able to put publicly the criteria used in any given year and the justifications. This should complement the education forums that they use to promote awareness to the public. A system should then be put in place that automates the process of selection and hence making it easier for all.
- A list of the selected projects should be made public possibly with the justifications.
- A solution proposed for a given constituency should be replicated in all the other constituencies at any given time [Project selections are done annually].

3.2 ANALYSIS

The analysis of the proposed system is carried out in the following steps described below:-

3.2.1 Use Cases

Use cases are an effective way to capture the potential functional requirements of the proposed new system. Each use case represents one or more scenario that demonstrates how the system should interact with the end user or another system to achieve a specific goal.

There are a number of standards for representing use cases. The most popular is Unified Modeling Language [UML] specification [67] which is adopted for this work. Though use cases are extensively used by object –oriented practitioners, their applicability is not restricted to object-oriented according to L. Hampton et al, 1997 in their paper, a *critique of use cases [39].*

Hence it is possible to apply use cases without modification to capture the functional requirements of Multi-agent systems.

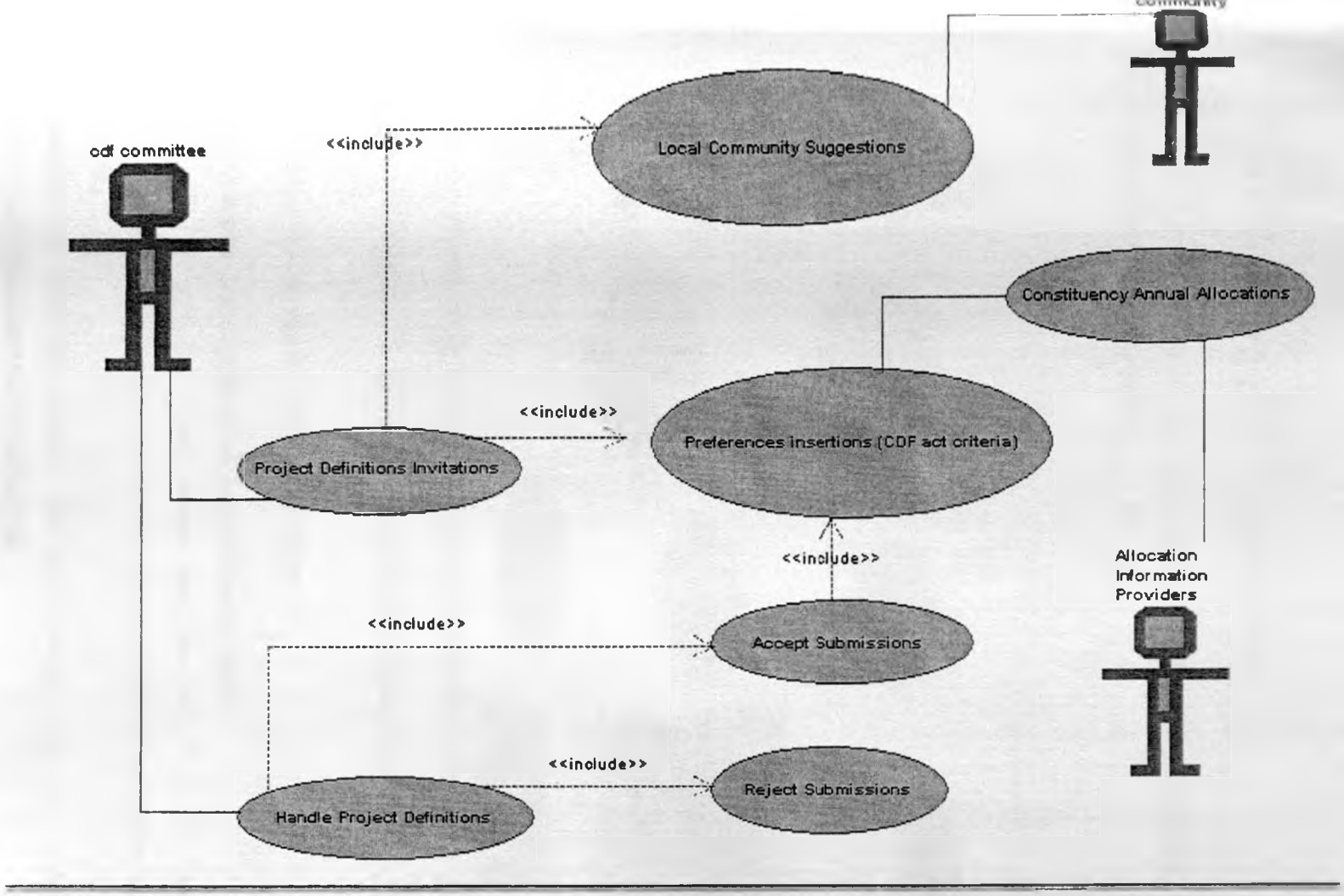


Figure 9: USE CASE diagram for the CDF project identification case study

3.2.2 Initial Agent Type Identification

This step involves identification of the main Agents types and subsequent formation of a first draft of the Agent diagram.

Rules

- One type of Agent per user/device
- One type of agent per resource (software application)

By applying the above rules to the CDF project identification case study, the initial diagram obtained is as shown below:-

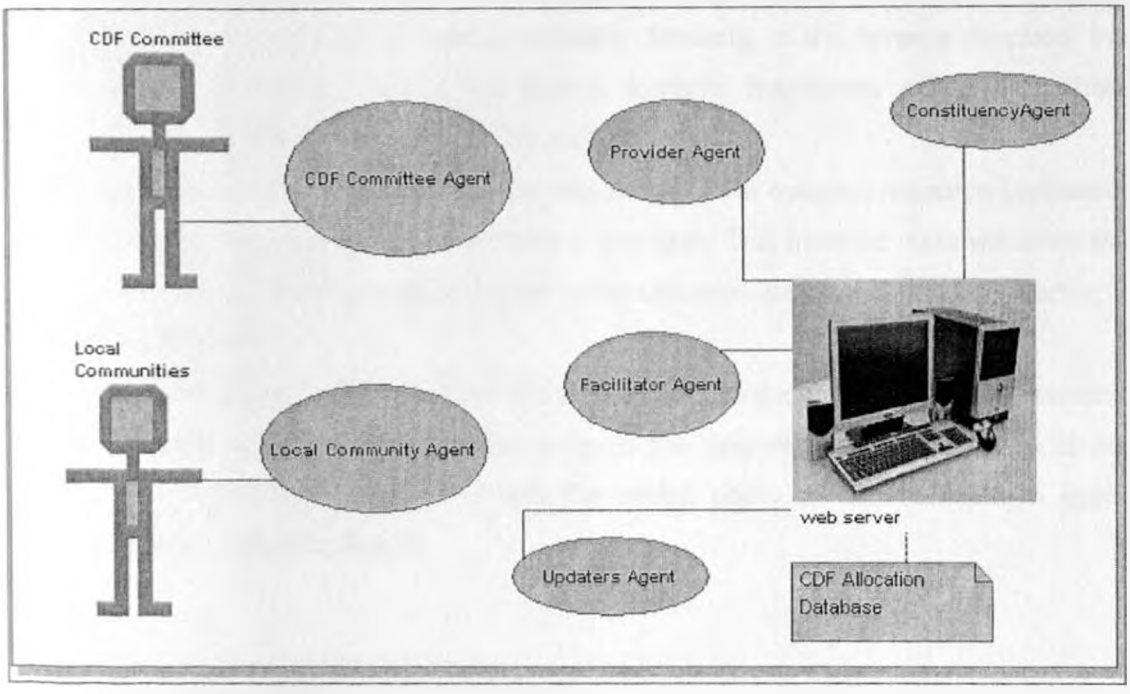


Figure 10: Agent Diagram for CDF Projects Identification Case Study

The agent diagram includes four types of elements:-

1. **Agent types:** actual agent types represented by circles
2. **Human:** People that must interact with the system under development represented by the system under development represented by the UML actor symbol
3. **Resources:** External systems that must interact with the system under development represented by rectangles.

4 **Acquaintances:** Represented by an arrow linking instances of the above elements specifying that the linked elements will have to interact in some way while the system is in operation

The way humans and other software systems are going to interact with the agents in this multi-agents system is considered and the option of the wrapper is chosen. M. N. Genesereth et al, 1994, define three techniques that account for external systems interactions with Agent systems.

- **Use of Transducers:** The transducer agent serves as an interface between an external system and other agents in the system. The transducer agent accepts messages from the agents in the system (in agents communication language), translates them into the external system. Similarly, in the reverse direction, the transducer agent receives the legacy systems responses and makes them available to the other agents in the system.
- **Insertion of a wrapper.** A code is injected into the external resource (software) provided the external resource code is available. This inserted code will allow the resource to communicate in agent communication language, thus converting it into an agent.
- **Rewriting the code:** This is the most extreme approach, which involves rewriting the code to mimic (and possibly extend) the operation and capabilities of the external software system but with the added ability to communicate in agent communication language.

3.2.3 Responsibility Identification

An initial list for each identified agent type is made of its main responsibilities.

Rules

- Derive the initial set of responsibilities from the use cases
- Consider the agents where these responsibilities are clearer first and delay the identification of responsibilities for other agents to later steps.

Table 5: Responsibility Table for CDF Projects Case Study

CDF Committee Agent	• Serve requests to provide annual allocations and avail the interface for project Submissions • Present the interface and annual allocations • Choose data required for brokering at any particular time • Request the facilitator to prepare the offerings data • Request the broker to initiate the brokering activity passing the year and the Constituency Code • Present the preferences of the projects submitted • Receive the address of the projects selected (results) after brokering
Local Community Agent	• Initiate requests for Annual Project Allocations and an interface for Project Submissions • Submit the Project Proposals • Receive a list of all the projects submitted for the year in question in that particular Constituency • Present the projects offerings for the particular year and constituency • Receive the address of the projects selected (results) after brokering
Facilitator/Gateway Agent	• Prepare data for brokering as requested by the CDF Committee. This preparation involves conversion of the project offerings from the XML format to RDF format which is the format that inference engine uses. • Prepare the rules by adding a reference to the data prepared for brokering • Do semantic and syntactically checks • Alert the Broker Agent that the data is ready for brokering
Broker Agent	• Agree or disagree to do any brokering activity • Inform the committee Agent on the willingness to broker

	• On receiving the brokering parameters from the CDF Committee Agent, requests for the data address from the local Community Agent. • Download the rules and the project offers for brokering • Initiate the brokering activity • Communicate the project offers selected after the brokering activity
Updaters Agent	• Receive the CDF Year Allocations from the Java servlet (CDF Committee Console) and make the updates on the Database
Constituency Agent	• Receive the Constituency updates (addition of constituencies) from the Java servlet (CDF Committee Console) and make the updates on the Database

3.2.4 *Acquaintance Identification*

The focus of this identification is on who needs to interact with whom and the previous Agent diagram (*Figure 10: Agent Diagram for CDF Projects Identification Case Study*) is updated by adding the acquaintance relations.

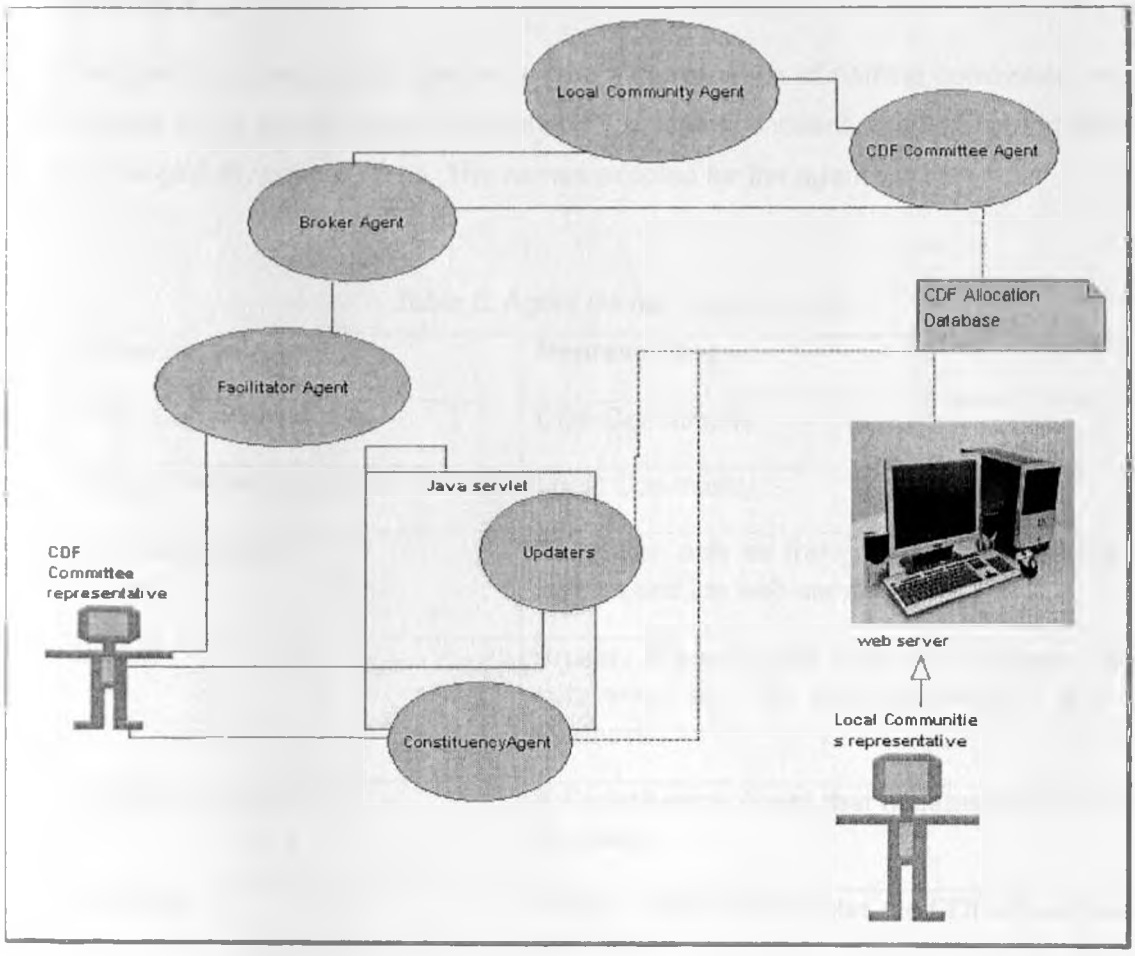


Figure 11: Agent Acquaintance Diagram

3.2.5 *Agent Refinement*

The set of agents types identified in the previous steps are refined by applying a number of considerations.

Support: - What supporting information agents need to accomplish their responsibilities, and how, when and where is this information generated/stored.

Discovery: - How agents linked by an acquaintance relation discover one another

Management and Monitoring: - whether the system is required to keep track of existing agents or starting and stopping agents on demand.

Support

All the information required in this system has already been put into consideration and therefore there is no need for any new agent type.

Discovery

The agent discovery in this system will be a combination of naming convention and adoption of the yellow pages mechanism. The naming convention entails giving each agent a globally unique name. The names adopted for the agent will be:-

Table 6: Agent Global Names Used

Name of the Agent	**Representing**
CDF_Committee Agent	CDF Committees
Local_Community Agent	Local Community
Gateway_Agent	Facilitator, acts as the gateway between the agent system and the web server
Broker	Broker, brokering the activities between the CDF committee and the local Community and giving feedback
ConstituencyAgent	A Constituency Agent that updates additions on the Database
Updaters	Update Agent that updates the CDF Allocations year after year

As far as the yellow pages mechanism is concerned, the discovery will be made on the basis of the characteristics e.g the services the agents provide.

Below are the characteristics that go with each agent:-

Table 7 : Services offered by each Agent aiding in Discovery

Agent Name	**Agent services / Characteristics**
CDF_Committee	Requesters, Agents requesting for project submissions
Local_Community	Providers , Agents providing the project

	submissions
Broker	Brokering-services- Brokering for required services
Gateway_Agent	Gateway services – Gateway between the web server and the agent system
Updaters	Allocations Update Services between the web server and the agent system and the database
ConstituencyAgent	Constituencies Update Services between the web server and the agent system

A yellow pages mechanism can be fully distributed across all agents or centralized with a single agent. This system adopts a centralized approach since the design is done with the JADE platform as the target. This approach completely maps to the discovery facilitator agent provided by JADE.

Management and Monitoring

Agent types may be added in a system to address issues like monitoring agent faults and restoring them. Since the number of agents in this work is small, no new agent types are required to be added.

3.2.6 Agent Deployment Diagram

The final artifact for the analysis stage is the Agent deployment diagram. This diagram is not intended to give any detailed information about deployment. Its sole purpose is to highlight basic deployment requirements that are referred to during design when applying considerations such as agent splitting/merging or communication efficiency.

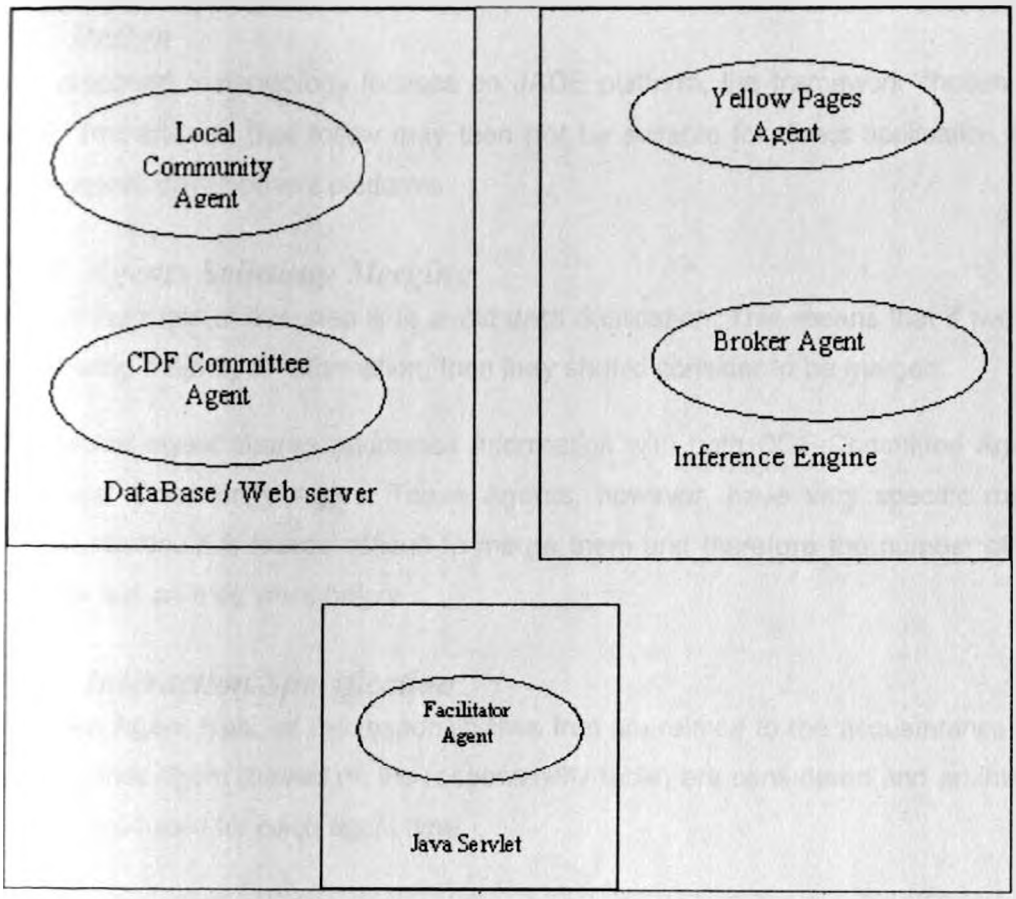


Figure 12: Agent Deployment Diagram

3.3 Design

The proposed methodology focuses on JADE platform, the framework chosen for this work. The artifacts that follow may then not be suitable for direct application to other Multi-agents development platforms.

3.3.1 Agents Splitting/ Merging

One of the rules of this step is to avoid data duplication. This means that if two agents are sharing majority of information, then they should consider to be merged.

The broker agent shares addresses information with both CDF-Committee Agent and the Local Community Agent. These Agents, however, have very specific roles and representation. It is hence difficult to merge them and therefore the number of Agents remains just as they were before.

3.3.2 Interaction Specification

For each Agent type, all the responsibilities that are related to the acquaintance relation with another agent (based on the responsibility table) are considered and an interaction table is produced for each agent type.

Table 8: Interaction Table for the CDF Projects Identification

Interaction	Responsibility	IP(Interaction Protocol)	Role	With	When/ Description
Request for allocations and for Projects Submission interface	1	SimpleAchieveREInitiator	I	Provider Agent	At the Start up Local Community Agent initiates a request
Respond to the allocations request by opening them	2	SimpleAchieveREResponder	R	Requester Agent	Presents the Projects Fund Allocation list and the interface for submission, after receiving the Allocation request
Inform the Local Community Agent	3	ACL Message -Inform	R	Requester Agent	After Project interface is presented
Pass Parameters for the Brokering activity(Agent -User Interaction)	4	SimpleBehavior	R	Facilitator Agent	When the CDF Committee representative submits the brokering parameters
Respond to data Preparation Requests	6		R	Java Servlet Resource (Facilitator Agent)	After Parameters of the Brokering Activity are submitted
Alert after the data preparation is done	7	ACL Message Inform	I	Facilitator Agent	After Data preparation is over
Receive data preparation alert and brokering credentials and receive request for Brokering activity initiation			I	Broker Agent	After being alerted by the Facilitator agent of data readiness
Asking for Project offers		FIPA Query-IF REF		Broker Agent	After receiving the Brokering request

Respond to brokering request by agreeing or disagreeing. On Agreement, Download Rules and offers		SimpleAchieveREResponder	R	Broker Agent	On checking whether the offers are available or not
Initiate Inference		SimpleAchieveREResponder	I	Broker Agent	
Inform, Pass the Results Address and Display Results to both the Providers and Requester Agents		ACL Message Inform	I	Broker Agent	when the inference is Done
Receive address of results			R	Requester Agent/ Provider Agent	Marks the end of agents interactions

Key

Agent Name	Agent Service Name
The Local Community Agent	Provider Agent
The CDF Committee Agent	Requester Agent
Gateway Agent	Facilitator Agent
I	Initiator
R	Responder

1. The local community agent requests for annual project allocations and an interface for project submissions from the CDF Committee Agent.
2. The CDF committee responds by retrieving the allocations from the Project Allocation database and presents them through the browser.
3. The CDF Committee Agent alerts the local Community about their availability
4. The Local Community Representative submits the Project submissions to the web server which is the repository
5. After the submissions are done, the CDF Committee representative chooses the particular project definitions that needs brokered. The choices made are the particular constituency and year of allocation that needs brokered. He also chooses the kind of brokering results that he needs.
6. The brokering particulars are then submitted to a JAVA servlet through a Java server Pages Platform, where the Gateway agent (otherwise known as facilitator) resides. The facilitator agent prepares the Project Submissions by converting them from XML to RDF format (the format used by the engine). It also prepares the rules by adding a reference to the RDF file of the chosen offers and the conclusions that will be exported.
7. The Facilitator Agent alerts the CDF Committee Agent of the readiness of the data for inference.
8. The CDF Committee Agent requests the broker Agent to initiate brokering by sending the credentials of the projects for brokering.
9. If the broker Agent agrees to do brokering, it then requests for the address of the offers from the local agent and then initiates the brokering.
10. After the brokering is complete, the broker Agent sends the results to both the CDF Committee Agent and the Local Community.

3.3.3 Ad-Hoc Interaction Protocol Definition

Whenever possible, existing interaction protocols defined by FIPA should be adopted [30]. However, it is often the case that the Interactions requires Ad-hoc interaction protocol to be defined (i.e. when none of the FIPA defined interaction protocols are deemed adequate). Since all the agents of the brokering system use FIPA Request Interaction Protocol, then no Ad-hoc interaction Protocols is designed for this work.

3.3.4 Message Templates

All the interaction protocol roles identified in the previous step are implemented as JADE behaviors (*description in the preceding sub-section*). In this step, suitable *MessageTemplate* objects are specified to be used in these behaviors to receive incoming messages, and these templates are added to the rows of the interaction table.

Since the entire Message Templates used are FIPA-REQUEST, then the deliverable of this design step is [*Table 8: Interaction Table for the CDF Projects Identification*] with an addition field of message Template with a value "*FIPA-REQUEST*" for all values.

3.3.5 Descriptions to Be Registered /Searched

In this step, the naming convections and the services registered/searched by agents in the yellow pages catalogue maintained by the JADE Directory Facilitator are formalized (where relevant). A class Diagram form is proposed to describe the service registration/searches as shown in [*Figure 13: Service Registration/Search Diagram 1*]

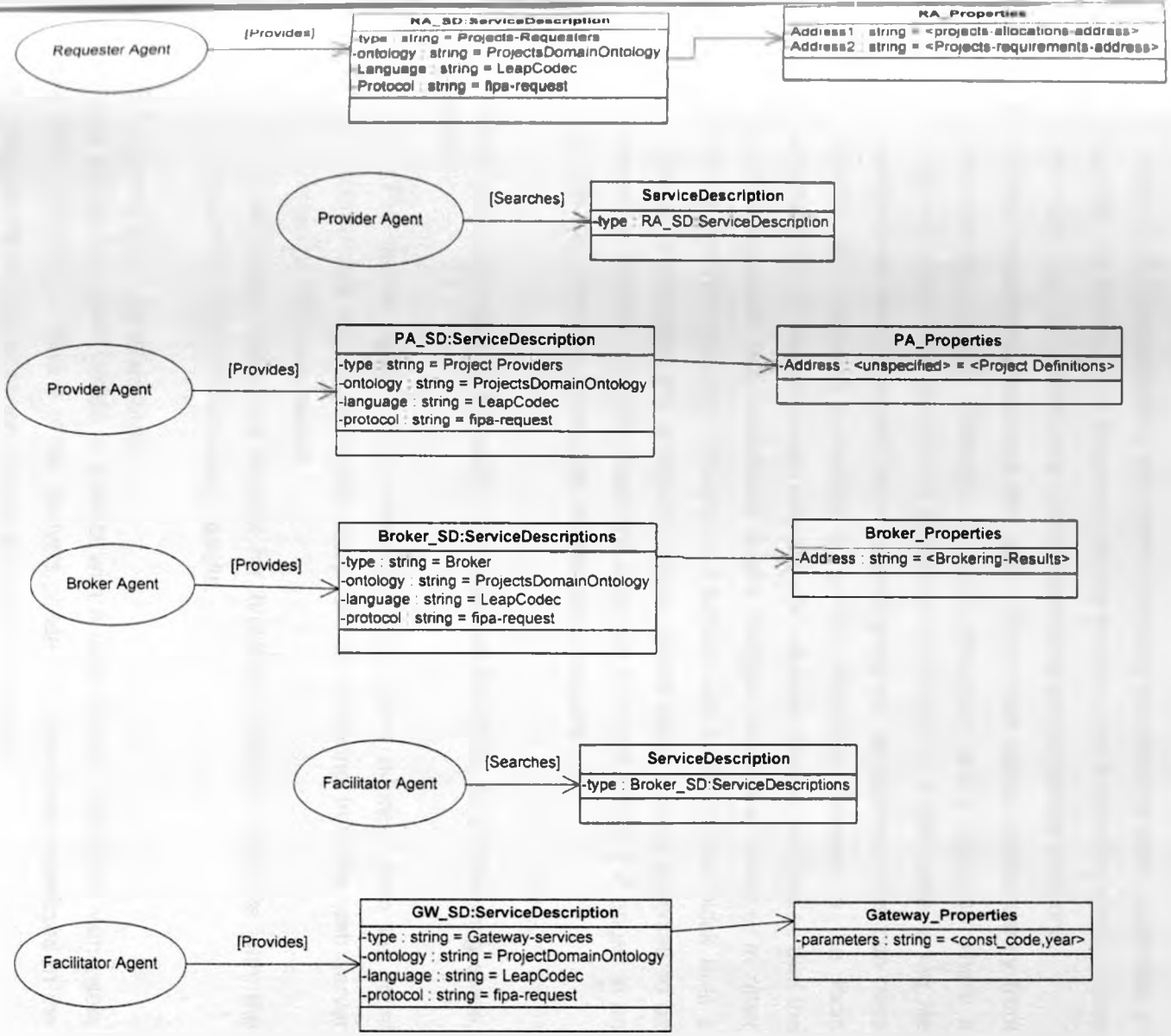


Figure 13: Service Registration/Search Diagram 1

3.3.6 Agent-Resource Interaction

It is often the case that one or more agents in the system must interact with external resources such as databases, files storing information, or legacy software. Agents interacting with external resources have been identified in [*Initial Agent Type Identification*] of the analysis phase, and are expressed in the agent diagram by an acquaintance relation with a resource element. Such resources can be classified into two main categories:

- ***Passive resources:*** resources that change their status only as a consequence of some stimulus issued by the agent controlling the resource itself. Examples of passive resources are a database fully controlled by the interacting agent, a data file in the local file system or a C library providing computational functions.
- ***Active resources:*** resources that may change their status independently from the controlling agent. Examples of active resources are a database where a human operator (or an external program) can insert or modify data, a log file continuously filled (updated) by an external program, an appliance that can raise alarms and software controlling a sensor detecting changes in the local environment. Active resources may provide a listener-based interface so that the controlling agent can immediately detect changes inside the resource. In other cases, the resource may provide an interface with methods that block until a change is detected, e.g. a network socket where some data is expected to be received. Finally, in certain cases the only way to detect relevant changes in an active resource is to periodically poll the resource itself.

In this work, examples of the active resources are identified, in the following steps, where:-

- The broker agent, after receiving project offers address from the local community agent, downloads the offers for brokering from the web server where the files are hosted.
- The broker agent, on request for brokering initiation, interacts with the inference engine for brokering results.

3.3.7 Agent-User Interactions

In many cases, an agent needs to interact with a user. Agents interacting with users have been identified in Step 4 of the analysis phase (*3.2.4*) and are expressed in the agent diagram by an acquaintance relation with an actor element.

There are several ways a human user can interact with a piece of software such as a JADE agent. Here, the focus is on the graphical user interface (GUI), which is by far the most commonly used type of user interface. Two cases are distinguished:

- A local GUI typically implemented using Swing, the Abstract Windowing Toolkit (AWT) or some other graphical toolkit.
- A Web GUI implemented using JavaServer Pages (JSP) technology.

GUI can be viewed as an active resource and fit in the design phase step 6 but there are elements which make the GUI case more complex with respect to a generic external resource. Thus, a small discussion of local GUI and the web GUI cases follows in more details in the next sub-section.

Local GUI

With the Local GUI, the agent and the GUI must typically work on the same data (the agent to perform its domain specific tasks, and the GUI to present the data to the user), but must organize this data in different ways.

As an example, the developer may want to organize a list of elements as a tree to support fast searches based on some criteria, while a java.awt.List would be the best structure to keep the same data from an AWT GUI point of view. The risk, therefore, is to end up with duplication of data and consequent problems in keeping this data consistent.

Using a toolkit such as Swing, which implements the Model-View-Controller architecture [65] , it is possible to overcome this problem by:

- Storing data inside the agent in structures that are suitable to accomplish agent tasks.
- Having structures fully synchronized.
- Having the structures to implement the proper swing model interfaces and link them to the related graphic classes inside the GUI module.
- Defining proper renderer classes when necessary.

With reference to the example above, by applying the above mentioned recommendations, it is possible to end up with a tree structure optimized to perform fast searches and it would be made to implement the

javax.swing.ListModel interface. Inside the GUI module, a *javax.swing.JList* object would be kept linked to the *ListModel* implemented by the tree structure.

Unfortunately, the swing model interfaces are often quite complex, and as a consequence, following the presented approach may not be straight forward. Therefore, the decision to adopt this approach was found not fitting.

Web GUI based on JSP

This involves developing a web interface to a JADE multi-agent system implemented using Java Server Pages (JSP) technology [47]. The proposed approach is based on the creation of a dedicated agent inside the JSP, acting as gateway between the JSP and the JADE world (i.e. the other agents in the system).

In particular, it is suggested to use the *JADEGateway* and *GatewayAgent* classes included in the jade.wrapper package in JADE 3.6[10], which hides from JSP developer's point of view the volatile nature of the agent acting as gateway. This is to say that an agent created inside a JSP is not guaranteed to stay alive forever, since the JSP container may suddenly shut it down. Therefore, managing a direct pointer to the agent may become complex since one always needs to take into account that the agent may have died in the meantime. The *JADEGateway* is essentially a proxy that hides to JSP developers the above complexity and automatically manages agent re-creations when needed.

There is a singleton *JADEGateway* object that embeds a container and a *GatewayAgent* on it. The *JADEGateway* provides the *execute()* method by means of which generic objects can be submitted to the *GatewayAgent* inside the *JADEGateway*. The *processCommand()* method of the *GatewayAgent* must be redefined to process objects passed to the *JADEGateway execute()* method. The *execute ()* method returns as soon as the GatewayAgent processing the Object passed to the execute() method calls the releaseCommand() method specifying that Object as argument.

In this work, an agent-user interaction exists where the CDF Committee representatives chooses the Brokering parameters which must be passed to the Agent system. This

[10] The current Version of JADE (25 November 2008)

choice of Web GUI is preferred since it embraces the web technology, one of the objectives of the study.

The classes mentioned in the JSP based Web GUI are exploited by means of CDF Projects domain specific "*bean-behaviours*" as arguments when invoking the *JADEGateway.execute()* method. The term "*bean-behaviours*" refers to behaviours with proper get and set methods for all specific year and Constituency Code parameters to be passed by the JSP to the JADE system. There are no parameters needed to be passed from the JADE system back to the JSP.

3.3.8 Internal Agent Behaviors

The actual job an agent has to do is typically carried out within the agent's "behavior(s)". This design step maps the responsibilities identified in the analysis phase and maps them to agent behaviors.

Table 9: Internal Agent Behaviors

Responsibility	**Behavior class**
Request for Allocations and interface Submissions	*AllocateRequest* Behavior a subclass of *SimpleAchieveREinitiator*
Response to Allocations Request	*HandleAllocations* Behavior a sub-class of *SimpleAchieveREResponder*
Pass the Parameters for the Brokering Activity	"*Bean-Behaviour*" class of the *JADEGateway* Agent, *MessageReceiver*, a subclass of the *SimpleBehaviour*
Response to Data Preparation Requests	GateWay Servlet, a subclass of the *HttpServlet*
ReceiveMessages, ask for offer addresses and initiate Brokering Requests	*HandleInitiateBrokering* SubBehaviour of the Sequential Behaviour , extending *SimpleAchieveREResponder*
Display the Brokering Results	*DisplayResults* SubBehaviour of the Sequential Behaviour, extending *OneShotBehaviour*

3.3.9 Defining an Ontology

When agents in the system interact, they exchange information that refers to entities, abstract or concrete, that exist in the environment agents reside in. These entities may be primitive, such as a String or a number, or may have complex structures defined by templates specified in terms of a name and a set of slots whose values must be of a given type. These complex entity templates are referred to as Concepts. Below is a sample of the concepts used in CDF Projects Identification Case Study. The complete ontology used in this work is presented in the implementation chapter.

Offerings

- imp-cost (int)
- op-cost (int)
- project-name (string)
- ann-allocation (int)
- complementary-proposal (string)

CDF-Allocations

- Allocation-address (string)

CDF-RULES

- Rule-address (string)

4 AGENT SYSTEM IMPLEMENTATION

4.1 Introduction

This chapter discusses the key features of FIPA (Foundations for intelligent physical agents) specifications in order to position and define widespread concepts like Agent, behavior and communication in a reference context and then an in-depth examination of the proposed CDF projects brokering system.

In more details, the chapter is articulated as follows: In section 4.2, the standard architecture designed by FIPA for an agent platform is examined, with a presentation of the JADE platform along with its most important characteristics as outline in the FIPA specifications. In section 4.3, the overall architecture of the system is presented and a description of the modules of the architecture along with the interactions and actions that are performed during the brokering case. The next part of the section presents the modeling of the CDF projects brokering domain by means of a message content ontology. As in the case of brokering, the service/product is of great importance and it is modeled along with its characteristics in separate domain ontology, using the RDF-RDFS technology.

Subsequently, a description is made of the protocol used among the agents who participate in brokering. As a next step, the brokering scenario is fabricated and a presentation of the CDF Committee requirements preferences and the presentation of the offers.

4.2 The FIPA Abstract Architecture

The development of agent systems has a recent history. Little time has elapsed since the scientific world perceived the promise of using the agent paradigm to solve a great variety of problems. This realization prompted researchers to design independently, their own infrastructures on which to activate their own agents.

The resulting working proposals were often optimal, very efficient for a specific problem domain, but not devoid of defects. The programming language, the communication paradigm and the other technical details generally made the frameworks unsuitable for purposes other than those for which a given approach was originally conceived.

The total absence of genuine attention towards the system design and development process (and consequent documentation) hindered the growth and maintenance of these applications. This brought about the need for standardization.

The importance for standardization is such a pivotal issue that an international organization, Foundation for Intelligent Physical Agents (FIPA) was founded to promote the intelligent agents industry by openly developing specifications supporting interoperability among agents based applications.

The work of the FIPA focuses mainly on the definition of the agent platform (AP); this is defined as the physical infrastructure where agents can be deployed.

Most of the standardization work, therefore, concerns the definition of some key-points that an AP has to comply with. Thanks to these standards, agents living in two or more FIPA compliant platforms are able to communicate and interoperate with each other.

The principal aspects defined by FIPA specifications are:

a. The message level, which describes the composition of a message (expressed with the Agent Communication Language), a set of primitive messages with a specific semantic (referring to the speech acts theory [63]) and the sequence of speech acts that compose a correct communication (the Agent Interaction Protocol);
b. The transport level, which details how a message has to be moved from a sender to a receiver;
c. The service level, which defines the mechanism used by each agent to offer its own services and to discover the services offered by other agents in the platform.

4.3 Architecture Overview

One of the main goals of FIPA specifications is to promote inter-operability between agent applications and agent systems and this is achieved by defining the Abstract Architecture Specification. This is a collection of architectural elements that characterize each FIPA-compliant platform. The term 'abstract' means that the architecture defines only some functional requirements but it is neutral about the technologies used to achieve them.

The agent-platform architecture (represented in ***Figure 14***) is centered on three mandatory components:

- DF (Directory Facilitator) component,
- AMS (Agent Management System) component,
- MTS (Message Transport System) component

All of these elements will be examined in more detail in the paragraphs that follow.

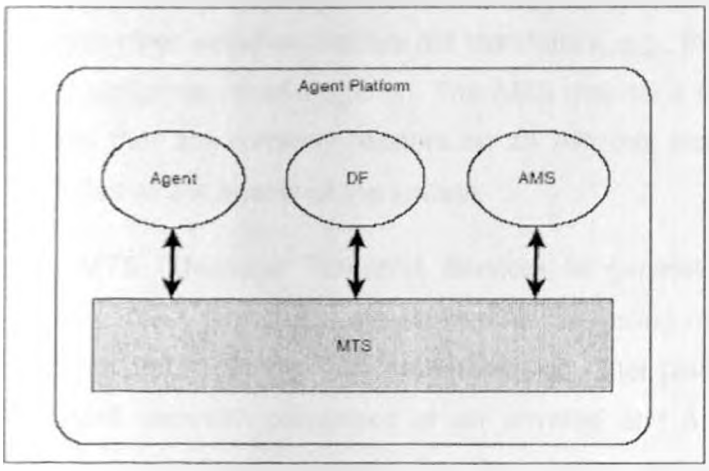


Figure 14: Overview of the FIPA abstract architecture

4.3.1 Infrastructures for Agent Interactions

The **DF** component of an AP provides the yellow pages service to agents 'living' on that platform. It defines the support for agents' collaborations centered on the concept of service where a service is defined as an activity that an agent performs on the request of another one belonging to the same community. Agents may interact with the DF in two different ways: registration and search. To advertise that a specific service is available to the community the provider agent can register it in the DF with a significant name. Generally an agent can provide more than one service, each one of them being registered in the DF with a different name.

An agent has no a-priori knowledge about the other agents of the system. In order to discover if another can be of any help in reaching its own goal(s), the agent may search the DF. Consequently, the agent obtains a vector of DF-entries; each entry contains the univocal address of an agent of the system that performs that service.

Generally speaking, the result is a vector, because more than one agent can provide the required service.

The **AMS** is responsible for managing the operation of an AP; the main functionalities of the AMS are the creation, deletion and life-cycle management of agents. The AMS may support other activities that are not mandatory, e.g., the migration of agents to and from other platforms (mobile agents). The AMS maintains the physical index (AID) of all the agents that are currently resident on an AP; this index is an address that univocally identifies all the agents of the system.

The **MTS** (Message Transport Service) is generally invisible to agents and their developers. It provides a mechanism for delivering messages among agents within a platform and to agents that are resident on other platforms. Messages are coded in a standard structure composed of an envelop and a payload. The envelop contains transport information needed for the correct delivery of the message. Transport information could specify a network protocol like HTTP or SMTP and the address of the agent if it is reachable using that protocol (something like *www.CDFprojects.net/abc* or *agentname@host.domain.org*). The payload record is coded in a language called Agent

Communication Language (ACL) and it contains the information content that is to be delivered.

4.3.2 Agent Social Relationships

Social relationships are among the most important characteristics of agents. A multi-agent system is composed of a number of autonomous and interacting agents and it is frequently represented as a well organized society of individuals. In this context each agent has its own personal goals and plays one or more different roles during its life to interact with other community members.

Agents interact through messages only and, most commonly, their interaction is composed of a series of messages, thus composing what we define as a conversation. It is more correct to think about an agent interaction as a conversation rather then one

simple message. A conversation, and specifically a FIPA conversation, is essentially composed of one or more messages. As already mentioned, each message needs a transport infrastructure in order to be delivered. This allows the effective implementation of a conversation but does not ensure any usefulness for it. In order to add a semantic value five important concepts must be adhered to (see *Figure 15:* ***Structural diagram illustrating the elements constituting a FIPA Message and the relationship among them***): ontology, content, content language, communicative act and agent interaction protocol (AIP).

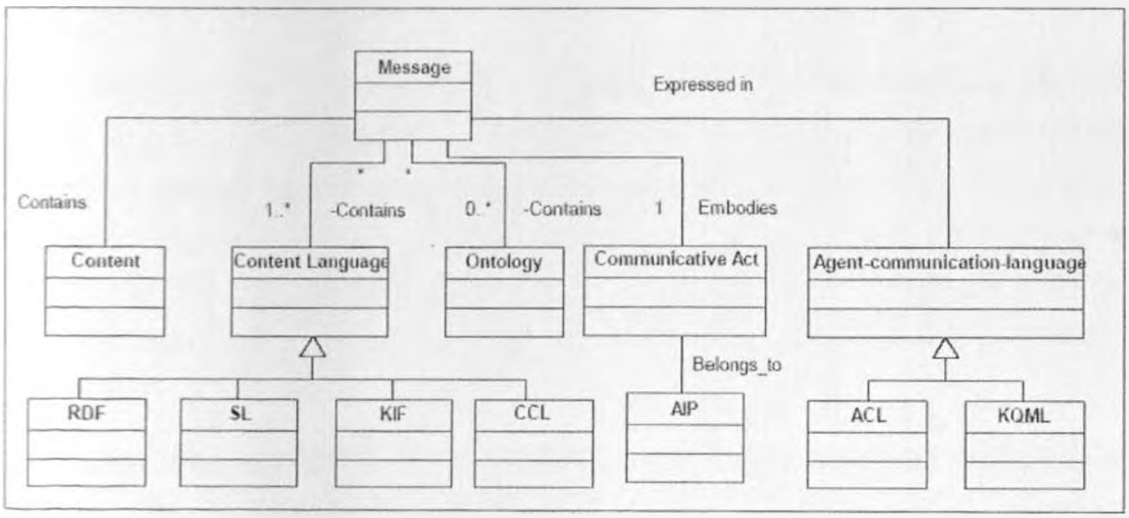


Figure 15: Structural diagram illustrating the elements constituting a FIPA Message and the relationship among them

Modeling the Communication Semantic with an Ontology

An ontology is a representation of the categories that exist in a specific domain; it is a vocabulary used to describe the terms and the relationships among them with a subject matter. Ontology allows the specification of the type of terms an agent may handle and what type of manipulation and reasoning it is able to perform on them. Referring to the same ontology, two agents can interact without the risk of a misunderstanding. They refer to the same set of concepts and, if they adopt the same (content) language, the communication will be meaningful for both of them. On the contrary, the lack of a common ontology introduces the risk that a term used by an agent with some specific significance will be interpreted by another in a different way, thereby jeopardizing agents' interaction and the entire system's performance.

Ontology defines the meaning of categories and the relationship among them but in order to manage it agents need a language that can represent both the ontology structure and content. In many approaches, the ontology structure is composed of three kinds of elements (concepts, predicates and actions), and the associations among them.

Many authors have dealt with the representation of the ontology using Unified Modeling Language (UML). Examples of these works are *CraneField S and Purvi M (1999) [23]* and *F.Bergenti and A.Poggi (2000) [14]*. In this work, an adoption of Altova UModel notation that uses a UML class diagram is made. Concepts, predicates and actions are represented as classes. ***Figure 16*** represents a portion of the ontology designed for the CDF Projects Identification case study.

As an example, Project-Offerings class represents a concept of the ontology; a concept stands for one of the categories of the specific domain, and in this example, project offerings represent the project proposals issued by the local community. It has some attributes e.g imp-cost, which is the implementation cost for the project proposed by the constituents. A concept may be related to other concepts; for example project offerings have yearly allocations i.e. the amount of money allocated to a particular constituency in a given year.

A predicate represents a particular statement or belief surrounding some concept, as in the case of the ***AreAvailable*** predicate shown in ***Figure 16***. This is used to announce that the CDF rules are available in a given location for use.

An action indicates the type of operation that can be performed on elements of the ontology, thus possibly provoking some changes to the internal knowledge of the agent. ***EngineInvoker*** in ***Figure 16*** is an example of an action specifying the request from one agent to another to invoke the inference engine for some project offers.

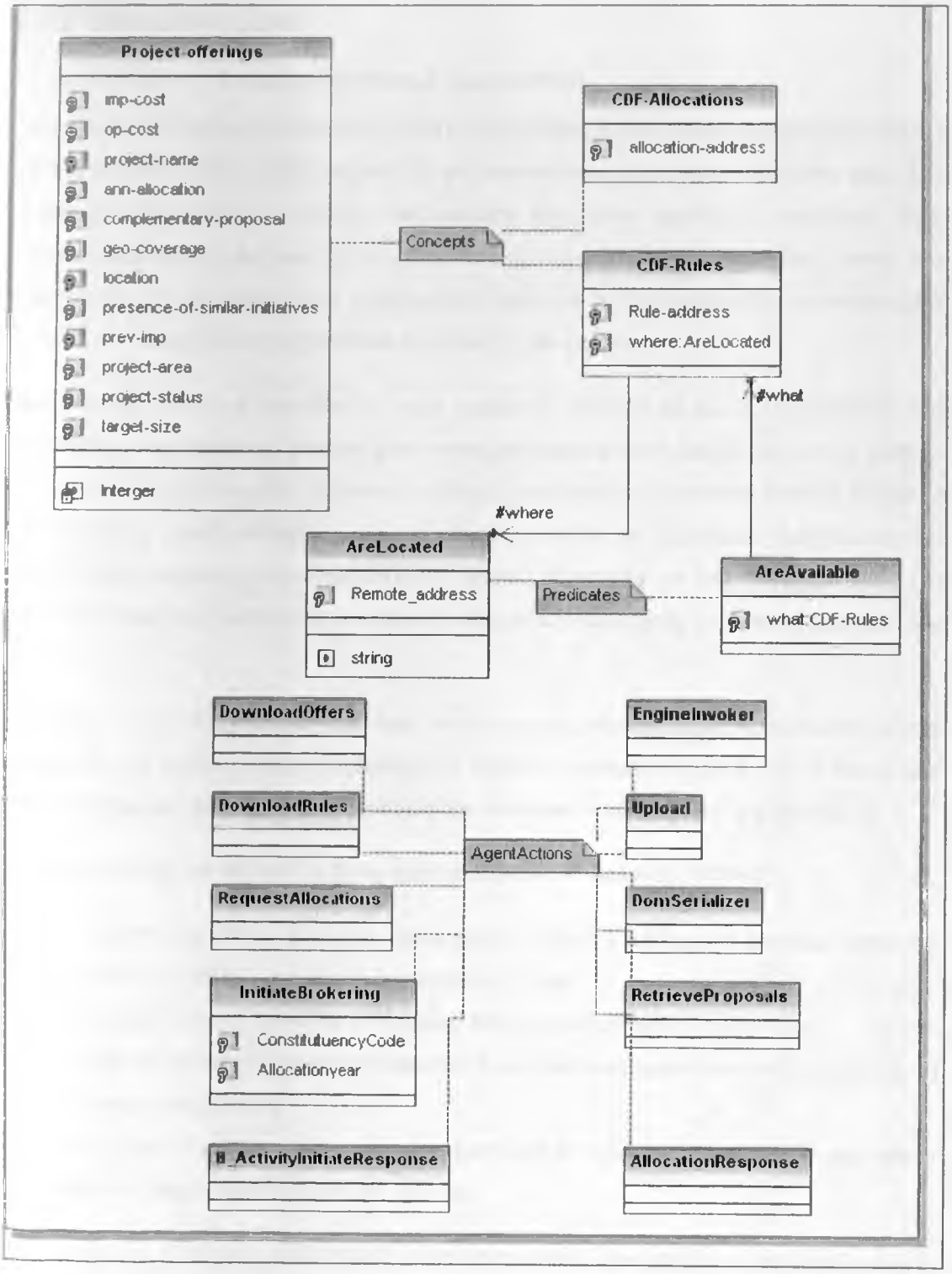


Figure 16: Example of ontology expressed using a UML class diagram

4.3.3 Interaction Level

Coordination - Cooperation versus Competition

Coordination is defined in many ways but in its simplest form it refers to ensuring that the actions of independent actors (agents) in an environment are coherent in some way. The challenge therefore is to identify mechanisms that allow agents to coordinate their actions automatically without the need for human supervision, a requirement found in a wide variety of real applications. Cooperation refers to coordination with a common goal in mind; non-antagonistic agents that succeed or fail together.

Research to date has identified a huge range of different types of coordination and cooperation mechanisms, ranging from emergent cooperation (*which can arise without any explicit communication between agents)*, coordination protocols (*which structure interactions to reach decisions*) and coordination media (*or distributed data stores that enable asynchronous communication of goals, objectives or other useful data*), to distributed planning (*which takes into account possible and likely actions of agents in the domain*).

Interaction and cooperation with one another are amongst the fundamental social capabilities of agents. The co-operation of agents depends on the kinds of tasks and activities they are expected to do, and the environment in which they are operating.

Cooperation may be defined in three ways according to Adler et al (1992):

(1) Agents (systems) that have been placed under a framework because they can perform problem solving in a common domain,
(2) Agents working together to improve their individual performance, and
(3) Agents working together to improve the collective performance of the system to which they belong.

In the CDF Case study, the third type of cooperation is important because it brings about the question of dependency between agents.

According to J. E. Doran et al (1996), such cooperation can either be *communicative* in that the agents communicate (the intentional sending and receiving of signals) with each other in order to cooperate or it can be *non-communicative*.

In the latter case, agents coordinate their cooperative activity by each observing and reacting to the behaviour of the other. The Broker agent, in this work, responds to the

brokering request from the facilitator agent by initiating inference. It is only after receiving the parameters through messaging that the broker initiates the brokering activity.

Additionally the CDF Committee agent relies on the agreement of the broker to perform brokering, a form of cooperation, for any brokering results to be obtained.

The actions of multiple agents need to be coordinated because of dependencies between agents' actions, their need to meet global constraints, and no one agent has sufficient competence, resources or information to achieve such system goals.

Examples of coordination include supplying timely information to other agents or ensuring that the actions of agents are synchronized. In an environment of heterogeneous agents coordination is a difficult task because agent must reconcile conflicting or incomplete views of the environment they are acting upon or interacting with.

Intentional communication can take at least two forms—agents can deliberate or they can negotiate. In deliberative systems agents jointly plan their actions so as to cooperate with each other. Negotiating systems have an aspect of competition. The definition of competition, the interaction compared with cooperation in this write up, is the ability of an agent to interact with other agents where the success of one agent implies the failure of others (the opposite of cooperation). Very small aspects of competition are found in this research, if any, that are not designed for but may evolve through complex social relationships.

In this work, scheduling mechanisms have been implemented where data objects are shared like access of the input file by both facilitator and broker agent to avoid any competition which would other wise cause resources synchronization issues.

If an agent needs to communicate with other agents, it needs to know the underlying model of the other agents. The model deals with heterogeneity, domain knowledge, designs and skills of the agents. Therefore a global standard data communication which is accessible by every entity within the overall system is required. For a large category of agent systems, the main mechanism for knowledge sharing is the Agent Communication Language (ACL), R. Cost, et al (2001).

An ACL relies on a three-layer conceptual breakdown of the knowledge sharing problem.

The concepts originated in the work of the Knowledge Sharing Effort (KSE) as stated by R. Neches et al (1991) and have found their way into Knowledge Query Manipulation

Languages (KQML) [6], Y Labrou and T Finin (1997), the first ACL, and in FIPA ACL, the ACL proposed by the Foundation for Intelligent Physical Agents (FIPA), a standardization organization in the area of agents.

The CDF Semantic brokering Case study uses the ACL as the communication language and all the agents use Lightweight Extensible Agent Platform (Leap) as the Content language in their communication. A simple formulation of the knowledge sharing between intelligent agents "the expressions in a given agent's native language" should be understood by some other agent that use a different implementation language and domain assumptions.

The three-layered approach distinguishes between *propositions* and *propositional attitudes*. The first and the second layers are concerned with sharing the meaning of *propositions* and the third layer is concerned with sharing the meaning of *propositional attitudes*.

So, the first layer is that of (*syntactic*) translation between languages in the same family (or between families) of languages. The second layer is concerned with guaranteeing that the *semantic content of tokens* is preserved across applications; in other words, the same concept, object, or entity has a uniform meaning across agents even if different "names" are used to refer to it. Every agent incorporates some view of the domain (and the domain knowledge) to which it applies. This background knowledge is the ontology. More formally, an ontology is a particular conceptualization of a set of objects, concepts and other entities about which knowledge is expressed and of the relationships that hold among them. Ontology consists of terms, their definitions, and axioms relating them Thomas R. Gruber (1993); terms are normally organized in taxonomy. *ProjectDomainOntology* is the name of the ontology used in this work, designed using Protégé[11]

The final layer addresses the communication between agents. This is not merely about transporting bits and bytes between agents; agents should be able to communicate complex "attitudes" about their information and knowledge content. Agents need to ask other agents, to inform them, to request their services for a task, to find other agents who can assist them, to monitor values and objects, and among other things.

---

[11] *Protégé is a free, open source ontology editor and knowledge-base framework.*

Agents issue requests by specifying not a procedure but a desired state in a declarative language, i.e., in some ACL. Assuming effective translations between their respective representation languages, with shared ontologies, an ACL allows them to share their knowledge content.

An ACL offers agents building blocks for coordination. To the extend that coordination is communicative, as is often the case in heterogeneous agent communities, agents can use the propositional attitudes supplied by ACLs in order to sustain the complex interactions with other agents that are necessary for coordination.

4.4 Implementation Platform: JADE

FIPA describes an abstract architecture that cannot be directly implemented; since the main focus of these specifications regards agent interoperability, not many details are provided on the platform implementation details.

On this basis a great number of different solutions have been proposed over the last years, a list of which can be found in the FIPA Website [31]. Among the mostly widely used are FIPA-OS, JADE (Java Agent Development Framework) and Zeus. In this section, the JADE Agent platform is briefly analyzed in order to illustrate some of the specific implementation details.

For the development of the CDF Projects case study brokering system, the JADE platform is chosen [13] [46]. According to Nguyen T. Giang et al (2002), it exhibits very interesting features compared to other Multi-Agent Frameworks. In addition, a review of literature of Multi-Agent applications reveals an interesting popularity and acceptance of JADE.

JADE [12] was completely developed in Java language by the Telecom Italia Lab with the collaboration of University of Parma. JADE provides the basic services necessary to distributed peer-to-peer applications in the fixed and mobile environment. One of these is the support it provides for Agent mobility, which allows its use for the creation of distributed applications where mobility plays an important role e.g. searching. It allows each agent to dynamically discover other agents and to communicate with them according to peer-to-peer paradigm. From the application point of view, each agent is identified uniquely by a name and provides a set of services.

It can register and modify its services and/or search for agents providing given services, it can control its life cycle and, in particular, communicate with all its peers. Each running instance of the JADE runtime environment is called ***container*** as it can contain several agents. The set of active containers is called ***Platform.*** A single special main container must always be active in a platform and all the other containers register with it as soon as they start.

It follows that the first container to start in a platform must be the Main Container while all the other containers must be "normal" (non-main) and "must be told "where to find (host and port) their main container. Jade uses RMI technology for intra-platform communication and CORBA and HTTP technology for inter-platform communication.

A JADE agent is based on a class that extends the Agent Super class (*a UML class diagram representing the Providers (Local Community) agent from the CDF Projects case study reported in the next subsection is shown in Figure 17).*

The agent class usually contains a constructor (required by Java and, by convention, in Jade used to initialize data structures) and the ***setup*** Method which, automatically invoked by the platform once the constructor ends, is often used to begin the agent activity. An agent can be instantiated only by the platform, when this happens, an univocal ID is assigned to the agent and the constructor followed by the setup method are executed.

Another method automatically invoked by the platform is ***shutdown*** which arises when an agent is about to terminate. It contains the code needed to properly conclude the agent's activities and to reallocate the assigned resources. This is normally sufficient to successfully shutdown the agent.

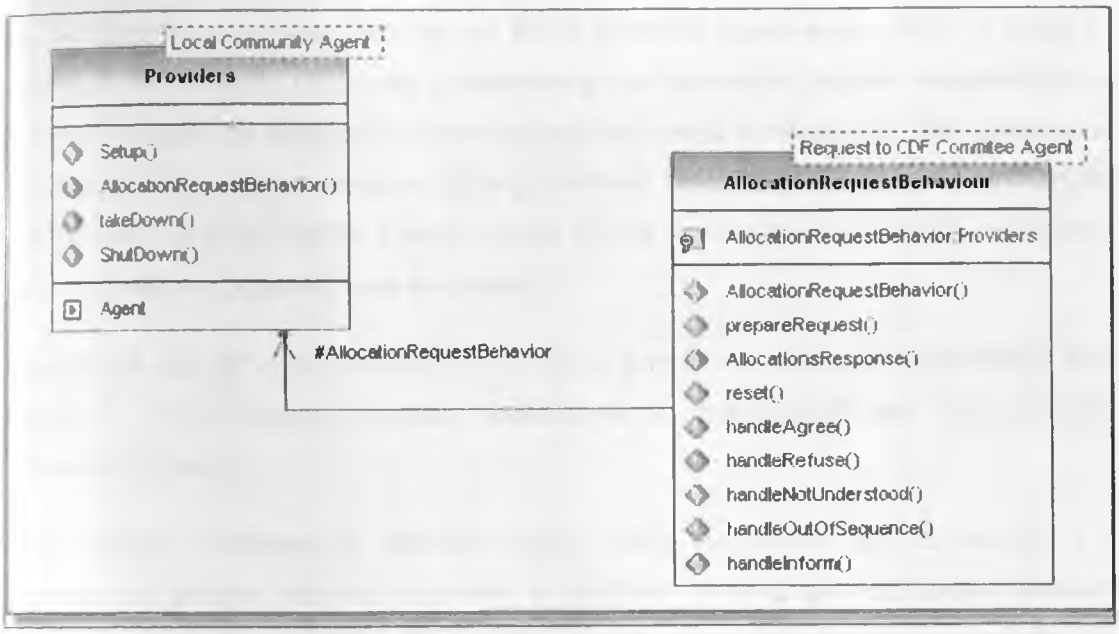


Figure 17: Structure of a Jade agent with a behavior

Agent activities are typically not described in the base class methods, but are located in some sub-classes called ***Behaviors***.

A behavior represents the atomic element of decomposition of the agent's tasks. Operations needed to reach a goal of the agent are partitioned among its behaviors. For instance, communication with another agent is delegated to a specific behavior *(an example in the AllocationRequestBehaviour class shown in the Figure 17, structure of the JADE agent with a behavior).*

Concretely, a behavior is a class that extends a JADE super-class called ***Behavior***. All the Behaviors must contain an ***action*** Method. Like the ***setup*** method, ***action*** is automatically invoked by the platform, after which the class constructor is completed.

A behavior class can contain several methods; a communication behavior is usually made up of a set of methods in order to catch all the incoming messages of a specific Protocol. If a behavior is used to initiate a Request communication, (*as the AllocationRequest behavior in the Figure 17*), it must contain *HandleRefuse*, *HandleInform*, *HandleAgree* and *Handlefailure* methods.

JADE agents communicate by exchanging asynchronous messages, a communication model almost universally accepted for distributed and loosely coupled communications

i.e. heterogeneous entities that do not know anything about each other. In order to communicate, an agent just sends a message to the destination. Agents are identified by a name (no need for destination object reference to send a message). The sender and the receiver could not be available at the same time. The receiver may not even exist (or not yet exist) or could not be directly known by the sender that can specify a property (e.g. "All brokering agents") as a destination.

Despite this type of communication, Security is preserved, since, for applications that require it, JADE, provides proper mechanisms to authenticate and verify "rights" assigned to agents.

When needed therefore, an application can verify the identity of the sender of a message and prevent actions not allowed to platform. All messages exchanged between agents are carried out within an envelop including only information required by the transport layer. That allows, among other things to encrypt the content of a message separately from the envelope.

The structure of a message complies with the ACL Language defined by FIPA [28] and includes fields such as variables indicating the context message refers-to and timeout that can be waited before an answer is received, aimed at supporting complex interactions and multiple parallel conversations. To further support the implementation of Complex conversations, JADE provides a set of skeletons of typical interaction patterns to perform specific tasks such as negotiations, auctions and task delegation.

By using these skeletons, (*implemented as java abstract classes*), programmers can get rid of the burden of dealing with synchronization issues, time-outs, errors conditions and in general, all those aspects that are not strictly related to the application logic.

To facilitate the creation and handling of message content, JADE provides support to automatically converting back and forth between format suitable for content exchange including XML and RDF, and format suitable for content manipulation (*Java objects*). The support is integrated with ontology creation tools e.g Protégé, allowing programmers to graphically create their Ontologies.

JADE is opaque to the underlying inference engine system. If inferences are needed for a specific application, like the CDF projects case study, it allows programmers to re-use their preferred system. It has been already integrated and tested with JESS, PROLOG and DR-DEVICE, the engine chosen for this work.

Another very important feature consists in the availability of a rich suite of graphical tools supporting both the debugging and management/monitoring phases of application life cycle. By means of these tools, it is possible to remotely control agents, even if already deployed and running.

The above described pieces of functionalities make JADE very well suited to support the development and execution of distributed, machine-to-machine, Multi-party, intelligent and proactive applications.

4.5 Brokering System Implementation

This section presents an architecture for a brokering system. Six roles are identified. The Project requesters (*CDF Committee*), Project Providers (Local Community), Broker (Matchmaker), Facilitator (Gateway between RDF repository and Agent Platform), Directory Facilitators (Yellow Pages service) and Updaters – updates the yearly data used and any constituency additions.

An assumption is made that agents do not know the addresses of each other and use the yellow pages service to subscribe, or search for a particular service. Offers or advertisements of projects are expressed in RDF and requirements are expressed in defeasible logic. The domain of the projects is modeled in RDFS ontology. The main interaction protocol is FIPA request and is hard-coded to all the participating agents. The allowed actions an agent can perform and other attributes regarding brokering procedure itself (not projects' special characteristics) are expressed in a message content ontology in protégé.

4.5.1 Brokering System Architecture

This section presents the architecture of the brokering system. At first, a description of the basic modules of the system and consequently the interactions among the agents, which participate in a brokering scenario. The overall architecture of the brokering system is depicted in *Figure 18: Brokering System Architecture.*

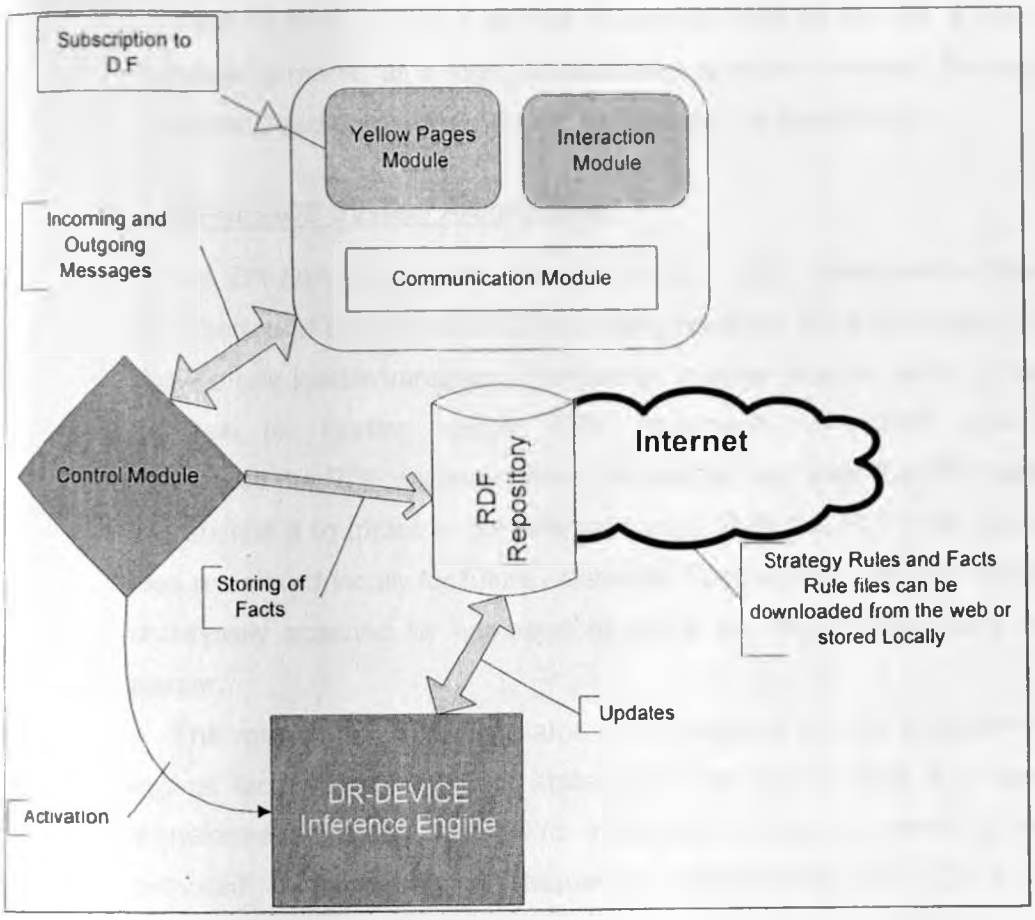


Figure 18: Brokering System Architecture

Description of the Modules

- Reasoning Module
- RDF Module
- Communication Module
- Control Module

Reasoning Module

The role of the reasoning module is to apply the queries to files, which contain the facts and the rules, and to evaluate the results. The results are then exported to an RDF/XML document through the RDF extractor .as local or remote file which is then made available to both the requesters and the providers.

DR-DEVICE is used as the reasoning module for this system. This choice is made, as a logic programming system is needed for supporting well-founded semantics. DR-DEVICE offers this functionality.

DR-DEVICE System Architecture

The *DR-DEVICE* system consists of two major components (*Figure 19: Architecture of the DR-DEVICE reasoning Module*): the RDF loader/translator and the rule loader/translator. The former accepts from the latter (or the user) requests for loading specific RDF documents. The RDF triple loader downloads the RDF document from the Internet and uses the ARP parser [50] to translate it to triples in the N-triple format. Both the RDF/XML and N-triple files are stored locally for future reference. Furthermore, the RDF document is recursively scanned for namespaces which are also parsed using the ARP parser.

The role of the RDF translator is to transform the RDF statements into logical facts, and the RDFS statements into logical facts and rules. This transformation allows the RDF/S information to be processed by the rules provided by the Service Requester (representing the CDF's Projects requirements and preferences).

The rule translator accepts from the rule loader (or directly from the user) a set of rules in *DR-DEVICE* notation and translates them into a set of CLIPS production rules. The translation of the defeasible logic rules is performed in two steps: first, the defeasible logic rules are translated into sets of deductive, derived attribute and aggregate attribute rules of the basic *R-DEVICE* rule language, and then, all these rules are translated into CLIPS production rules. All compiled rule formats are kept into local files, so that the next time they are needed they can be directly loaded, increasing speed. When the translation ends, CLIPS runs the production rules and generates the objects that constitute the result of the initial rule program or query. Finally, the result-objects are exported to the user as an RDF/XML document through the RDF extractor.

RDF Module

The RDF module is responsible for all the actions related with the handling of the advertisements and the domain ontology. The most important functions of this module are:

- Initial storing of RDFS ontology and XML instances into RDF repository
- Update of RDF repository with XML descriptions that are sent by the service providers and correspond to Project advertisements.
- Storage of the RDF Instances after conversion from XML instances
- Preparation of CDF requirements/preferences and forwarding to RDF repository.
- Semantic-Syntactic Validator
- Acquisition of RDF Brokering results

The role of this module is to download the files, which contain the rules in defeasible logic format and the project offers in XML format.

Using the Java and XML Document Object Model (DOM) serializers [52], the XML file is converted to RDF file, the format which is tested with the *DR-DEVICE* engine. Depending with the brokering parameters passed, the serializers also append the input offers RDF file(s) URI in the *rdf_import* attribute of the *rulebase (root)* element of the RuleML document.

The module also has a parser for validating RDF/XML descriptions. During the serialization, the document is first parsed by this module. The tests performed, among others are: class hierarchy loops, property hierarchy loops, domain/range of sub-properties, source/target resources of properties and types of resources.

It also implements methods for file handling (downloads and uploads). For the implementation of this module we used the API of java for File Management and the API for Networking.

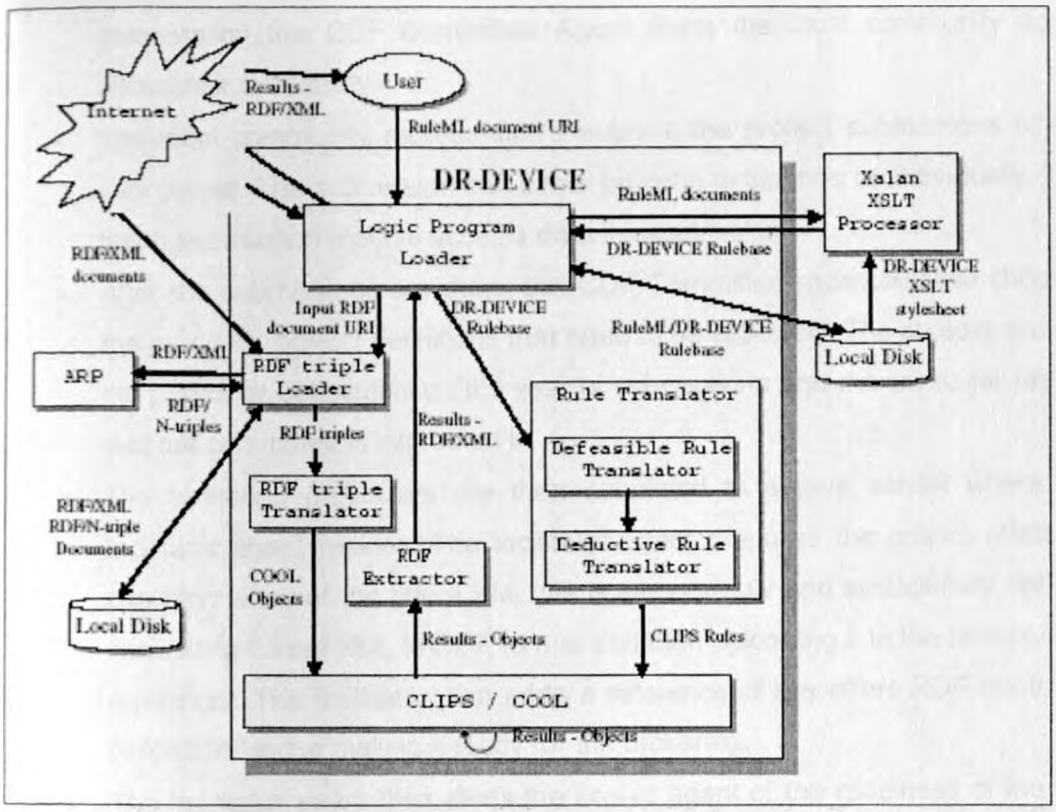


Figure 19: Architecture of the DR-DEVICE reasoning Module

Interaction and Communication Modules

The communication module is responsible for sensing the network and notifying the other modules when an event (e.g. a request message) occurs. In order to decide the course of action based on the incoming message's type, the particular agents extracts the message from the queue and examines its type, i.e. whether it is a "Broker Request", "Download Request" message etc. Accordingly it activates the interaction module. Interaction module consists of different interaction protocols that extend the standard FIPA Request interaction protocol. For the implementation of these modules, we used the API of JADE framework.

Description of the Interactions

- The local community Agent requests for annual project allocations and an interface for project allocations and an interface for project submissions from the CDF Committee Agent.
- The CDF Committee Agent responds by retrieving the allocations from the allocations Database and presents them through the browser. After the

presentation the CDF Committee Agent alerts the local community agent about their availability.

- The local community representative submits the project submissions to the web server. The submission can either be done in batches or individually. The batch submission module accepts data in excel format.
- After the submissions are done, the CDF Committee representative chooses the particular project definitions that need to be brokered. The choices are per the particular constituency , the year of submissions and the particular results that the committee is interested in.
- The brokering particulars are then submitted to a java servlet where the facilitator agent resides. The facilitator agent prepares the project offers by checking whether the offers XML file is semantically and syntactically correct, converting it from XML to RDF format and then uploading it to the remote RDF repository. The facilitator also adds a reference of the offers RDF file to the RuleML rules file making it ready for the brokering.
- The facilitator agent then alerts the broker agent of the readiness of the files for inference and sends the address of the offers files. The facilitator agent also sends a request to the broker asking it to initiate brokering.
- The broker responds by either agreeing or disagreeing, and if it agrees, it hence asks the CDF Committee agent for the rules address.
- On receiving the rules address, the broker invokes the inference engine. After the brokering activity is complete, the broker converts the results from RDF format (*the output of the engine*) to XML, a format which then uses XSLT Stylesheets for readability to the users. The broker then sends the results address to both the CDF and Local Community Agent after which it invokes their display.

4.6 Case Study

4.6.1 Constituency Development Funds Allocation Description

Project Eligibility Criteria

1. No single proposed project should take more than 50% of the total Constituency allocation (which excludes recurrent expenditure) for projects in current budget year.

2. The project should be self-sustaining. Not more that 3% of the total allocation should go to the recurrent expenditure.
3. The Project should be in the following project areas in line with the current Government Development Policies. Poverty Reduction, Education, Health, Human Rights, Sports, Monitoring and evaluation and Environment. Statutory requirements for distribution of funds allocated to the Constituency are as follows:-
    - Maximum of 3% for administration.
    - 5% of the total fund for CDF set aside for emergencies.
    - Maximum of 3% for recurrent expenses.
    - Maximum of 2% for sports activities.
    - Maximum 2% for Monitoring and Evaluation.
    - Maximum 2% for Environmental activities.
    - Maximum of 15% for Education Bursary scheme, Mocks and Continuous assessment tests.

4. The Project should be community based where a community is not a group of organized people but a cross-section of people so that the prospective benefits of the project are available to a wide-spread cross section of the inhabitants of a particular area.
5. Locations that had benefited in the previous budget year should not be considered in the current year to ensure equitable distribution of funds unless the projects are on-going which in that case receives precedence.
6. If there are other similar initiatives whether government or other bodies, then the proposals should be complementary hence promoting harmonization.

Preferences

- The CDF committee rates the importance of implementing projects in the Poverty reduction area more than both the target group sizes and cost. However, the committee prefers a cheaper project than a larger group size one.
- Ongoing projects take precedence over commencing Projects.

Formalization of Projects' Requirements

The following predicates to describe properties of Projects:

Table 10: Rule File Predicates

acceptable(X,Y)	Project X satisfies all the project requirements
Allocation(X)	The CDF allocation for the current year
offer(X,Y)	Committee is willing to offer Y funds to project X
imp-cost (X,Y)	Y is the implementation cost of Project X
op-cost (X,Y)	Y is the operational/sustenance cost of Project X per year
project-area(X,Y)	Y is the Project area of Project X
target-group(X,Y)	Y is the Beneficiary group size of Project X
previous-imp(X)	Project X is proposed in a location where another Project was implemented in the Previous Budget Year
project_status(X,Y)	Y is the status of Project X proposal, either a continuation of an ongoing project or a completely new proposal
location(X,Y)	The location where the Project is being Implemented
Geo-Coverage(X,Y)	The geographical coverage in square Kilometers of Project Y.
presence_of_similar_initiatives(X)	Presence of Similar Project initiatives by Government and other Bodies in the Location/Ward.
complementary_proposal(X)	The Project X is a complementary effort of other project initiatives in the Locality

The following predicates are also included.

- Acceptable(X) - Project X satisfies CDF Committee's requirements.

- Offer(X, Y) - CDF Committee is willing to pay KES. Y for Project X
- The CDF committee is willing to pay 25% of the allocation to Poverty Reduction Projects, and an extra Kshs.250,000 for every 500 square Kilometers for a larger Project in terms of Geographical coverage over 5000 square Kilometers and Kshs.100,000 for every 500 people over 7000 on target group sizes.
- The CDF committee is willing to pay 20% of the Constituency allocation to a project benefiting a target group of 7000 and above.

Committee's firm requirements

Any Project is a priori acceptable.

r1: =>acceptable(X);

However, Y is unacceptable if one of Project's requirements is not met.

r2: imp-cost (X, Y), Allocation (Z), X>Z/2 => ¬acceptable(X);

r3: previous-imp(X), project_status (X,"different") => ¬acceptable(X);

r4: op-cost (X, Y), imp-cost (X, Z), Z>Y*0.03 => ¬acceptable(X);

r5: target-group(X, Y), Y<5000=> ¬acceptable(X);

r6: Geo-Coverage(X, Y), Y<2000=> ¬acceptable(X);

r7: presence_of_similar_initiatives(X),¬complementary_proposal(X) =>¬acceptable(X);

r8: project-area (X,"education"), imp-cost (X, Y), Y > 0.15(Allocation(X)) = ¬acceptable(X);

r9: project-area (X,"sports"), imp-cost (X, Y), Y > 0.02(Allocation(X)) = ¬acceptable(X);

r10: project-area (X,"Monitoring_Evaluation"), imp-cost (X, Y), Y > 0.02(Allocation(X)) = ¬acceptable(X);

r11: project-area (X,"environmental"), imp-cost (X, Y), Y > 0.02(Allocation(X)) = ¬acceptable(X);

r12 =>offer(X, Y)

Rules r2-r12 are exceptions to rule r1, so we add

r2 > r1, r3 > r1, r4 > r1, r5 > r1, r6 > r1, r7 > r1, r8 > r1; r9 > r1; r10 > r1; r11> r1; r12> r1

CDF flexibilities

r13: project-area (X,"Poverty_Reduction"), Geo-Coverage(X, Z), Z>2000=

Offer (X, (0.25*allocation(X) + ((Z-2000)/500)*250,000))

r14: offer(X, Y), imp-cost (X, Z), Y < Z) =>¬acceptable(X)

r14 > r1;

r15: target-group (X, Y), Y>7000 => Offer (X, (0.02 * allocation(X) + ((Y-7000)/500)*100,000))

Preferences

Committee's Implementation preferences are based on Poverty Reduction, Ongoing Projects, price and size, in that order. We represent them as follows:

r16: project-area (X,"Poverty_Reduction") => implement(X);

r17: project-area (X,"Poverty_Reduction"), cheapest(X) => implement(X);

r18: project-area (X,"Poverty_Reduction"), cheapest(X), largesttarget(X) => implement(X);

r19: project-area (X,"Poverty_Reduction"), cheapest(X), largesttarget (X), project_status (X,"ongoing") => implement(X);

r19 > r16;

r19 > r17;

r19 > r18;

r18 > r17;

r18 > r16;

r17 > r16;

The complete rule file is found in the appendix in DR-DEVICE *RULEML* notation.

5 Testing Verification and Results

5.1 Introduction

Testing is an activity in which a system or component is executed under specified conditions, the results are observed or recorded and compared against specifications or expected results, and an evaluation is made of some aspect of the system or component.

A test is a set of one or more test cases. The main aim of a test is to find faults.

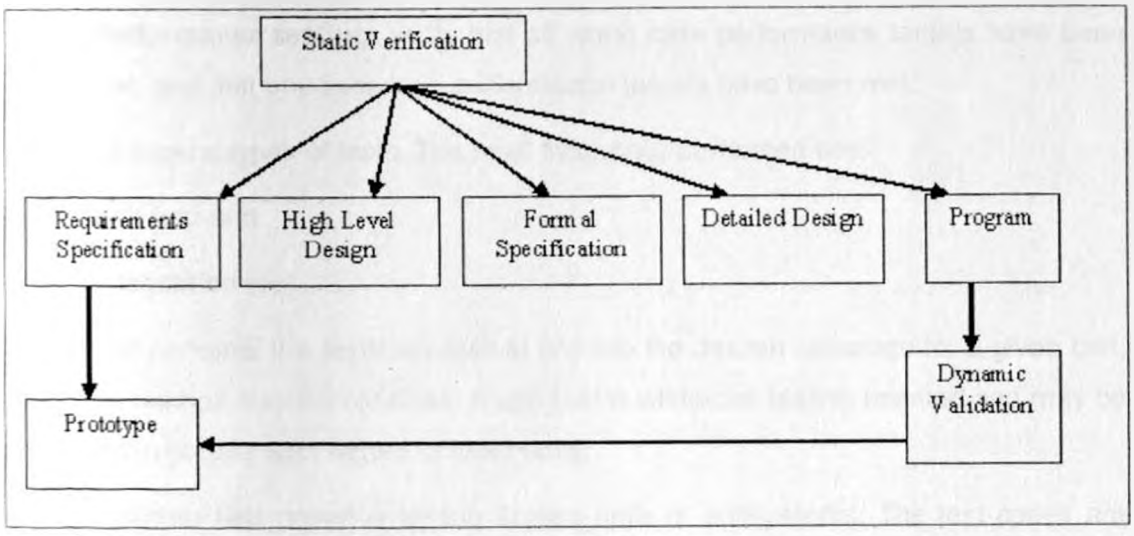


Figure 20: static verification and dynamic validation tests

There are two kinds of tests: static verification and dynamic validation. Static verification is based on code inspection or "walk through", symbolic execution, and symbolic verification. Dynamic validation generates test data and executes the program.

Figure 20 show where static verification and dynamic validation tests occur during the software life cycle [64].

There are several strategies for testing software. The main strategies found in literature ([64],[60]) for testing software which are related to this work are:-

- **Black-box testing:** also know as functional testing or specification-based testing, Testing without reference to the internal structure of the component or system.
- **White-box testing:** testing based on an analysis of the internal structure of the component or system. Test cases[12] are derived from the code e.g. testing paths.
- **Progressive testing:** it is based on testing new code to determine whether it contains faults.
- **Regressive testing:** process of testing a program to determine whether a change has introduced faults (regressions) in the unchanged code. It is based on re-execution of some/all of the tests developed for a specific testing activity.
- **Performance testing:** verify that all worst case performance targets have been met, and that any best-case performance targets have been met.

There are several types of tests. The most frequently performed are:-

1. **unit test** and
2. **Integration test**.

A **unit test** performs the tests required to provide the desired coverage for a given unit, typically a method, function or class. A unit test is white-box testing oriented and may be performed in parallel with regard to other units.

An **integration test** provides testing across units or subsystems. The test cases are used to provide the desired coverage for the system as a whole. It tests sub-system connectivity.

There are several strategies for implementing integration test:

- **bottom-up**, which tests each unit and component at lowest level of system hierarchy, then components that call these and so on;
- **top-down**, which tests top component and then all components called by this and so on;
- **big-bang**, which integrates all components together; and
- **Sandwich**, which combines bottom-up with top-down approach.

---

[12] *A test case is a set of test inputs, execution conditions, and expected results developed for a particular objective.*

The above techniques and strategies will appear in the testing approach described in the following sections. They relate with the testing works presented here and classify them according to each strategy or technique.

5.2 Performance Evaluation

In this Section, a report is made on the experimental evaluation that was conducted in order to measure the performance of system. The systematic interactions of the system modules are then presented.

5.2.1 Multi-Level Testing Approach

The importance of verifying the accuracy and reliability of software has been universally recognized. Concerning multi-agent systems, very few research works have been undertaken in order to provide developers with valuable tools supporting testing activities. Moreover these works are still at very early stage.

Actually formal methodologies give validation tests that are however applicable in very few and quite irrelevant cases. The main reason of this lack is that the activities, which should assure that the program performs satisfactorily, are very challenging and expensive since it is quite complicated to automate them [18].

There are at least three approaches for evaluating multi-agent systems:

(i) theorem proving, which corresponds to checking that a set of formulas satisfies a goal;

(ii) model checking, which builds a model of a system and verifies that a temporal logic formula holds for the model; and finally,

(iii) Testing, which consists in applying test cases to the system and checks if it behaves properly different from model checking which checks if agents are proved correct [44].

The several challenges in developing and debugging multi-agent systems make traditional component testing technologies not applicable [61]. Besides all the challenges due to its autonomy and parallelism, message specifications are often incorrect or incomplete and visualization of the messages is needed. Jade provides a sniffer agent who allows for visualization of messages through the sniffer window, as exchanged between agents. This tool after evaluation was considered fitting for this work as shall be shown later in this section.

Another challenge is the fact that agents can be programmed to learn, so successive tests with the same test data may give different results [61]. This work does not need any agents to learn, so successive test yield the same results.

5.2.2 Design and acquisition of the test data

The test data used in this system is obtained from the CDF website [21]. Some of the data that constitute a sample test case, a complete project definition, is not available from the site and hence have been computer-generated. The generations are guided by the requirements of the CDF Act 2003. The selected collection set of test cases were chosen arbitrarily from different years and different constituencies.

Below is a set of the sample test data that was used for testing, represented in its raw form and the generations that were made to make it complete.

Table 11: Langata constituency Projects Sample Data for the year 2007/2008

Name	Project Cost	Operating Cost	Target size	Project Category	Geographical Coverage	Location	Previous Year Beneficiaries	Project Status	Presence Of similar initiatives	Complementary Proposal
Office administration				administrative		All allocations	yes	on-going		
Emergency funds	-			administrative		All allocations	yes	on-going		
Design, monitoring &evaluation	-			monitoring_evaluation		All allocations	yes	on-going		
Const.of multipurpose office for constituency committees. Prime time	6,154,665			administrative		Kibera-Dos office Compound	yes	different		
Upgrading of DO health centre Kibera.	8,487,351			administrative		Kibera Dos Office compound?	yes	On-going		
Const. of 10door modern toilets anyix enterprises	943,680			human_rights		Gatwekera Village	yes	On-going		

Const. of laboratory ? Langata High. Roji Const.	3,773,688			education		Langata	yes	On going		
Construction of 10 door modern toilets plus 10000 lt water tank Kianda Village (Express Strategies)	683,247			human_rights		sarangombe	yes	on-going		
Drainage system (Kenberg)	1,100,000			health		Laini Saba	yes	on-going		
0	-					0	no	0		
Provision and installation of two 10,000lt water tank. (Jowo Ent)	789,991			poverty_reduction		Laini Saba Highrise village	yes	on-going		
Const. of 15 door? pit latrine (franton k. Ltd)	749,300			human_rights		Laini? Saba	yes	on-going.		
0	-					Highrise Village	no	Only foundation slab started		

Erection of security light mast. Adopt a light	2,500,000			others		Sarangombe (Makina village)	yes	Re-Allocated to Karen plains road-2007/8		
Repair of drainage system (Higir Tranp)	1,415,619			health		Makina	yes	on-going		
Construction of 4 door pit latrine & bathroom at darajani	225,500			human_rights		Kibera	yes	different		
Supply of Braille Books	200,000			education		Langata	no	different		
Construction of perimeter wall	800,000			education		Nairobi west	no	different		
Supply of 250 desks	280,000			education		Nairobi west	no	different		
Supply of 250 desks	320,000			education		makina	no	different		
Supply of 200 desks	280,000			education		sarangombe	no	different		
Rehabilitation of building	350,000			others		mungum	no	differen		

						oini		1		
Supply of 200 desks	320,000			education		Laini saba	no	different		
Repair of parking bay	480,000			others		Nairobi west	no	different		
Construction of 6door pit latrine	480,000			human_rights		Southland village	no	different		
Construction of 4door pit latrine	200,000			human_rights		Sarangombe chiefs camp	no	different		
Provision of water	500,000			poverty_reduction		Nairobi west	no	different		
Construction of jua kali sheds	700,000			poverty_reduction		mugumoini	no	different		
Construction of jua kali sheds	842,000			poverty_reduction		Nairobi west	no	different		
Connection of water pipes to install tanks	1,400,000			poverty_reduction		All locations with tanks	no	different		

From the above table, the system requires the data with eleven attributes namely:-

1. Project Name
2. Project Cost
3. Operating Cost
4. Target size
5. Project Category
6. Geographical Coverage
7. Location
8. Whether a Location is a Previous Year Beneficiary
9. Project Status (where a different project or on-going from the previous year)
10. Presence of similar initiatives by other development partners
11. If presence of development partners whether the project is Complementary Proposal

From the tabulated data above [*Table 11: Langata constituency Projects Sample Data for the year 2007/2008*], Operation Cost, target group size, geographical coverage, presence of similar initiatives and whether a project is a complementary proposal to other project initiatives is completely lacking.

This means that the missingness levels are more than 10%. These variables, with exception of geographical coverage attribute, may not be ignored since they contribute to a big extent, the outcome of the results and hence a consequential bias/ impact if ignored. This missing may be classified as **Non-ignorable Missingness**.

Since most of the data missing attributes are dependable attributes, then the way to handle the missing data is by imputation. Imputation refers to the replacement of missing data with a substitute that allows data analysis to be conducted without being misleading.

The following will be adopted as the driving principles/ imputation policies of this work:-

- Imputation should not lead to biases or distributional changes in the data, or significant extra variance to estimators.
- The imputation process should rely on data from the other variables rather than making external assumptions about the likely nature of missing data.

- Imputation should not lead to important sample estimates being based too heavily upon imputed values.
- The imputation should maintain, as far as possible, the underlying variability in the data.
- Target group Imputation will be based on data otherwise external to the research design. It will be estimated based on known population of the location data associated with each project submitted.

Table 12 : Other Missing values imputation Formulas

ibute	**Formula**	**Justifications**
rating Cost	=**RAND()*5***0.2*implementation cost	The CDF ACT stipulates that in any given allocation year, the recurrent expenditure should not be more than 30%. This requirement has been therefore cascaded to every project by ensuring that all the acceptable projects require operating cost less than 30% of the implementation cost every year. The imputation formula used in this work therefore takes a random variable between 0 and 5 and multiplies it with 20% of the implementation cost of the specific projects. This ensures that the some projects operating cost is below 30% of implementation cost and others are above so that the some are disqualified for implementation.
sence of ilar atives	=IF(**RAND()<=0.5**,"no","yes")	This attribute value is either "**yes**" or "**no**". To maintain some good distribution, a random variable has been used. If the variable value is less that 0.5, then a value of "**no**" is imputed and a "**yes**" if random variable is greater than 0.5
mplementary Dposal	=IF(**RAND()<=0.5**,"no","yes")	This attribute is used together with the "presence of similar initiatives" attribute to determine whether a project is acceptable or not. If there are similar project initiatives similar to the

		proposal being made, then these proposals should complement these initiatives in order to harmonize development. This variable therefore takes a Boolean variable of "**yes**" or "**no**". The imputation formula used for "Presence of similar initiatives" attribute is the same one used here.

Using the sample data above for the Langata Constituency, after the data imputation of all the missing attributes (*with exception of the geographical coverage which is ignored*) the complete set of data will may be represented by the table below:-

Table 13: Langata constituency Projects Sample Data for the year 2007/2008 after imputation

Name	Project Cost	Operating Cost	Target size	Project Category	Geographical Coverage	Location	Previous Year Beneficiaries	Project Status	Presence of similar initiatives	Complementary Proposal
Const.of multipurpose office for constituency committees. Prime time	6,154,665	2,354,890.88	117106	administrative		Kibera-DOs office Compound	yes	different	no	yes
Upgrading of DO health centre Kibera.	8,487,351	1,638,931.78	117106	administrative		Kibera DOs Office compound	yes	on-going	no	no
Const. of laboratory - Langata High. Roji Const.	3,773,688	1,648,498.52	22555	education		Langata	yes	on-going	yes	no
Construction of 10 door modern toilets plus 10000 lt water tank Kianda Village - Express Strategies	683,247	554,919.03	66548	human_rights		sarangombe	yes	on-going	yes	yes
Fencing of School Uhuru Garden Pri.	2,123,346	1,506,455.33	66548	education		Sarangombe	yes	on-going	no	no

							Beneficiaries		initiatives	
Sch.250m										
Rehab. Of school library, Langata Rd. pri.Everest Gen.	530,996	398,289.41	22555	education		Langata	yes	on-going	yes	no
Repair of Nzohe Rd. 3km Berlin	4,079,580	1,089,650.87	13663	others		Karen	yes	on-going	yes	yes
Cost of Jua kali sheds Construction of one classroom	613,745	512,543.35	13663	poverty_reduction		Karen (Tec. Inst.For the Deaf)	yes	on-going	no	yes
Drainage system - Kenberg	1,100,000	473,210.90	72792	health		Laini Saba	yes	on-going	yes	yes
Provision and installation of two 10,000lt water tank-Jowo Ent	789,991	41,371.35	72792	poverty_reduction		Laini Saba Highrise village	yes	on-going	no	yes
Const. of 15 door pit latrine -franton k. Ltd	749,300	337,032.75	72792	human_rights		Laini Saba	yes	on-going	yes	yes
Construction of	225,500	213,249.02	117106	human_rights		Kibera	yes	different	yes	no

Name	Project Cost	Operating Cost	[illegible]	[illegible]	Coverage	[illegible]	Year Beneficiaries	Status	[illegible] initiatives	[illegible]
4 door pit latrine & bathroom at darajani										
Supply of Braille Books	200,000	26,596.57	22555	education		Langata	no	different	no	yes
Construction of perimeter wall	800,000	130,528.59	59517	education		Nairobi west	no	different	yes	no
Supply of 250 desks	280,000	42,657.75	59517	education		Nairobi west	no	different	no	no
Supply of 200 desks	280,000	2,051.83	66548	education		sarangombe	no	different	yes	no
Rehabilitation of building	350,000	272,342.39	13299	others		mungumoini	no	different	yes	yes
Supply of 200 desks	320,000	76,542.79	72792	education		Laini saba	no	different	yes	yes
Repair of parking bay	480,000	53,295.33	59517	others		Nairobi west	no	different	no	yes
Construction of 4door pit latrine	200,000	133,778.72	66548	human_rights		Sarangombe chiefs camp	no	different	no	no
Provision of water	500,000	40,714.25	59517	poverty_reduction		Nairobi west	no	different	no	yes

Name	Project Cost	Operation Cost	Target size	Project category	Coverage		Year Beneficiaries	Status	of similar initiatives	
Construction of jua kali sheds	700,000	221,056.05	13299	poverty_reduction		mugumoini	no	different	no	no
Construction of jua kali sheds	842,000	331,304.39	59517	poverty_reduction		Nairobi west	no	different	no	no
Connection of water pipes to install tanks	1,400,000	722,226.34	368274	poverty_reduction		All locations with tanks	no	different	yes	no

To Note:
A few test cases are missing in vital data attributes like implementation cost and their location attributes are not using the recognized location names and consequently their target size may not be estimated. These cases will simply be omitted.

Projects Identification Results

Project Name	Truth Status	Recommendations	Offers	Allocation Percentage	Cumulative Percentage
Upgrading of DO health centre Kibera	defeasibly-proven	passed_criteria		21%	21%
Repair of Nzohe Rd 3km Berlin	defeasibly-proven	passed_criteria		10%	31%
Provision and installation of two 10,000lt water tank- Jowo Ent	defeasibly-proven	passed_criteria		2%	33%
Supply of Braille Books	defeasibly-proven	passed_criteria		0%	33%
Supply of 250 desks	defeasibly-proven	passed_criteria		1%	34%
Repair of parking bay	defeasibly-proven	passed_criteria		1%	35%
Provision of water	defeasibly-proven	passed_criteria		1%	36%

Results Discussions

The table above [*Table 14: Results after the first run*] is a representation of the results obtained in the category of Projects passes the basic criteria for implementation.

Truth status column is an explanation that the project has been proved defeasibly (*meaning that no rule with contrary evidence has defeated it*).

The recommendation column is an indication of the results produced in an inference instance. Other recommendations could be those projects passed for implementation, the cheapest or projects with the largest target as shall be demonstrated by the other results presented in later in this work. Each project has an implementation cost, and the allocation percentage column depicts the percentage cost of implementing a particular project rounded to the nearest whole number in relation to the yearly allocation of a given constituency. Finally the cumulative percentage is a calculation of the total percentage amount to be used if all the projects that have passed criteria were to be implemented.

To maintain lack of bias in imputation, one of the policies used in this work, the different imputation values obtained after formula runs will be used to test the system.

The results recorded are for 3 counts of imputation formula run.

On a second run of imputation (*Langata constituency Projects Sample Data*), the following sample data is recorded.

Name	Project Cost	Operating Cost	Target size	Project Category	Geographical Coverage	Location	Previous Year Beneficiaries	Project Status	Presence of similar initiatives	Complementary Proposal
Const.of multipurpose office for constituency committees. Prime time	6,154,665	4,457,586.55	117106	administrative		Kibera-DOs office Compound	yes	different	yes	no
Upgrading of DO health centre Kibera.	8,487,351	629,642.74	117106	administrative		Kibera DOs Office compound	yes	on-going	yes	no
Const. of laboratory - Langata High. Roji Const.	3,773,688	2,714,209.85	22555	education		Langata	yes	on-going	yes	yes
Construction of 10 door modern toilets plus 10000 lt water tank Kianda Village - Express Strategies	683,247	389,222.49	66548	human_rights		sarangombe	yes	on-going	yes	no
Fencing of School Uhuru Garden Pri. Sch.250m	2,123,346	931,753.58	66548	education		Sarangombe	yes	on-going	yes	no
Rehab. Of school library, Langata Rd. pri.Everest Gen.	530,996	381,676.01	22555	education		Langata	yes	on-going	no	yes
Repair of Nzohe Rd. 3km Berlin	4,079,580	44,461.02	13663	others		Karen	yes	on-going	yes	no

Cost of Jua kali sheds Construction of one classroom	613,745	372,746.20	13663	poverty_reduction		Karen (Tec. Inst.For the Deaf)	yes	on-going	no	yes
Drainage system - Kenberg	1,100,000	219,056.19	72792	health		Laini Saba	yes	on-going	no	no
Provision and installation of two 10,000lt water tank-Jowo Ent	789,991	726,966.88	72792	poverty_reduction		Laini Saba Highrise village	yes	on-going	yes	no
Const. of 15 door pit latrine -franton k. Ltd	749,300	47,437.04	72792	human_rights		Laini Saba	yes	on-going	no	yes
Construction of 4 door pit latrine & bathroom at darajani	225,500	74,632.30	117106	human_rights		Kibera	yes	different	yes	yes
Supply of Braille Books	200,000	113,250.56	22555	education		Langata	no	different	no	yes
Construction of perimeter wall	800,000	727,351.92	59517	education		Nairobi west	no	different	no	yes
Supply of 250 desks - Nairobi west	280,000	224,739.71	59517	education		Nairobi west	no	different	yes	yes
Supply of 200 desks - sarangombe	280,000	256,061.32	66548	education		sarangombe	no	different	no	yes
Rehabilitation of building	350,000	234,609.20	13299	others		mungumoini	no	different	no	yes

Supply of 200 desks	320,000	197,208.15	72792	education		Laini saba	no	different	yes	yes
Repair of parking bay	480,000	114,731.65	59517	others		Nairobi west	no	different	yes	no
Construction of 4door pit latrine	200,000	109,457.82	66548	human_rights		Sarangombe chiefs camp	no	different	no	no
Provision of water	500,000	227,139.80	59517	poverty_reduction		Nairobi west	no	different	no	yes
Construction of jua kali sheds-mugumoini	700,000	472,735.77	13299	poverty_reduction		mugumoini	no	different	yes	no
Construction of jua kali sheds - Nairobi west	842,000	219,618.70	59517	poverty_reduction		Nairobi west	no	different	no	yes
Connection of water pipes to install tanks	1,400,000	1.026,067.89	368274	poverty_reduction		All locations with tanks	no	different	yes	yes

From the tables above, comparing the data for the 2 runs, we see that there are substantial differences among the operating costs of the projects. This will give us specific comparisons of the results with different operating costs, presence of the similar project initiatives and whether the projects proposed are complementary to the initiatives on inference.

Projects Identification Results

Project Name	Truth Status	Recommendations	Offers	Allocation Percentage	Cumulative Percentage
Drainage system - Kenberg	defeasibly-proven	passed_criteria		3%	3%
Const of 15 door pit latrine -franton k. Ltd	defeasibly-proven	passed_criteria		2%	5%
Construction of jua kali sheds - Nairobi west	defeasibly-proven	passed_criteria		2%	7%

The projects recommended for implementation after the second run are different from what was recommended in the first case. To highlight a few cases, below is an instance on *Drainage system – Kenberg*, the data is the same apart from the 3 attributes being imputed as explained earlier(*Operating Cost*, *Presence of similar initiatives*, *Complementary Proposal*). This project is returned in the 2nd run as having passed the criteria for implementation and not in the 1st run. The explanation to the results is that it is disqualified in the first and third case since its operating cost is greater that 30% of the implementation cost, one of the Defeaters rules against acceptance for implementation.

Table 17: Data samples in the 3 runs

Imputation Runs	Name	Project Cost	Operating Cost	Target size	Project Category	Geographical Coverage	Location	Previous Year Beneficiaries	Project Status	Presence of similar initiatives	Complementary Proposal
1st	Drainage system - Kenberg	1,100,000	473,210.90	72792	health		Laini Saba	yes	on-going	yes	yes
2nd	Drainage system - Kenberg	1,100,000	219,056.19	72792	health		Laini Saba	yes	on-going	no	no
3rd	Drainage system - Kenberg	1,100,000	840,797.47	72792	health		Laini Saba	yes	on-going	no	yes

Table 18: Langata constituency Projects Sample Data after 3rd run imputation

Name	Project Cost	Operating Cost	Target size	Project Category	Geographical Coverage	Location	Previous Year Beneficiaries	Project Status	Presence of similar initiatives	Complementary Proposal
Const.of multipurpose office for constituency committees. Prime time	6,154,665	1,284,849.08	117106	administrative		Kibera-DOs office Compound	yes	different	yes	yes
Upgrading of DO health centre Kibera.	8,487,351	2,397,647.68	117106	administrative		Kibera DOs Office compound	yes	on-going	yes	no
Const. of laboratory - Langata High. Roji Const.	3,773,688	3,430,029.35	22555	education		Langata	yes	on-going	yes	no
Construction of 10 door modern toilets plus 10000 lt water tank Kianda Village -Express Strategies	683,247	339,325.37	66548	human_rights		sarangombe	yes	on-going	yes	no
Fencing of School Uhuru Garden Pri. Sch.250m	2,123,346	1,545,276.82	66548	education		Sarangombe	yes	on-going	no	no
Rehab. Of school library, Langata Rd. pri.Everest Gen.	530,996	395,064.62	22555	education		Langata	yes	on-going	no	no

	Cost	Cost	size		Coverage		Beneficiaries		Initiatives	
Repair of Nzohe Rd. 3km Berlin	4,079,580	2,044,581.60	13663	others		Karen	yes	on-going	yes	no
Cost of Jua kali sheds Construction of one classroom	613,745	447,193.78	13663	poverty_reduction		Karen (Tec. Inst.For the Deaf)	yes	on-going	yes	yes
Drainage system - Kenberg	1,100,000	840,797.47	72792	health		Laini Saba	yes	on-going	no	yes
Provision and installation of two 10,000lt water tank-Jowo Ent	789,991	698,030.57	72792	poverty_reduction		Laini Saba Highrise village	yes	on-going	no	yes
Const. of 15 door pit latrine -franton k. Ltd	749,300	18,292.63	72792	human_rights		Laini Saba	yes	on-going	no	yes
Construction of 4 door pit latrine & bathroom at darajani	225,500	214,423.14	117106	human_rights		Kibera	yes	different	yes	yes
Supply of Braille Books	200,000	5,945.57	22555	education		Langata	no	different	yes	no
Construction of perimeter wall	800,000	51,028.48	59517	education		Nairobi west	no	different	yes	yes
Supply of 250 desks	280,000	5,053.42	59517	education		Nairobi west	no	different	yes	no
Supply of 200 desks	280,000	260,497.53	66548	education		sarangombe	no	different	yes	no

Rehabilitation of building	350,000	277,912.43	13299	others		mungumoini	no	different	no	yes
Supply of 200 desks	320,000	83,631.01	72792	education		Laini saba	no	different	no	yes
Repair of parking bay	480,000	461,561.91	59517	others		Nairobi west	no	different	no	no
Construction of 4door pit latrine	200,000	46,464.70	66548	human_rights		Sarangombe chiefs camp	no	different	yes	yes
Provision of water	500,000	134,625.39	59517	poverty_reduction		Nairobi west	no	different	yes	no
Construction of jua kali sheds	700,000	513,032.94	13299	poverty_reduction		mugumoini	no	different	no	no
Construction of jua kali sheds	842,000	23,042.68	59517	poverty_reduction		Nairobi west	no	different	no	no
Connection of water pipes to install tanks	1,400,000	422,400.71	368274	poverty_reduction		All locations with tanks	no	different	yes	yes

Table 19: Results after the 3rd run

Projects Identification Results

Project Name	Truth Status	Recommendations	Offers	Allocation Percentage	Cumulative Percentage
Const of 15 door pit latrine -franton k. Ltd	defeasibly-proven	passed_criteria		2%	2%
Construction of perimeter wall	defeasibly-proven	passed_criteria		2%	4%
Construction of 4door pit latrine	defeasibly-proven	passed_criteria		0%	4%

The conclusion can therefore be drawn that if this system is to be implemented in a real environment then the missing information from CDF website must be collected from the ground, from the local community representatives. This is evidence by the divergence of results caused a small variance of data in the 3 attributes imputed.

Agent Level Testing

This work adopts testing as described in PASSI [18], where testing is divided into two different steps.

- **Agent test** - verifying the behavior of each agent with regards to the system requirements that are under the responsibility of that agent
- **Society test** – Carrying out validation of the overall results of the different agents and verification of successful integration of the different agents.

This was incrementally done from agent to agent as the development progressed. The criterion used for this kind of testing is as outlined below:-

- Ensuring correctness in message addressing between agents
- Correct agent Action/Task execution by the behaviors as specified in the design chapter
- Correct Message parsing
- Checking the right performatives – the type of communicative act in the incoming messages.
- Agents' Life cycle – Granting that agents are active for the entire system life period (No agent termination prematurely)
- To test the correctness of the messages addressed between agents, status of agents at any time and the incoming messages communicative acts, a sniffer agent is started. The sniffer agent then asks the Agent Management System

(AMS), in-built in the Jade Platform, to sniff all the communications and show clearly how different agent entities communicate including the contents of the messages. An illustration of message exchanged and its content is shown in the *Figure 21: Sniffed Message –Message sent by the Local Community Agent to the CDF Committee Agent.* Remote Monitoring Agent (RMA) on the other hand keeps track of all registered agents and hence monitors the life cycle of all the agents.

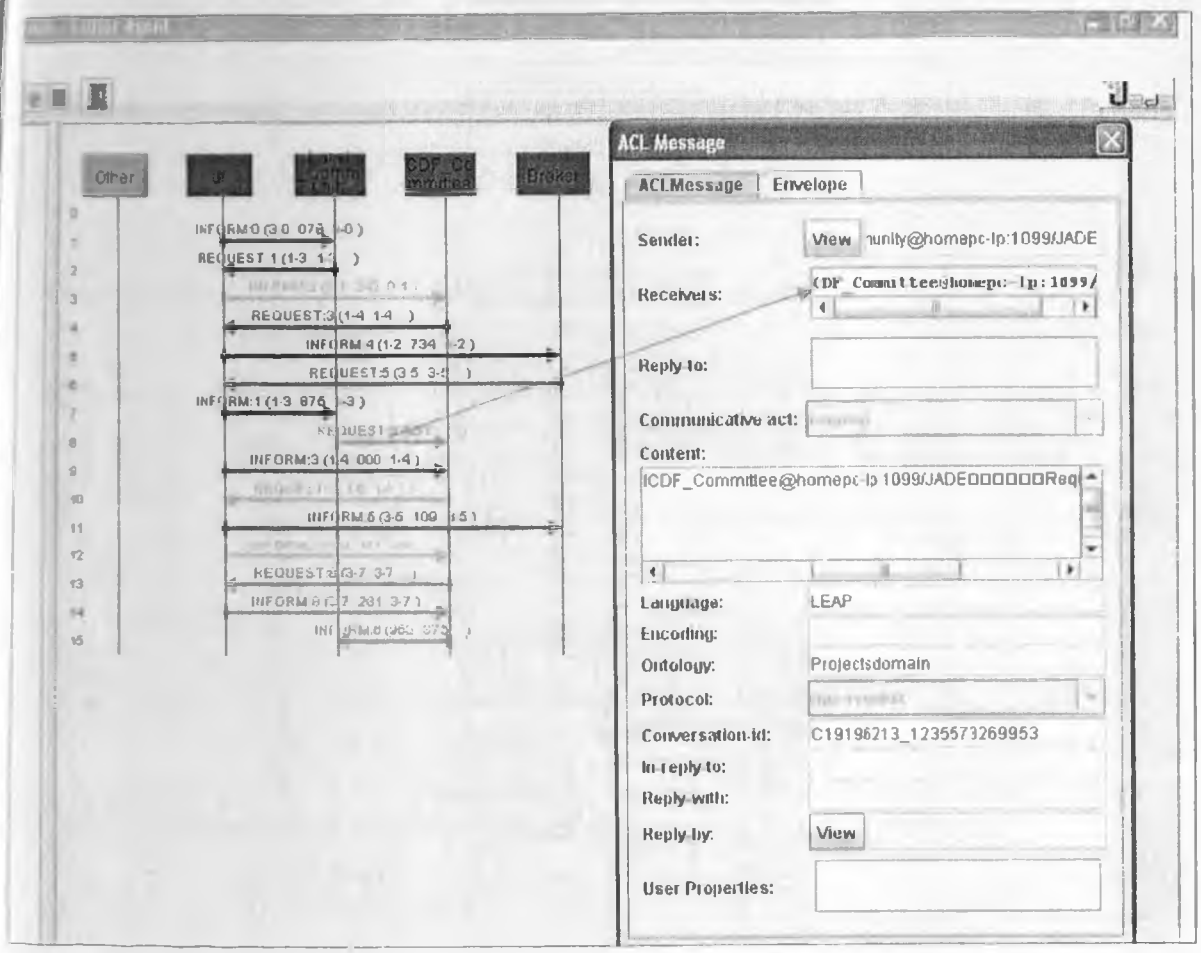


Figure 21: Sniffed Message –Message sent by the Local Community Agent to the CDF Committee Agent

The results obtained that underpin testing as guided by testing criteria above are as illustrated in the following samples: -

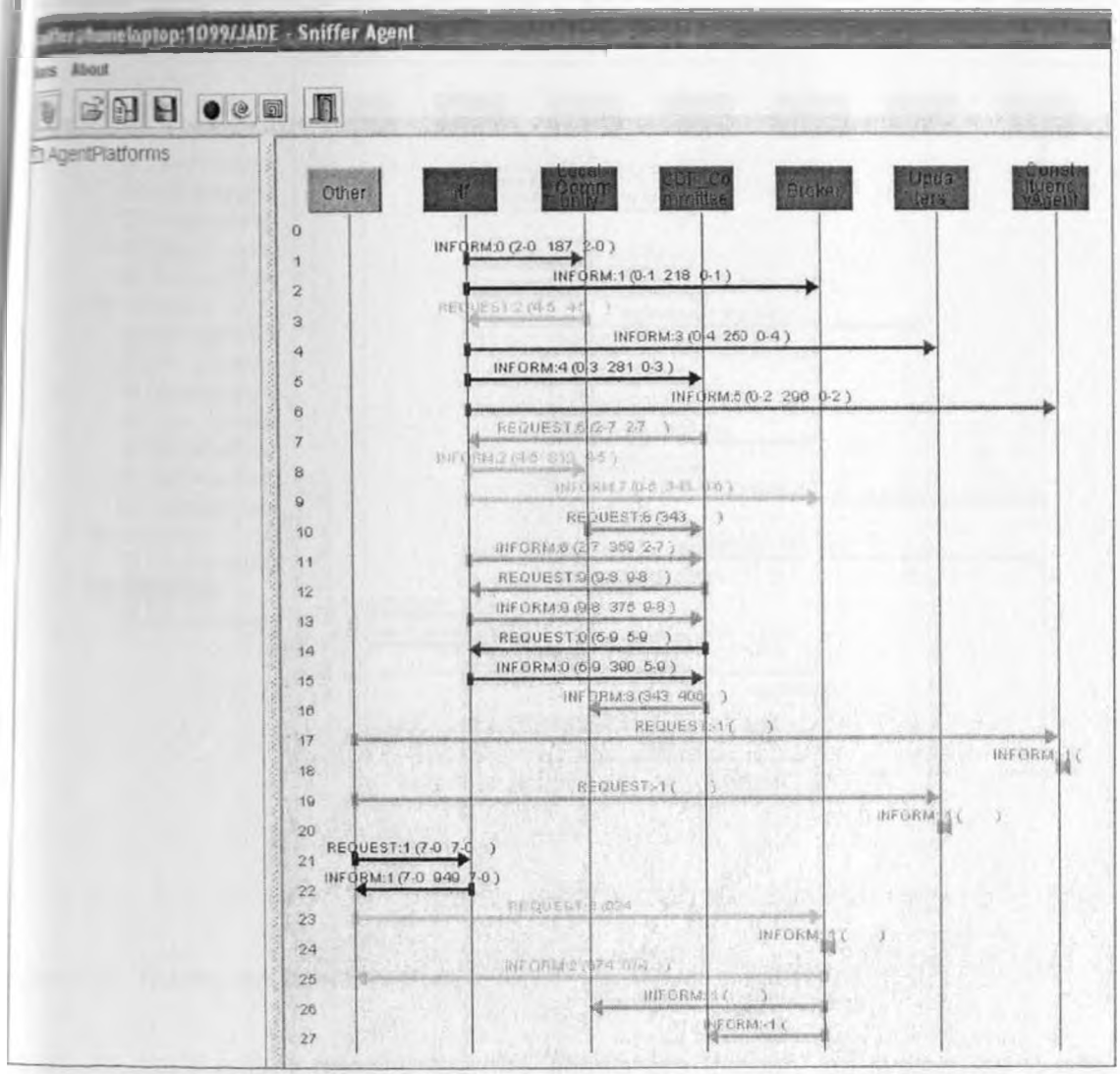


Figure 22: Request/Inform 8: right messages are passed to the right recipients

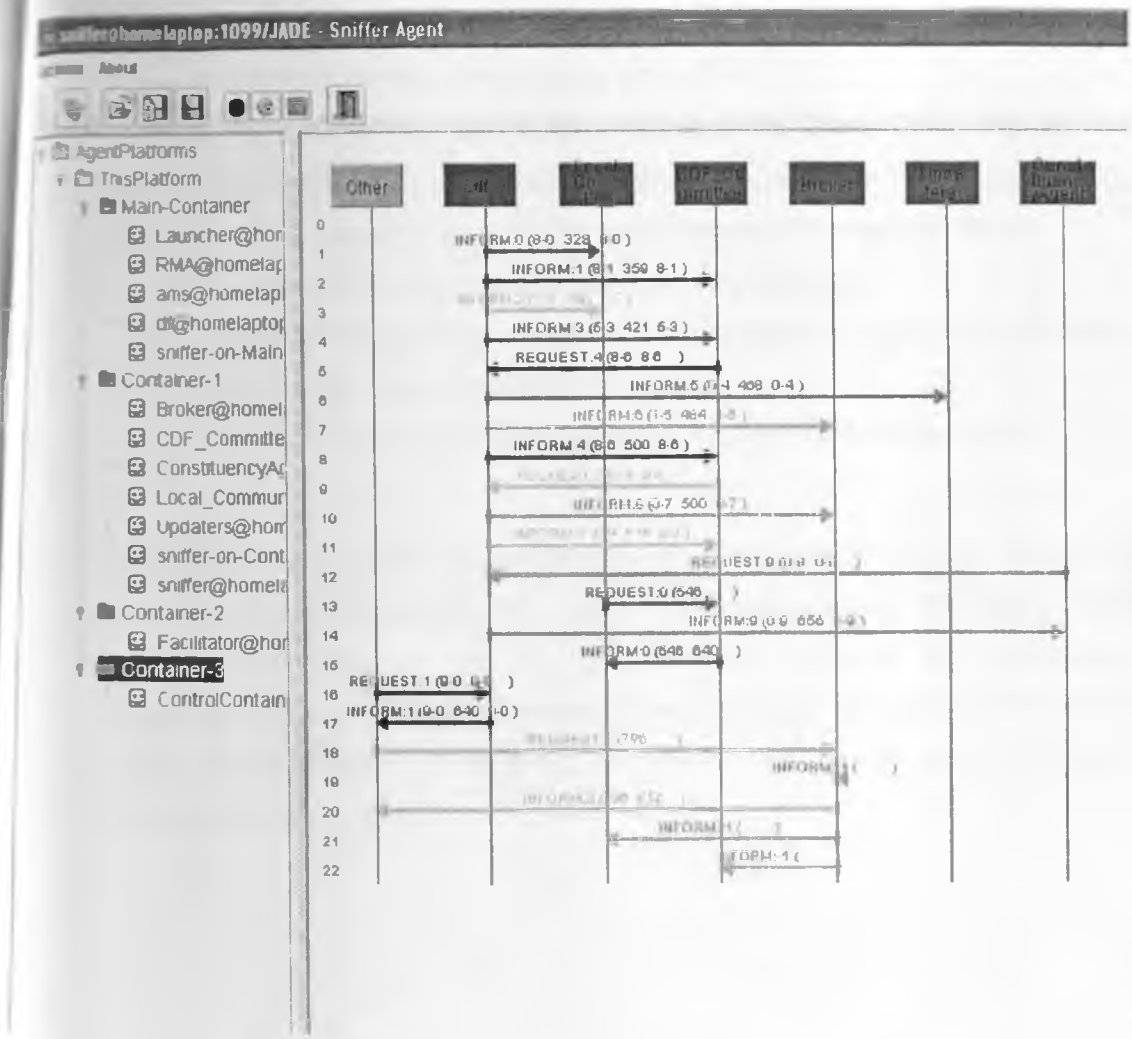


Figure 23: Testing the Agent life cycle

Agent life cycle entails granting that the agents live through the system life unless deliberately terminated. *Figure 23: Testing the Agent life cycle* is a snapshot of the sniffer window much after the brokering activity ends. It demonstrates that all the agents live much longer even after the system goes to idle mode, when all the operations are done.

Society Level Testing

During a society test, the validation of the overall results of the different agents is carried out and the successful integration of the different agents is verified. Testing communities of agents involves following issues:-

- Ensuring that the agents in the community work together as designed previously - This involves checking that each agent in the community/society receives the

correct messages from the correct agent, provides the correct responses, and interacts with environment correctly as a whole [71].

- Ensuring that the resultant work is the one expected - involve checking that the goal of this community or organization where the agents are interacting is being achieved. Listed below are the specific tests observed here among others:-
    - How the engine interacts with the Multi-agent system
    - Test whether the rules fire or not as illustrated in [*Figure 24: Brokering trace 1]* and *Figure 25: Brokering trace 2*
    - Inferencing results produced, whether they are explainable or not.

- **Scalability** - The larger the agent communities become, the harder it is to test them for proper functionality [71] Using traditional tools for debugging agent societies is insufficient, that is, they become inefficient and inadequate, especially because multi-agent systems are distributed systems. However, this work has identified a small group of agents and hence does not require advanced testing suites.

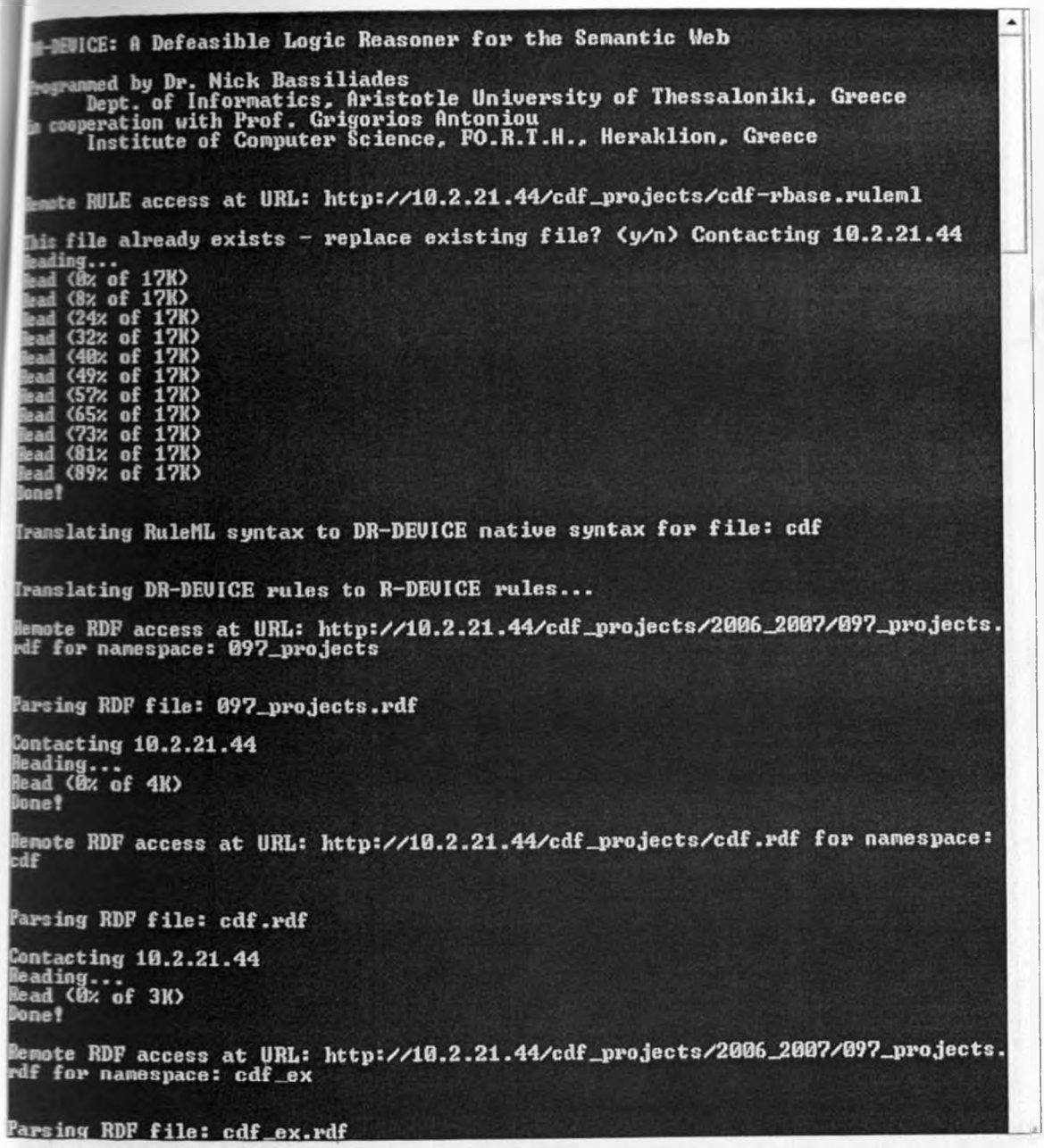


Figure 24: Brokering trace 1

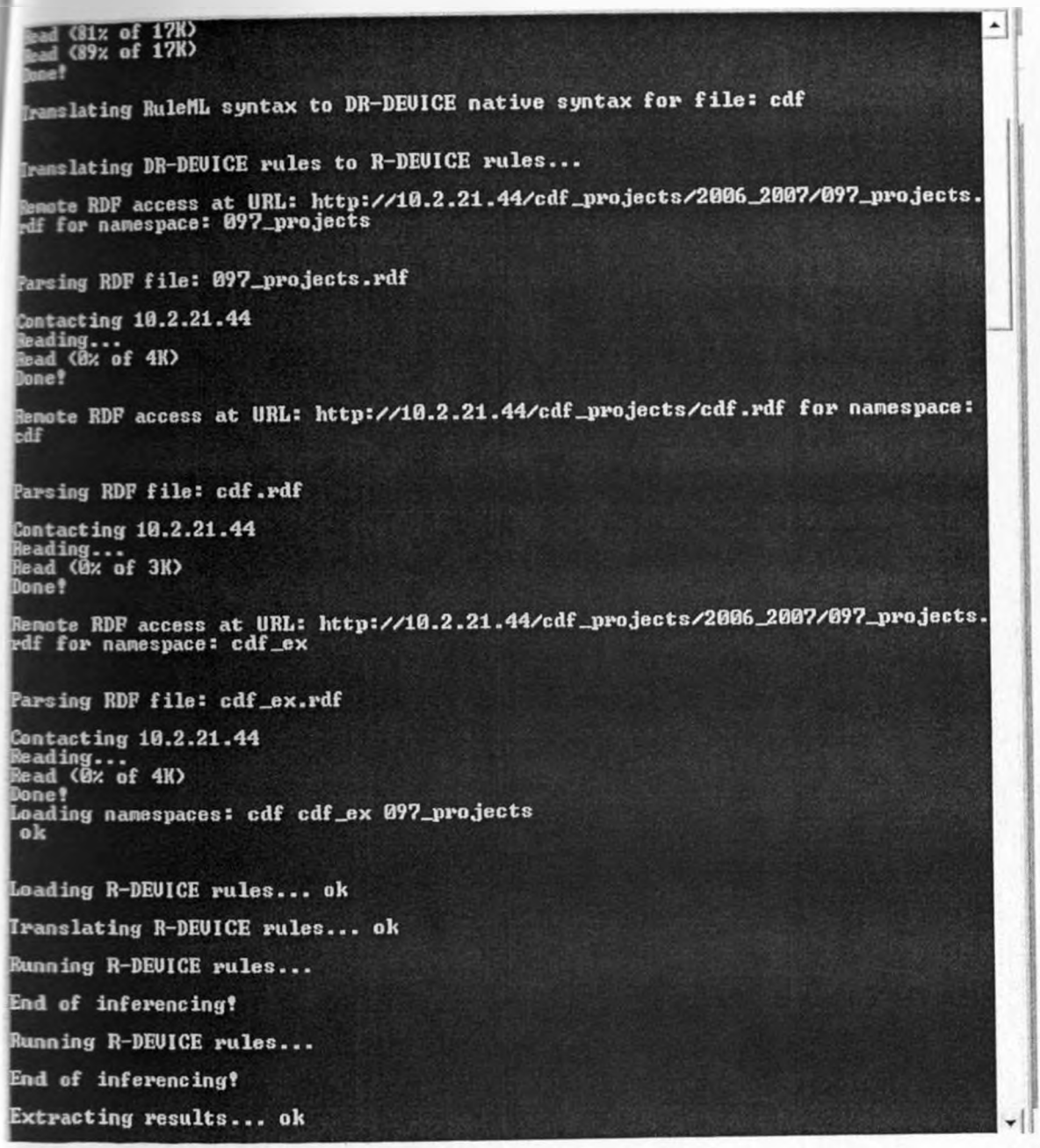


Figure 25: Brokering trace 2

5.3 Performance measures

The performance measures observed in the overall system can be classified and detailed as follows:-

- **Performance speed**

There are major modules of this system that determine the success or failure of the system in terms of performance. These modules include:

- **Preparation of the Project offers module-** This entails downloading of Project offers, conversion from XML to RDF format which is compatible with the inference engine and then uploading them back to the server.
- **Preparation of the rules module** –On every brokering instance, the user chooses different parameters: the particular constituency and year. The rules must therefore have the reference of the input file they are dealing with at any particular time. Consequently, on every brokering instance, the rules are parsed for correctness, reference to the specific input data added through the *rdf_import* attribute and the specification of the particular result export conclusions done that need to be displayed through the *rdf_export_classes* attribute.
- **Concurrency** – This is a test on whether the system can handle concurreny brokering activities. A concurrency test of the system has proved that the speed remain when two concurrent run are performed. This is illustrated in *Figure 27: Two concurrent inference sessions*
- **Inference Module** - This is the determining module as far as the results obtained are concerned. It is the "engine" of any brokering scenario. The time taken to get the inference report is an indication of the performance of the system and ensures that there is no infinite blocking of rules which may lead to occurrences of deadlocks. The system is designed with an option of indicating the amount of time taken after every brokering instance; see the Illustration at *Figure 26: Inference trace with a time report*. The more specific the rules are the more complex the inference rules are and this has a direct impact on the speed of inference.

gure 26: Inference trace with a time report

- **Agent Action speeds** – The agents must execute their goals within the shortest time possible.

- **Synchronization** – This involves making sure that all the modules execute at the right time and hence optimal usage of resources at a given moment. The system is therefore tested for proper synchronization especially now that there references to the same data by different agents at different times.

- **Validity of the results**: How do you measure the performance of brokering?
  - The kind of results produced, are they provable or not?

gure 27: Two concurrent inference sessions

5.4 *Results and Sample Programs*

This sub-section provides a detailed trace of the interactions, between the agents that participate in the brokering scenario and the results obtained at the end. It is a demonstration of the testing of the system as outlined in the earlier sub-section of this chapter.

5.4.1 Expression of Projects Offers

Every local community representative, who is a potential projects provider, can publish their offers on the web. A secured interface is provided where all constituencies fund allocations for all the years are provided and projects submission forms. A sample is shown in *Table 21: Projects Allocations and Submission Interface.* To submit an offer, a choice is first made for the particular constituency and the year whose projects offers are to be submitted.

The projects can either be submitted as a batch or individually. The illustration at *Table 20: The Local community interface after a successful login* shows clearly a link to both interfaces.

The representatives are the presented with a form where they can fill in the Project details. *Table 22: Submission Form* shows a sample of the form used .After a project offer publication, the offer is considered as a project advertisement. The advertisement is considered a complete project proposal and its XML format is depicted in *Figure 29: Expression of an Advertisement in XML.*

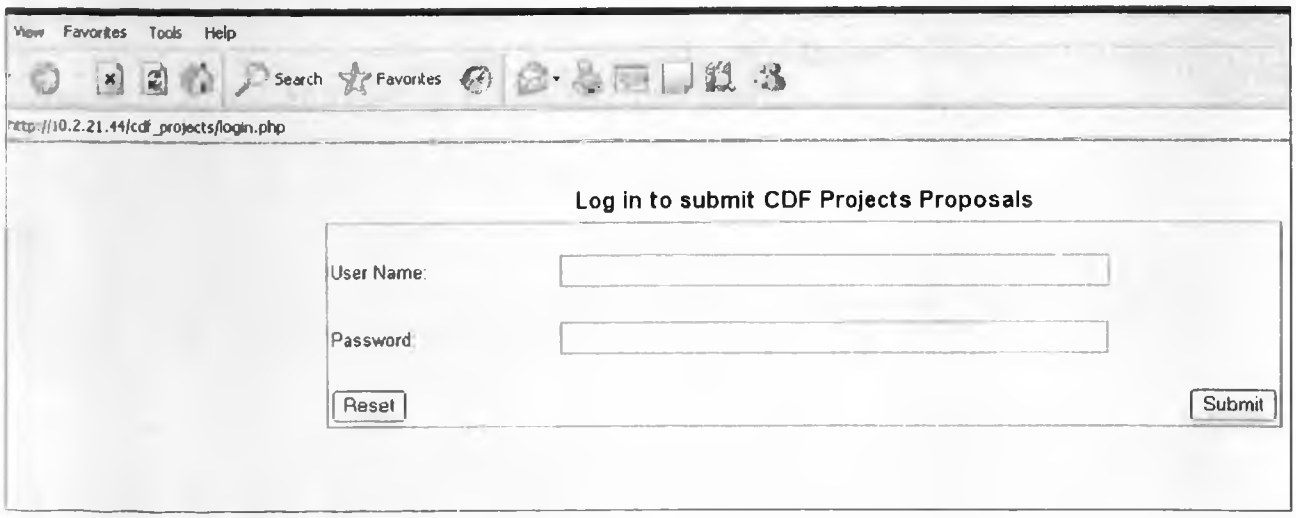


Figure 28: The local community secured interface

Table 20: The Local community interface after a successful login

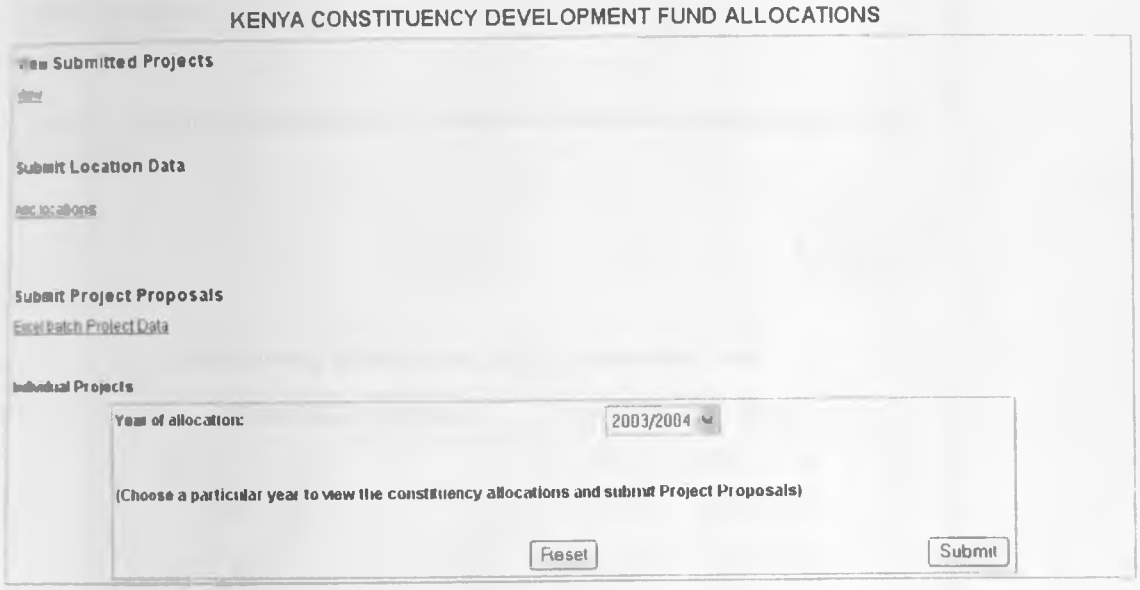


Table 21: Projects Allocations and Submission Interface

Submit Project Proposals

Year of allocation: 2003/2004

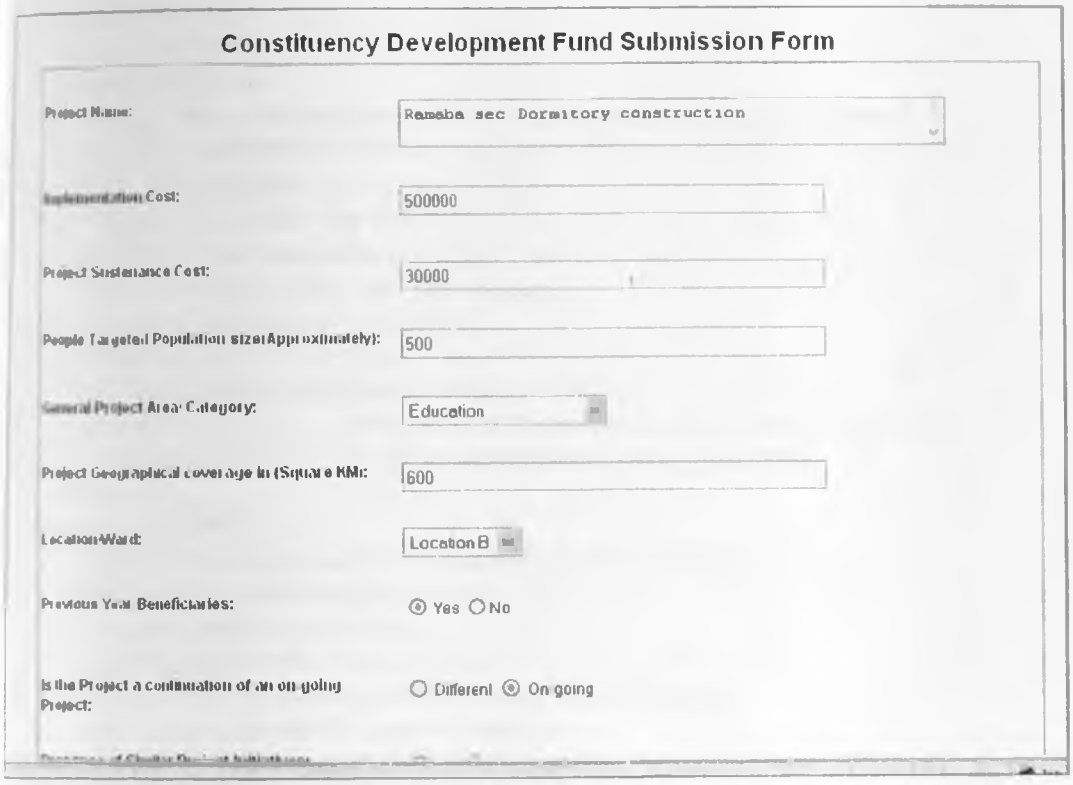
(Choose a particular year to view the constituency allocations and submit Project Proposals)

Reset Submit

Constituency Allocations as per Indicated Year

Code	Constituency Name	Amount	Year
001	MAKADARA	40,069,212	2007/2008
002	KAMUKUNJI	39,039,874	2007/2008
003	STAREHE	39,303,077	2007/2008
004	LANGATA	40,053,371	2007/2008
005	DAGORETTI	39,901,431	2007/2008
006	WESTLANDS	37,999,272	2007/2008
007	KASARANI	41,826,279	2007/2008
008	EMBAKASI	42,445,755	2007/2008
009	CHANGAMWE	38,914,543	2007/2008
010	KISAUNI	40,180,063	2007/2008
011	LIKONI	37,406,281	2007/2008
012	MVITA	36,584,989	2007/2008

Table 22: Submission Form



```
<?xml version="1.0" ?>
- <projects year="2007_2008" code="183">
  <allocation>49819646</allocation>
 - <project id="P1">
 + <project id="P2">
 - <project id="P3">
    <project-name>Kitambo Sec Lab construction</project-name>
    <imp-cost>3000000</imp-cost>
    <op-cost>100000</op-cost>
    <target-size>2000</target-size>
    <project-area>education</project-area>
    <geo-coverage>600</geo-coverage>
    <location>LocationF</location>
    <prev-imp>no</prev-imp>
    <project-status>different</project-status>
    <presence_of_similar_initiatives>no</presence_of_similar_initiatives>
    <complementary_proposal>no</complementary_proposal>
  </project>
 - <project id="P4">
    <project-name>DamsExcavations</project-name>
    <imp-cost>1000000</imp-cost>
    <op-cost>30000</op-cost>
    <target-size>5000</target-size>
    <project-area>poverty_reduction</project-area>
    <geo-coverage>4000</geo-coverage>
    <location>LocationD</location>
    <prev-imp>no</prev-imp>
    <project-status>on-going</project-status>
    <presence_of_similar_initiatives>yes</presence_of_similar_initiatives>
    <complementary_proposal>yes</complementary_proposal>
  </project>
 + <project id="P5">
 + <project id="P6">
```

Figure 29: Expression of an Advertisement in XML

The proposal includes the project name, implementation cost, operating cost, the project location, the project category among other things. For this scenario, an assumption is made that more than one representative can publish an advertisement, the system allows for this since it is web-based. These advertisements are captured in XML description/format; a sample can be found in *Figure 29: Expression of an Advertisement in XML* and later transformed into an RDF format (*Figure 30: Expression of an Advertisement in RDF*) compatible with the inference engine. The transformation is done by the Gateway agent that resides in the Java servlet.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE rdf:RDF (View Source for full doctype...)>
<rdf:RDF xmlns:cdf_ex="http://10.2.21.44/cdf_projects/2007_2008/183_projects.rdf#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:cdf="http://10.2.21.44/cdf_projects/projects.rdf#">
- <rdf:Description rdf:about="http://10.2.21.44/cdf_projects/2007_2008/183_projects.rdf#allocation">
    <cdf:allocation>49819646</cdf:allocation>
  </rdf:Description>
- <cdf:projects rdf:about="http://10.2.21.44/cdf_projects/2007_2008/183_projects.rdf#P1">
    <cdf:project-name>Misori Disp Construction</cdf:project-name>
    <cdf:imp-cost rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">400000</cdf:imp-cost>
    <cdf:op-cost rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">50000</cdf:op-cost>
    <cdf:target-size rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1000</cdf:target-size>
    <cdf:project-area>Health</cdf:project-area>
    <cdf:geo-coverage rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">3000</cdf:geo-coverage>
    <cdf:location>LocationA</cdf:location>
    <cdf:prev-imp>no</cdf:prev-imp>
    <cdf:project-status>different</cdf:project-status>
    <cdf:presence_of_similar_initiatives>yes</cdf:presence_of_similar_initiatives>
    <cdf:complementary_proposal>no</cdf:complementary_proposal>
  </cdf:projects>
- <cdf:projects rdf:about="http://10.2.21.44/cdf_projects/2007_2008/183_projects.rdf#P2">
    <cdf:project-name>Kasiri Pry Construction</cdf:project-name>
    <cdf:imp-cost rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">500000</cdf:imp-cost>
    <cdf:op-cost rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">100000</cdf:op-cost>
    <cdf:target-size rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">500</cdf:target-size>
    <cdf:project-area>education</cdf:project-area>
    <cdf:geo-coverage rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">600</cdf:geo-coverage>
    <cdf:location>LocationD</cdf:location>
    <cdf:prev-imp>no</cdf:prev-imp>
    <cdf:project-status>different</cdf:project-status>
    <cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>
    <cdf:complementary_proposal>no</cdf:complementary_proposal>
```

Figure 30: Expression of an Advertisement in RDF

This prototype also includes a XSLT[13] module (*Table 23: Expression of the Advertisements in XSLT format*) to view all the submitted projects in a user-friendly format (XML and RDF format are used by the inference engine) for a given constituency for a given year. Representatives from a given locality can therefore view the already submitted projects to avoid duplication.

[13] *XSLT is used to transform an XML document into another XML document, or another type of document that is recognized by a browser, like HTML and XHTML. Normally XSLT does this by transforming each XML element into an (X)HTML element.*

Table 23: Expression of the Advertisements in XSLT format

Year 2007/2008 RARIEDA Constituency

Project Submissions

Kasin Disp Construction	400000	50000	1000	Health	3000	LocationA	no
Ruma Community Hall	600000	30000	3000	others	4000	LocationA	no
Mukisembo Sec Sch bus	3500000	80000	2000	education	600	LocationB	no
Kisembe Ndori Water Project	3000000	100000	5000	poverty_reduction	6000	LocationC	yes
BansExcavations	1000000	30000	5000	poverty_reduction	4000	LocationD	no
Kasin Pry Construction	500000	100000	500	education	600	LocationD	no
Acces Roads	900000	50000	3000	environmental	3000	LocationE	no
Kisembe Sec Lab Construction	3000000	100000	2000	education	600	LocationF	no

5.4.2 Formalization of Requirements and Preferences

The CDF Act's requirements and the preferences (as stipulated in the CDF Act 2003) about acceptable offers are modeled through the following defeasible logic rules as depicted in *Figure 31: Defeasible DR-DEVICE rule in RuleML-like syntax.*

More rules and priorities could be used to express selection preferences among acceptable offerings. A complete file for the rules and preferences is found in the Appendix. To perform the reasoning, DR-DEVICE translates the RDF/RDFS descriptions of the available data (project offers) into logical facts, and the rules describing the preferences into logical facts and rules; it then uses rules capturing the semantics of RDF Schema to reason with the rules and the ontology data.

The version of DR-DEVICE used in this prototype does not support proof and trust capabilities. Even when a new release of DR-DEVICE system with proof capabilities has been released, there is no enough documentation of the syntax of the rules for this system that would facilitate implementation using this version and hence a proof file production. Consequently, proof files production will therefore be part of the future works of this platform.

```
<imp>
  <_rlab ruleID="r17" ruletype="defeasiblerule" superior="r15 r16">
    <ind href="&cdf_rb;r17">r17</ind>
  </_rlab>
  <_head>
    <atom>
      <_opr>
        <rel>implement</rel>
      </_opr>
      <_slot name="projects">
        <var>x</var>
      </_slot>
    </atom>
  </_head>
  <_body>
    <and>
      <atom>
        <_opr>
          <rel href="cdf:projects"/>
        </_opr>
        <_slot name="cdf:project-name">
          <var>x</var>
        </_slot>
        <_slot name="cdf:project-area">
          <ind>"poverty_reduction"</ind>
        </_slot>
      </atom>
      <atom>
        <_opr>
          <rel>cheapest</rel>
        </_opr>
        <_slot name="projects">
          <var>x</var>
        </_slot>
      </atom>
    </and>
  </_body>
</imp>
```

Figure 31: Defeasible DR-DEVICE rule in RuleML-like syntax.

5.5 *Brokering Trace*

Initially a Launcher agent, residing in the Main Container, which calls the rest community of agents (*CDF_Committee, Local_Community, ConstituencyAgent, Updaters and the Broker*), is started as can be depicted in [*Figure 32: Traces after the Agent Launcher is started*].

The five agent types are started in the second container in the Platform, named container 1.

they register themselves to the directory facilitator as depicted in [*Figure 32: Traces after the Agent Launcher is started]* and the latter sends them an acknowledgement message, *(REQUEST:7, INFORM:7, REQUEST:2 , INFORM:2, REQUEST:5 , INFORM:5)* respectively.

Local_Community Agent in turn searches the directory facilitator for the CDF_Committee and when he learns about his address, he sends him an "Avail_Allocations" message. This message is depicted *in Figure 21: Sniffed Message –Message sent by the Local Community Agent to the CDF Committee Agent*. The CDF_Committee responds by sending a message containing the address of the Project Allocations and submission interface and goes ahead to open the location of page with actual interface. The users interact with the system at this point to submit the project offers/definitions using this web interface. In the interface also is a view of the projects submitted earlier to avoid any duplication.

CDF_Committee representative then chooses the "Brokering" parameters. This submission interacts with the facilitator agent that resides in the Web server. The facilitator agent then forwards these parameters to the other Agents sub-system through the Broker agent in terms of a "Brokering" request.

In this message, the facilitator provides to the broker information such as, the URL address of the chosen project offers along with the brokering results the CDF-Committee is interested in and the language he uses for brokering.

On Brokering request message receipt, the broker agrees to brokering and initiates the activity by calling the engine. After the execution, the broker broadcasts a message with the results URL to the *CDF_Committee* and *Local_Community* Agents and goes ahead to open the results. Below is a samples of the results obtained.

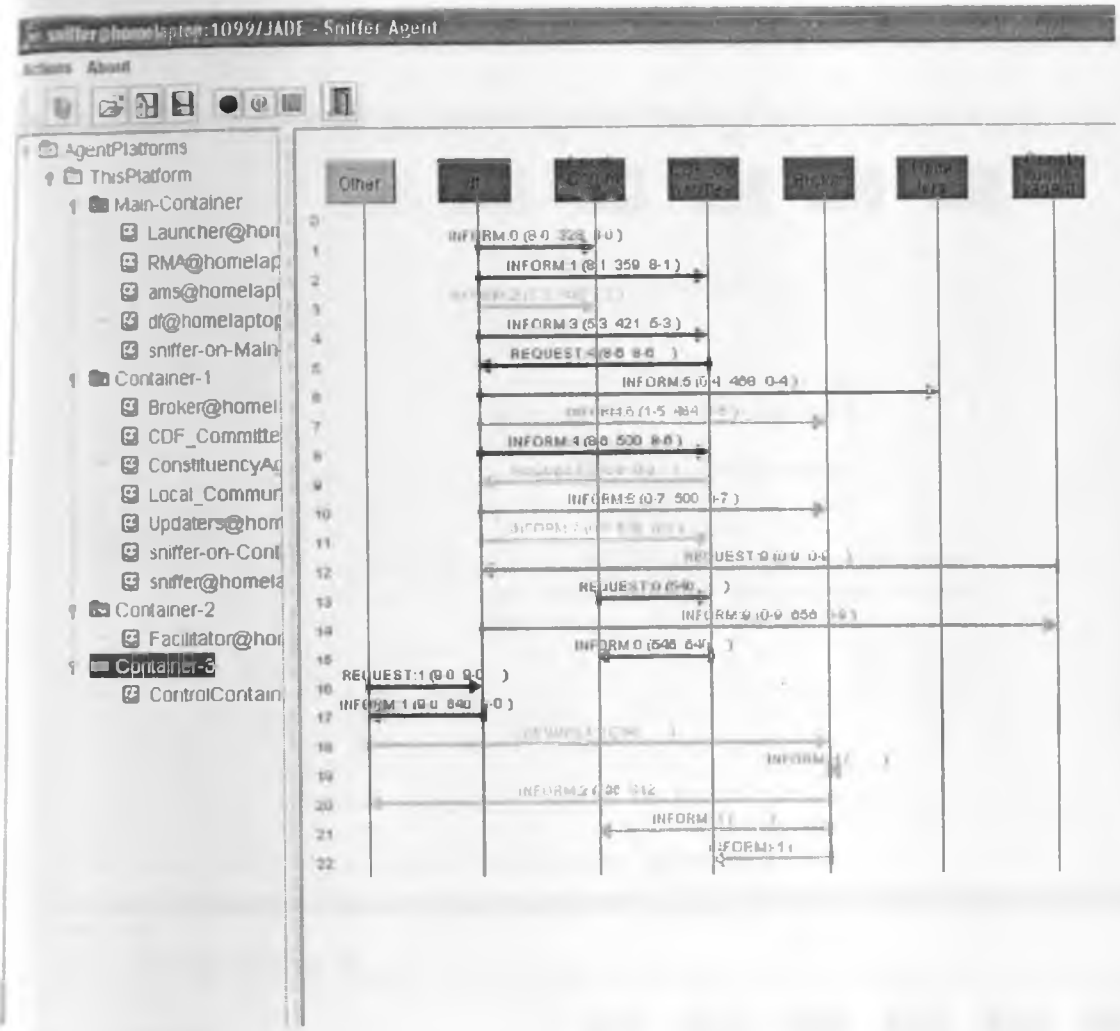


Figure 32: Traces after the Agent Launcher is started

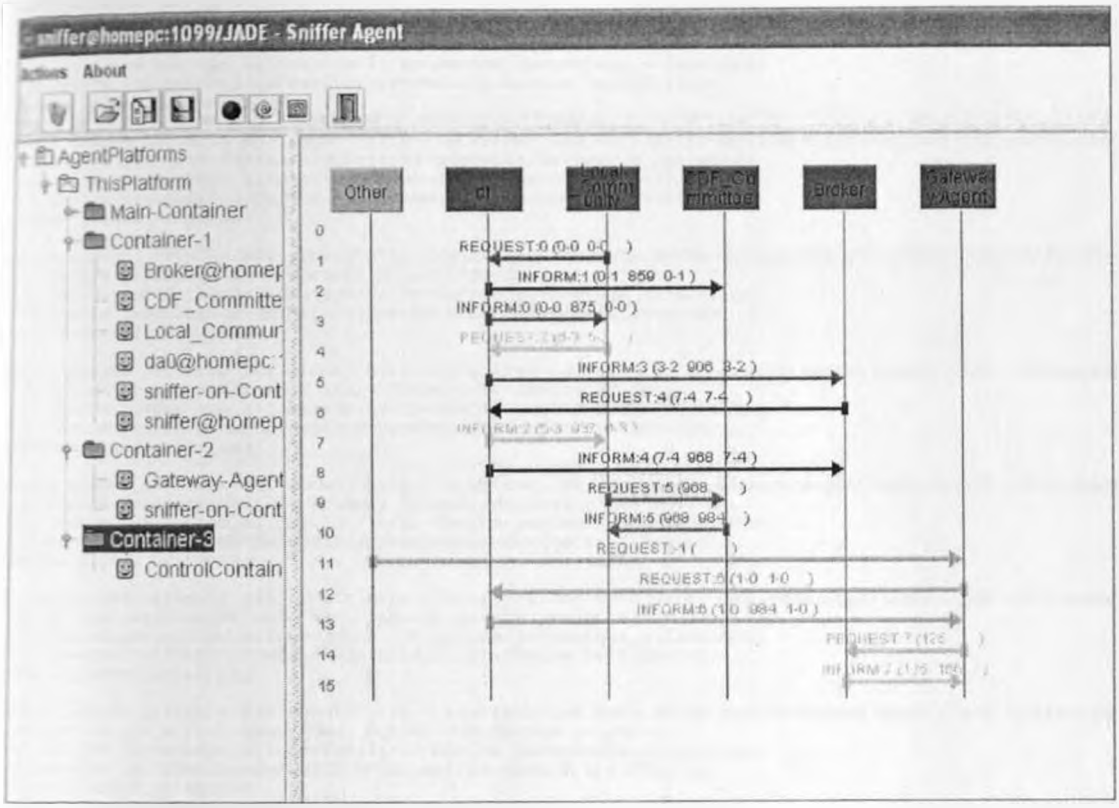


Figure 33: Traces after the submission of the brokering parameters

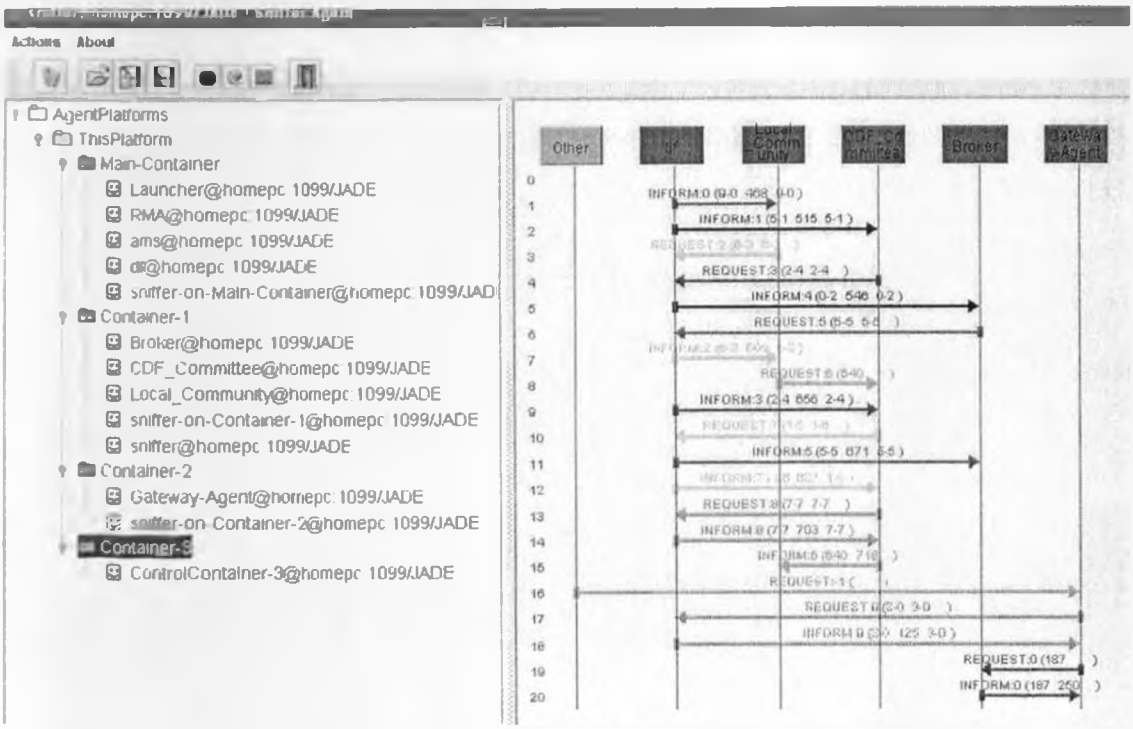


Figure 34: Illustration with Gateway Agent after the submission of Brokering Parameters

```
-device passed_criteria rdf:about="http://startrek.csd.auth.gr/dr-device/export/export-cdf.rdf#passed_criteria1">
  <dr-device:projects>Busulwa Primary School</dr-device:projects>
  <dr-device:percentage_allocation>1</dr-device:percentage_allocation>
  <dr-device:truthStatus>defeasibly-proven</dr-device:truthStatus>
</dr-device:passed_criteria>

<dr-device:passed_criteria rdf:about="http://startrek.csd.auth.gr/dr-device/export/export-cdf.rdf#passed_criteria2">
  <dr-device:projects>Muraka electricity project</dr-device:projects>
  <dr-device:percentage_allocation>6</dr-device:percentage_allocation>
  <dr-device:truthStatus>defeasibly-proven</dr-device:truthStatus>
</dr-device:passed_criteria>

<dr-device:passed_criteria rdf:about="http://startrek.csd.auth.gr/dr-device/export/export-cdf.rdf#passed_criteria3">
  <dr-device:projects>Ivona Nursery School</dr-device:projects>
  <dr-device:percentage_allocation>1</dr-device:percentage_allocation>
  <dr-device:truthStatus>defeasibly-proven</dr-device:truthStatus>
</dr-device:passed_criteria>

<dr-device:passed_criteria rdf:about="http://startrek.csd.auth.gr/dr-device/export/export-cdf.rdf#passed_criteria4">
  <dr-device:projects>Muraka Primary School</dr-device:projects>
  <dr-device:percentage_allocation>1</dr-device:percentage_allocation>
  <dr-device:truthStatus>defeasibly-proven</dr-device:truthStatus>
</dr-device:passed_criteria>

<dr-device:passed_criteria rdf:about="http://startrek.csd.auth.gr/dr-device/export/export-cdf.rdf#passed_criteria5">
  <dr-device:projects>Shirulu Primary School</dr-device:projects>
  <dr-device:percentage_allocation>1</dr-device:percentage_allocation>
  <dr-device:truthStatus>defeasibly-proven</dr-device:truthStatus>
</dr-device:passed_criteria>

<dr-device:passed_criteria rdf:about="http://startrek.csd.auth.gr/dr-device/export/export-cdf.rdf#passed_criteria6">
  <dr-device:projects>Shipalo Sec  Lab</dr-device:projects>
  <dr-device:percentage_allocation>2</dr-device:percentage_allocation>
  <dr-device:truthStatus>defeasibly-proven</dr-device:truthStatus>
</dr-device:passed_criteria>

<dr-device:passed_criteria rdf:about="http://startrek.csd.auth.gr/dr-device/export/export-cdf.rdf#passed_criteria7">
  <dr-device:projects>Lugango Sec  School</dr-device:projects>
  <dr-device:percentage_allocation>2</dr-device:percentage_allocation>
  <dr-device:truthStatus>defeasibly-proven</dr-device:truthStatus>
</dr-device:passed_criteria>

<dr-device:passed_criteria rdf:about="http://startrek.csd.auth.gr/dr-device/export/export-cdf.rdf#passed_criteria8">
  <dr-device:projects>Shipalo Pri. School</dr-device:projects>
  <dr-device:percentage_allocation>1</dr-device:percentage_allocation>
  <dr-device:truthStatus>defeasibly-proven</dr-device:truthStatus>
</dr-device:passed_criteria>
```

Figure 35: Raw Results Obtained from the engine

Table 24: Results obtained after XSLT format conversion

Projects Identification Results

Project Name	Truth Status	Recommendations	Offers	Allocation Percentage	Cumulative Percentage
Busuiwa Primary School	defeasibly-proven	passed_criteria		1%	1%
Muraka electricity project	defeasibly-proven	passed_criteria		6%	7%
Iwona Nursery School	defeasibly-proven	passed_criteria		1%	8%
Muraka Primary School	defeasibly-proven	passed_criteria		1%	9%
Shiulu Primary School	defeasibly-proven	passed_criteria		1%	10%
Shipalo Sec. Lab	defeasibly-proven	passed_criteria		2%	12%
Lugango Sec. School	defeasibly-proven	passed_criteria		2%	14%
Shipalo Pri School	defeasibly-proven	passed_criteria		1%	15%
Mukumu G Sec School	defeasibly-proven	passed_criteria		2%	17%
Mwitsa Pri School	defeasibly-proven	passed_criteria		1%	18%
Secheno Pri School	defeasibly-proven	passed_criteria		1%	19%
Musembe Dispensary	defeasibly-proven	passed_criteria		2%	21%
St Pauls Shibuye	defeasibly-proven	passed_criteria		1%	22%
Shagungu Pri ??	defeasibly-proven	passed_criteria		1%	23%
New Shivagala Pri School	defeasibly-proven	passed_criteria		1%	24%
Isenyende Pri School	defeasibly-proven	passed_criteria		1%	25%
Shiswa Primary School	defeasibly-proven	passed_criteria		1%	26%
Lirhanda Mixed Pri	defeasibly-proven	passed_criteria		1%	27%
Navangala Pri School	defeasibly-proven	passed_criteria		1%	28%
Chilovani Pri School	defeasibly-proven	passed_criteria		1%	29%

5.6 Final Results Discussion

The system produces a category of results depending with the needs of the CDF Committee.

Below is a list of the results that may be obtained from the system with a brief explanation.

- Acceptable Projects - projects that have passed all the criteria of acceptable projects as per the CDF Act. The rules are as stipulated in rules represented in section 4.6 of this document.
- Over 7000 target size CDF offer – All the acceptable projects with over 7000 targetted population and the flexibilities of the CDF Committee in terms of the cost ceiling if a certain amount is to be added per a certain group of people above 7000.

- Poverty Reduction CDF offer – All acceptable projects under the poverty reduction category and the ceiling cost that the committee would be willing to pay above the requested amount.
- Cheapest Project
- Project with the largest target Group

Table 25: *Results grouped per the Project Area* is a representation of the results grouped per all the indicated project areas in aid of knowing what percentage allocation is implemented per category.

Table 25: Results grouped per the Project Area

Projects Allocations by Project Area

Category: administrative	Cost	Target Size	Percentage Allocation (%)
Kayole Administration Police residential houses	500000	137866	1%
Sub-Total Percentage Allocation (%)			**1%**
Category: education	**Cost**	**Target Size**	**Percentage Allocation (%)**
Reuben Primary - 70 chairs	200000	86697	0%
Dandora primary - buy new desks	200000	154157	0%
Kayole Secondary-temporary- Dining Hall	300000	137866	1%
Reuben Centre School-Construction of class rooms	1000000	86697	2%
Mihango Primary School	1397742	13322	3%
Njiru Primary- Construction of classrooms	2000000	25251	5%
Kayole Secondary Dormitory	3000000	9413	7%
Dandora Secondary Sch (dormitory)	3000000	154157	7%
Utawalla village secondary (construction)	3000000	13322	7%
Sub-Total Percentage Allocation (%)			**32%**
Category: health	**Cost**	**Target Size**	**Percentage Allocation (%)**
Embakasi Primary - Repair of toilets	200000	32027	0%
Dandora Hospital Rehabilitation	500000	154157	1%
Soweto Social Hall toilet	920000	137866	2%
Tena Sewer Line	1000000	10149	2%
Tassia Sewer Line	1000000	10149	2%
Ruai Hospital - Rehabilitation	1000000	17531	2%
Embakasi Public Toilet -Rehabilitation	1000000	32027	2%
Sub-Total Percentage Allocation (%)			**11%**
Category: others	**Cost**	**Target Size**	**Percentage Allocation (%)**
Mukuru Roads	1000000	86697	2%
Tassia Road	2000000	10149	5%
Emergency reserve	2332619	32027	5%
Sub-Total Percentage Allocation (%)			**12%**
Category: poverty_reduction	**Cost**	**Target Size**	**Percentage Allocation (%)**
Tena Estate market gate	160000	10149	0%
Jehova Jire- Primary Energy saving jikos	200000	154157	0%
Njiru Market Shade	500000	25251	1%
Sub-Total Percentage Allocation (%)			**1%**
Total percentage			**57 %**

6 Conclusions and Further Work

A study was done of how projects identification is done, case of CDF Projects; that is how the requirements and preferences are matched against a set of project offers as submitted by the local community representatives.

The proposed solution uses the semantic web standard of RDF to represent the offers and a deductive logical language for expressing the requirements and preferences.

The motivation and the explanation of the approach proposed is presented; a report on the prototypical implementation exhibiting the desired functionality is then done in a Multi-agent Environment.

This approach has obvious advantages compared to other project identification approaches.

1. The algorithm used for identification is dynamic since it is the users who specify the algorithm on the basis of preferences.
2. The architecture provided is highly re-usable. The system can be applied in any domain with only the addition of a new ontology and new rules that capture the preferences.
3. A highly expressive language for preference specification is used with interesting features such as conflicting rules and priority among rules.
4. We use RDF (semantic web standard) for expression of advertisements. This choice supports interoperability among agents and applications; facilitates the easy publication, collection and dissemination of the project offers.
5. The system allows for specification of different export conclusions through the rdf-export-classes attributes.
6. Using JADE, the advantages of peer-to-peer systems are exploited and open FIPA agent communication and discovery standards are utilized.

6.1 Drawbacks of the System

1. The advertisements can not be removed automatically if need be.
2. An assumption is made that all the CDF projects can be submitted in the same format i.e. that there is a shared ontology.

6.2 *Further works*

In the future, this work will be extended in the following directions:

6.2.1 Extension of the proof module

The upper levels of the semantic web tower have not been researched enough and contain critical issues like accessibility, trust and credibility. This poses a big challenge on the results produced. The main difference a query posed to a traditional database system and a semantic web system is that the answer in the first case is returned from a given collection of data while for the semantic web system the answer is the result of a reasoning process. While in some cases the answer speaks for itself, in other cases the user will not be confident in the answer unless he/she can trust the reasons why the answer has been produced. Thus, for a semantic web system to gain the trust of a user must be able, if required, to provide an explanation or justification for an answer.

The justification can be given as a derivation of the conclusion with the sources of information for the various steps. The justification as it stands now is in logic, which is quite unfamiliar with most users.

6.2.2 *Advertisement Removal Utility*

A utility should be added to remove advertisement once they become invalid or expired. A system with a user interface which is more integrated.

6.2.3 *Dynamic Rule file*

Further works would involve exploring the possibility of making the rule file dynamic. Currently the rule file may only be changed by the system designers which bring about some rigidity.

7 References

[1] *About CDNow,* CDNow Online Inc., 1005 Virginia Dr. Ft. Washington, PA 19034 Retrieved August 2007, from http://www.cdnow.com

[2] *Agent Construction Tools.* Retrieved May 5, 2004, from http://www.agentbuilder.com/AgentTools/index.html

[3] Antoniou G., M. J. Maher and D. Billington (2000). *Defeasible Logic versus Logic Programming without Negation as Failure*. Journal of Logic Programming 41,1

[4] Antoniou G., D. Billington, G. Governatori and M.J. Maher (2001). *Representation results for defeasible logic.* ACM Transactions on Computational Logic 2, 2 (2001): 255- 287

[5] Antoniou Grigoris, Frank van Harmelen (2004). *A Semantic Web Primer.* MIT Press 2004.

[6] ARPA Knowledge Sharing Initiative. *Specification of the KQML agent-communication language*. ARPA Knowledge Sharing Initiative, External Interfaces Working Group, July 1993.

[7] Bassiliades N., Vlahavas I. (2003). *Capturing RDF Descriptive Semantics in an Object Oriented Knowledge Base System.* Electronic Poster Proc. 12th Int. WWW Conf. (WWW2003), 20-24 May 2003, Budapest, Hungary.

[8] Bassiliades N., Antoniou G., Vlahavas I (2004). *DR-DEVICE: A Defeasible Logic System for the Semantic Web.* Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR04), Sebastian Schaffert (Ed.), Springer-Verlag, LNCS 3208, pp. 134-148, St Malo, France, Sept. 2004.

[9] Bassiliades N., Antoniou G., Vlahavas I. (*2004*). *A Defeasible Logic Reasoner for the Semantic Web*, Third International Workshop on Rules and Rule Markup Languages for the Semantic Web (RuleML 2004), G.Antoniou, H. Boley (Ed.), Springer-Verlag, LNCS 3323, pp. 49-64, Hiroshima, Japan, 8 Nov. 2004.

[10] Bassiliades N., Antoniou G., Vlahavas I. (2004). *DR-DEVICE: A Defeasible Logic RDF Rule Language.* Demo, Demonstration at 3rd

International Semantic Web Conference (ISWC2004), 7-11 November 2004, Hiroshima, Japan. Retrieved June 12, 2001, from http://iswc2004.semanticweb.org/demos/index.html.

[11] D. Beckett (2004). *RDF/XML Syntax Specification*, W3C Recommendation, Retrieved February 2004 from World Wide Web Consortium: http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/

[12] Bellifemine F., Poggi, A., Rimassa, G. (2001). *JADE - A FIPA2000 Compliant Agent Development Environment*. In Proc. Agents Fifth International Conference on Autonomous Agents (Agents 2001), pp. 216-217, Montreal, Canada, 2001

[13] Bellifemine F., G Caire, Poggi, A., Rimassa, G. (2003). *JADE a White Paper*. Telecom Italia EXP magazine Vol 3, No 3 September 2003.

[14] F. Bergenti, A. Poggi (2000). *Exploiting UML in the Design of Multi-Agent Systems*. In A. Omicidi, R. Tolksdorf, F. Zambonelli, eds., Engineering Societies in the Agents World - Lecture Notes on Artificial Intelligence, volume 1972, pp 106-113, 2000. Berlin, Germany, Springer Verlag Publ.

[15] Bray Tim, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler (2000). *Extensible Markup Language (XML) 1.0 (Second Edition)*. Retrieved from World Wide Web Consortium, W3C Recommendation at: http://www.w3.org/TR/2000/REC-xml-20001006.

[16] D Brickley, RV Guha (2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. Retrieved from World Wide Web Consortium, W3C Recommendation at: http://www.w3.org/TR/2004/REC-rdf-schema-20040210

[17] Caire, G., Coulier, W., Garijo, F., Gomez, J., Pavon, J., Leal, F., Chainho, P., Kearney, P., Stark, J., Evans, R., and Massonet P. (2001). *Agent Oriented Analysis Using Message/UML*. In M. Wooldridge, P. Ciancarini, and G. Weiss, editors, Second International Workshop on Agent-Oriented Software Engineering (AOSE-2001), pages 101–108.

[18] Caire, G., Cossentino, M., Negri, A., Poggi, A., and Turci, P.(2004). *Multi-agent systems implementation and testing.* In Fourth International Symposium: From Agent Theory to Agent Implementation (AT2AI-4), Vienna, Austria (EU).

[19] YC Chen, W-T. Hsu, P-H. Hung (2003). *Towards Web Automation by Integrating Semantic Web and Web Services.* In Proceedings of the. 12th International Conference on WWW. Budapest Hungary 2003.

[20] A. Collinot, A. Drogoul, and P. Benhamou (1996). *Agent oriented design of a soccer robot team.* In Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96), pages 41–47, Kyoto, Japan.

[21] *Constituency Development Fund.* Retrieved May 2009 from: www.cdf.go.ke

[22] R. S. Cost, T. Finin, and Y. Labrou (2001). *Coordinating Agents Using ACL Conversations.* In Coordination of Internet Agents: Models, Technologies, and Applications, pages 183–196, 2001.

[23] Cranefield, S., and Purvis, M. (1999). *UML as an ontology modeling language.* In Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99) Stockholm, Sweden.

[24] Crosby B. 1992. *Stakeholder analysis: A vital tool for strategic Managers.* Washington DC: USAID

[25] *Decentralized Funds report.* Kenya Institute for Public Policy Research and Analysis (KIPPRA). Retrieved Jan 2007 from http://www.kippra.org/DGSP-Booklet.doc

[26] J. E. Doran, S. Franklin, N. R. Jennings & T. J. Norman (1996).*On Cooperation in Multi-Agent Systems.* First UK Workshop on Foundations of Multi-Agent Systems

[27] E-Bay, Retrieved May 2007 from http://www.ebay.com

[28] Engel J. and R. Blackwell. (1982). *Consumer Behavior.* CBS College Publishing.

[29] *FIPA ACL Message Structure Specification (2002).* Retrieved October 2002 from http://www.fipa.org/specs/fipa00061/SC00061G.html.

[30] *FIPA: Interaction Protocol Specifications* (Retrieved May 2002 from http://www.fipa.org/repository/ips.php3)

[31] *FIPA Website.* The Foundation for Intelligent Physical Agents specification Retrieved May 5 2004: www.**fipa**.org/

[32] Roberto A. Flores-Mendez (1999).*Towards a Standardization of Multi-Agent System Frameworks.* ACM Crossroads Journal – Intelligent Agents. Summer 1999.

[33] Gary R. Heerkens, PMP (2002) *Project Management.* New York: McGraw-Hill

[34] M. R. Genesereth and S. P. and Ketchpel (1994. *Software Agents.* Communication of the ACM, vol. 37(7).

[35] GOK (2003); the Constituencies Development Fund Act, Government Printer.

[36] B. N. Grosof (1997). *Prioritized conflict handing for logic programs.* In Proceedings of the 1997 International Symposium on Logic Programming, MIT Press, Cambridge, 197-212.

[37] Thomas R. Gruber (1993). *A translation approach to portable ontology specifications.* Knowledge Acquisition, 2:199-220, 1993.

[38] P. Haase, J. Broekstra, A. Eberhart, R. Volz (2004). *A Comparison of RDF Query Languages.* In Proceedings of the 3rd International Semantic Web Conference (ISWC) Number 3298 in LNCS, Springer-Verlag. pages 502–517

[39] L. Hampton, R. C. Martin, F. G. Pitt, and T. Ottinger. *A Critique of Use Cases.* Retrieved July 09, 2008 from: http://ootips.org/use-cases-critique.html

[40] Hayes P., *RDF Semantics.* W3C Recommendation, Retrieved February 2004 from http://www.w3.org/TR/rdf-mt/.

[41] Minghua He, Nicholas R. Jennings, Ho-Fung Leung (2003). *On Agent-Mediated Electronic Commerce. IEEE Transactions on Knowledge and Data Engineering Vol.* 15, No. 4.

[42] *History Solutions*. Retrieved July 2009 from http://www.historysolutions.com/servlet/prezview?get=template&TemplateName=Glossary.htm&letter=C

[43] Li, L., Horrocks, I. (2003). *A Software Framework for Matchmaking Based on Semantic Web Technology*. In Proceedings of the 12th International World Wide Web Conference (WWW 2003), pages 331-339 , ACM

[44] Huget, M.P., Demazeau, Y., (2004). *Evaluating multi-agent systems: a record/replay* approach. Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)

[45] *Introduction into German constitutional law*. Staatsrecht. Retrieved June 2008 from http://staatsrecht.honikel.de/en/glossar.htm?w=C

[46] *Java Agent Development*. Retrieved July 2008 from: http://jade.cselt.it

[47] *JavaServer Pages (JSP) Technology*, Sun Developer Network. Retrieved December 2008 from http://java.sun.com/products/jsp/.

[48] Yannis Labrou and Tim Finin (1997). *A proposal for a new kqml specification*. Technical Report Technical Report TR-CS-97-03, University of Maryland Baltimore County, 1997.

[49] Y Labrou, T Finin, Y Peng (1999). *Agent Communication Languages: The current Landscape*. IEEE Intelligent Systems Journal March/April 1999.

[50] McBride B. (2001), *Jena: Implementing the RDF Model and Syntax Specification*. In: Steffen Staab et al (eds.): Proceedings of the Second International Workshop on the Semantic Web - SemWeb'2001.

[51] D.L. McGuinness, F. van Harmelen (2004). *OWL Web Ontology Language Overview* W3C Recommendation, Retrieved February 2004 from http://www.w3.org/TR/owl-features/

[52] B McLaughlin, (2001). *Java & XML, 2nd Edition*. O'Reilly Second Edition September 2001, pp. 90-137, 2001.

[53] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout (1991). *Enabling technology for knowledge sharing*. AI Magazine, 12(3):36-56, Fall 1991

[54] Nguyen T. Giang, Dang T. Tung (2002). Agent Platform Evaluation and Comparison. Technical report for Pellucid 5FP IST-2001-34519. Bratislava, Slovakia.

[55] Nikraz, M., Caire, G., and Bahri, P.A. (2006). *A methodology for the development of multi-agent systems using the jade platform.* Computer Systems Science and Engineering, vol. 21, no. 2, pp. 99–116.

[56] M. Nodine, W. Bohrer, A. Hee Hiong Ngu (1998). *Semantic Brokering over Dynamic Heterogeneous Data Sources in InfoSleuth.* In Proc. 15th International Conference on Data Engineering (ICDE), IEEE Computer Society, pages 358–365.

[57] D. Nute (1994). *Defeasible logic.* In D.M. Gabbay, C.J. Hogger and J.A. Robinson (eds.): Handbook of logic in artificial intelligence and logic programming (vol. 3): nonmonotonic reasoning and uncertain reasoning. Oxford University Press, 353–395

[58] J. Odell (2000). *Objects and agents: how do they differ?.* Journal of Object-Oriented Programming 13(6), 50–53. Retrieved May 2009 from www.jamesodell.com/publications.html

[59] Peterson, D., (2001). *About Net Perception.* Net Perception Inc. Retrieved December 2001 from http://www.netperception.com.

[60] Pressman, R. S. (2002). *Engenharia de Software*, 5th ed., Rio de Janeiro, McGraw-Hill

[61] C Rouff, A SAIC (2002). *A Test Agent for Testing Agents and Their Communities.* Aerospace Conference Proceedings, 2002. IEEE Volume 5, 2002 Page(s):5 - 2638 vol.5 Digital Object Identifier 10.1109/AERO.2002.1035446

[62] RuleML. *The Rule Markup Language Initiative*, Retrieved 26th February 2005 from http://www.ruleml.org

[63] Searle, J.R. (1969). *Speech Acts: An essay in the philosophy of language.* Cambridge University Press

[64] Sommerville, I. (2000). *Software Engineering (Sixth edition).* Addison Wesley Longman.

[65] *The Model-View-Controller*, Java Blue Prints. Retrieved Nov 2004 from http://java.sun.com/blueprints/patterns/MVC-detailed.html

[66] Trastour, D., Bartolini, C. and Gonzalez-Castillo, J. (2001). *A Semantic Web Approach to Service Description for Matchmaking of Services*. In Proceedings of the Semantic Web Working Symposium, Stanford, California, USA, pp. 447-461, July 30 - August 1 2001

[67] *Unified Modeling Language (UML)*. Retrieved March 2009 from http://www.uml.org/

[68] A. van Gelder, K. Ross and J. Schlipf (1991). *The well-founded semantics for general logic programs*. Journal of the ACM 38 (1991): 620—650

[69] *Why, When, and Where to Use Software Agents*. Retrieved May 5 2004 from Agent Builder at http://www.agentbuilder.com/Documentation/whyAgents.html

[70] M. Wooldridge, N. R. Jennings, and D. Kinny (2000). *The gaia methodology for agent-oriented analysis and design. International Journal of Autonomous Agents and Multi-Agent Systems*, vol. 3(3), pp. 285-312.

[71] M. Wooldridge. (2002). *An Introduction to Multi-agent Systems*. Published by John Wiley and Sons, March.

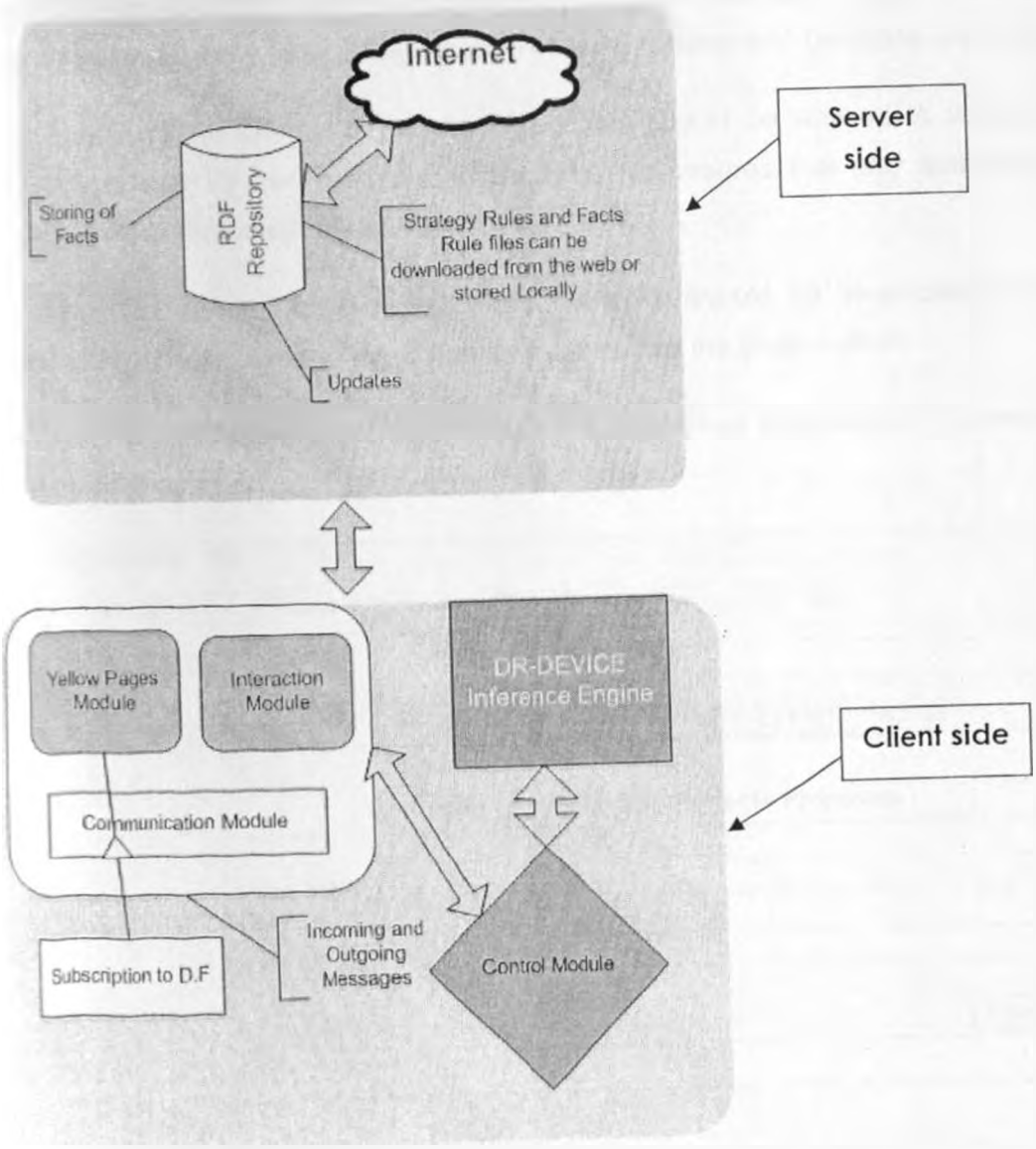
[72] *Xalan-Java XSLT processor.* Retrieved October 2006 from *http://xml.apache.org/xalan-j/*

8 APPENDICES

8.1 *User Guide and Installation Manual*

Introduction

The CDF Project identification system is packaged to have a server and a client side as shown in the diagram below.



The server side system does not need any installation unless there is migration from one server to another.

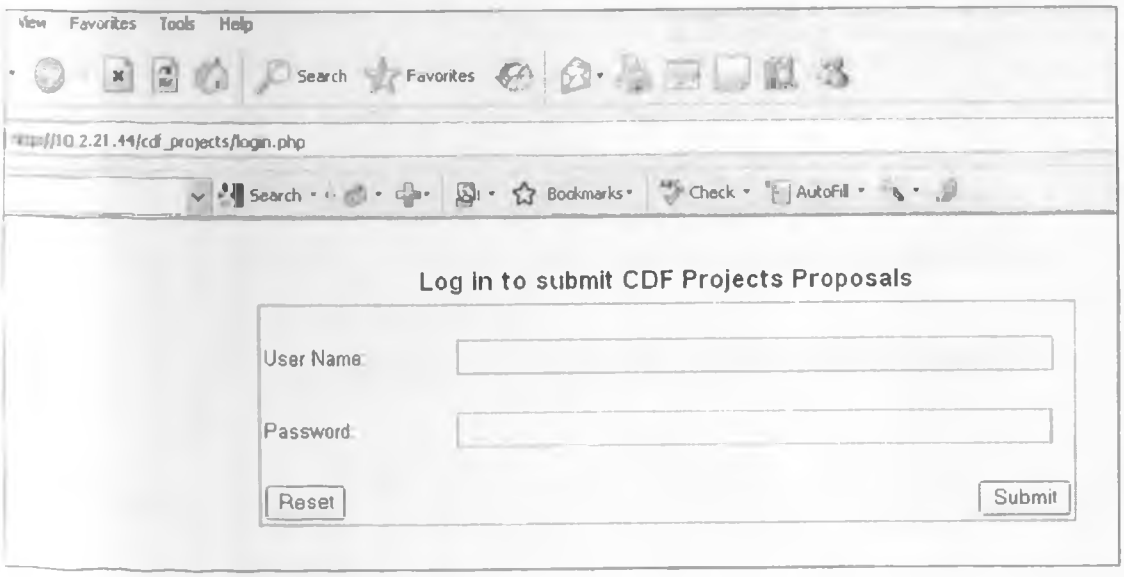
Should migration happen then, the server side system files are in a folder called ***"cdf_projects"*** which contains all the necessary operation files. These files include the ontology files, Rule Files and all the others that make up the RDF repository module. These files are packaged together with the in the same directory structure with the ***"coreservlets",*** the client package.

This server should also have PHP 5.5 or later as the scripting language with the XML and XSLT functionalities installed. Additionally, it should also host a MySQL database, where the constituency data, Constituency Yearly Allocations and Locations are hosted.

The client machine Internet Protocol (IP) address should be allowed to access the MySQL database in the MySQL configuration. This ensures that only authenticated machines can interact with the server.

On successful set-up of the server side, The URL should be broadcasted to the stakeholders (Local Community) so that they can submit the project offers.

Below is a sample server side console window in the current installation at the writing of these guidelines.



This server side interaction of the system is secured so that only the authenticated users can perform submissions. The authentications are controlled by a user table in the MySQL database.

Client Installation

➢ *Software Requirements*

Software necessary in order to run the system includes:-

- Valid installation of Java 1.5.0 and above
- JADE 3.5 and Above – Agent Platform
- DR-DEVICE Engine- The core reasoning engine
- Tomcat Server 5.5 and above

➢ *Tested Environments*

The system has been successfully installed and operated using the following settings:

- Windows 2000 & JRE 1.5
- Windows XP & JRE 1.6.0

The system is designed to operate on any Operating System that can support Java Run Time Environment 1.5 or higher.

➢ *First-Time Installation Procedure*

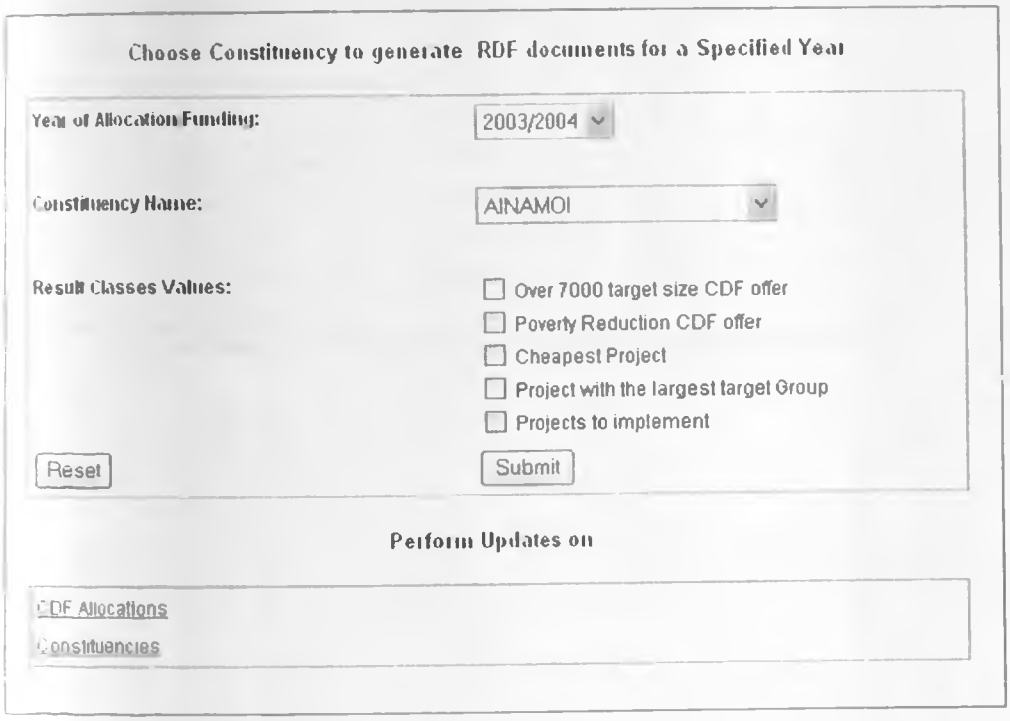
- Access the published URL for the CDF Project Identification Project.
- Download the ***coreservlets*** zipped package and un-package it in the Tomcat Development Directory
- Copy the ***CDFAgents.bat*** Batch file in the desktop, the Launcher of the system

➢ *Usage*

Launch a program

- Click on the *CDFAgents* icon 

on the Desktop to launch the Agent system – All the agents are launched at this point, ready to co-ordinate the other operations of the system.
- On the Launch, the CDF Committee console is opened on browser window, where the brokering parameters may be submitted to the agent system as shown in the image below.



This console is used to submit the particular constituency and year details whose project identification results need be brokered.

The user then chooses from the results class values the category of results they are interested in.

Once the parameters are chosen, the agents get to invoke the engine which performs the inferencing and opens the results on a browser window as shown below.

Projects Allocations by Project Area

Category: administrative	Cost	Target Size	Percentage Allocation (%)
Kayole Administration Police residential houses	500000	137866	1%
Sub-Total Percentage Allocation (%)			1%
Category: education	**Cost**	**Target Size**	**Percentage Allocation (%)**
Reuben Primary - 70 chairs	200000	86697	0%
Dandora primary - buy new desks	200000	154157	0%
Kayole Secondary-temporary- Dining Hall	300000	137866	1%
Reuben Centre School-Construction of class rooms	1000000	86697	2%
Mihango Primary School	1397742	13322	3%
Njiru Primary- Construction of classrooms	2000000	25251	5%
Kayole Secondary Dormitory	3000000	9413	7%
Dandora Secondary Sch (dormitory)	3000000	154157	7%
Utawalla village secondary (construction)	3000000	13322	7%
Sub-Total Percentage Allocation (%)			32%

8.2 *Sample Programs*

Project Domain Ontology

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
        <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
        <!ENTITY cdf "http://10.2.21.44/cdf_projects/projects.rdf#">
        <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
        <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDF xmlns:cdf="http://10.2.21.44/cdf_projects/projects.rdf#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
        <rdfs:Class rdf:about="&cdf;allocation" rdfs:label="allocation">
                <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
                <rdfs:range rdf:resource="&xsd;integer"/>
        </rdfs:Class>
        <rdfs:Class rdf:about="&cdf;projects" rdfs:label="projects">
                <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
                <rdfs:label>projects</rdfs:label>
        </rdfs:Class>
        <rdf:Property rdf:about="&cdf;project-name" rdfs:label="project-name">
                <rdfs:domain rdf:resource="&cdf;projects"/>
                <rdfs:range rdf:resource="&rdfs;Literal"/>
        </rdf:Property>
        <rdf:Property rdf:about="&cdf;imp-cost" rdfs:label="imp-cost">
                <rdfs:domain rdf:resource="&cdf;projects"/>
                <rdfs:range rdf:resource="&xsd;integer"/>
        </rdf:Property>
        <rdf:Property rdf:about="&cdf;op-cost" rdfs:label="op-cost">
```

```
        <rdfs:domain rdf:resource="&cdf;projects"/>
        <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:about="&cdf;prev-imp" rdfs:label="prev-imp">
        <rdfs:domain rdf:resource="&cdf;projects"/>
        <rdfs:range rdf:resource="&xsd;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&cdf;project-status" rdfs:label="project-status">
        <rdfs:domain rdf:resource="&cdf;projects"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&cdf;target-size" rdfs:label="target-size">
        <rdfs:domain rdf:resource="&cdf;projects"/>
        <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:about="&cdf;project-area" rdfs:label="project-area">
        <rdfs:domain rdf:resource="&cdf;projects"/>
        <rdfs:range rdf:resource="&xsd;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&cdf;geo-coverage" rdfs:label="geo-coverage">
        <rdfs:domain rdf:resource="&cdf;projects"/>
        <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:about="&cdf;complementary_proposal" rdfs:label="complementary-proposal">
        <rdfs:domain rdf:resource="&cdf;projects"/>
        <rdfs:range rdf:resource="&xsd;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&cdf;presence_of_similar_initiatives"
        rdfs:label="presence-of-similar-initiatives">
        <rdfs:domain rdf:resource="&cdf;projects"/>
        <rdfs:range rdf:resource="&xsd;Literal"/>
```

```
    </rdf:Property>
    <rdf:Property rdf:about="&cdf;location" rdfs:label="location">
        <rdfs:domain rdf:resource="&cdf;projects"/>
        <rdfs:range rdf:resource="&xsd;Literal"/>
    </rdf:Property>
</rdf:RDF>
```

Submitted project offers for TETU Constituency, Year 2006/2007

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY cdf "http://10.2.21.44/cdf_projects/projects.rdf#">
<!ENTITY cdf_ex "http://10.2.21.44/cdf_projects/2006_2007/081_projects.rdf#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDF xmlns:cdf="http://10.2.21.44/cdf_projects/projects.rdf#"
xmlns:cdf_ex=http://10.2.21.44/cdf_projects/2006_2007/081_projects.rdf#
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
<rdf:Description rdf:about="&cdf_ex;allocation"><cdf:allocation>39739318</cdf:allocation>
</rdf:Description>

##### Project Offer 1

        <cdf:projects rdf:about="&cdf_ex;P1">
                <cdf:project-name>Thagei-ini police post</cdf:project-name>
                <cdf:imp-cost rdf:datatype="&xsd;integer">351813</cdf:imp-cost>
                <cdf:op-cost rdf:datatype="&xsd;integer">50000</cdf:op-cost>
                <cdf:target-size rdf:datatype="&xsd;integer">3000</cdf:target-size>
                <cdf:project-area>human_rights</cdf:project-area>
                <cdf:geo-coverage rdf:datatype="&xsd;integer">500</cdf:geo-coverage>
                <cdf:location>LocationA</cdf:location>
                <cdf:prev-imp>yes</cdf:prev-imp>
                <cdf:project-status>on-going</cdf:project-status>
                <cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>
                <cdf:complementary_proposal>no</cdf:complementary_proposal>
```

```
        </cdf:projects>
#### Project Offer 2
        <cdf:projects rdf:about="&cdf_ex;P2">
        <cdf:project-name>Kaigonde Secondary school - Construction of Laboratory and dinning hall
                </cdf:project-name>
                  <cdf:imp-cost rdf:datatype="&xsd;integer">800000</cdf:imp-cost>
                  <cdf:op-cost rdf:datatype="&xsd;integer">30000</cdf:op-cost>
                  <cdf:target-size rdf:datatype="&xsd;integer">3000</cdf:target-size>
                  <cdf:project-area>education</cdf:project-area>
                  <cdf:geo-coverage rdf:datatype="&xsd;integer">500</cdf:geo-coverage>
                  <cdf:location>LocationA</cdf:location>
                  <cdf:prev-imp>no</cdf:prev-imp>
                  <cdf:project-status>different</cdf:project-status>
                  <cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>
                  <cdf:complementary_proposal>no</cdf:complementary_proposal>
        </cdf:projects>
##### Project Offer 3
        <cdf:projects rdf:about="&cdf_ex;P3">
                  <cdf:project-name>Ithekahuno Secondary School - Laboratory Construction
                  </cdf:project-name>
                  <cdf:imp-cost rdf:datatype="&xsd;integer">600000</cdf:imp-cost>
                  <cdf:op-cost rdf:datatype="&xsd;integer">30000</cdf:op-cost>
                  <cdf:target-size rdf:datatype="&xsd;integer">3000</cdf:target-size>
                  <cdf:project-area>education</cdf:project-area>
                  <cdf:geo-coverage rdf:datatype="&xsd;integer">500</cdf:geo-coverage>
                  <cdf:location>LocationA</cdf:location>
                  <cdf:prev-imp>yes</cdf:prev-imp>
                  <cdf:project-status>on-going</cdf:project-status>
                  <cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>
                  <cdf:complementary_proposal>no</cdf:complementary_proposal>
        </cdf:projects>
```

Project Offer 4

<cdf:projects rdf:about="&cdf_ex;P4">

<cdf:project-name>Mutathi-ini secondary school - Construction of two classrooms and Laboratory

</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">1300000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">50000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">4000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">800</cdf:geo-coverage>

<cdf:location>LocationB</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 5

<cdf:projects rdf:about="&cdf_ex;P5">

<cdf:project-name>Unjiru Self Help Dispensary - Construction and equipping of

maternity wing and general ward</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">1000000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">200000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">10000</cdf:target-size>

<cdf:project-area>Health</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">1000</cdf:geo-coverage>

<cdf:location>LocationC</cdf:location>

<cdf:prev-imp>yes</cdf:prev-imp>

<cdf:project-status>on-going</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 6

<cdf:projects rdf:about="&cdf_ex;P6">

<cdf:project-name>Constituency Development Fund office –

Construction and equipping</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">400000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">30000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">4000</cdf:target-size>

<cdf:project-area>monitoring_evaluation</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationC</cdf:location>

<cdf:prev-imp>yes</cdf:prev-imp>

<cdf:project-status>on-going</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 7

<cdf:projects rdf:about="&cdf_ex;P7">

<cdf:project-name>Giakanja Police Post - Construction of police lines</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">200000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">100000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">10000</cdf:target-size>

<cdf:project-area>human_rights</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationC</cdf:location>

<cdf:prev-imp>yes</cdf:prev-imp>

<cdf:project-status>on-going</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 8

<cdf:projects rdf:about="&cdf_ex;P8">

<cdf:project-name>Gaaki Chiefs camp - Rehabillitation</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">300000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">30000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">5000</cdf:target-size>

<cdf:project-area>Administrative</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationB</cdf:location>

<cdf:prev-imp>yes</cdf:prev-imp>

<cdf:project-status>on-going</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 9

<cdf:projects rdf:about="&cdf_ex;P9">

<cdf:project-name>Gaithuri Primary School - Rehabillitation of classrooms

and Latrines</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">200000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">20000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">3000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">1000</cdf:geo-coverage>

<cdf:location>LocationF</cdf:location>

<cdf:prev-imp>yes</cdf:prev-imp>

<cdf:project-status>on-going</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 10

<cdf:projects rdf:about="&cdf_ex;P10">

<cdf:project-name>Karo Primary School - Rehabillitation of Classrooms

</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">300000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">30000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">5000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">1000</cdf:geo-coverage>

<cdf:location>LocationA</cdf:location>

<cdf:prev-imp>yes</cdf:prev-imp>

<cdf:project-status>on-going</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 11

<cdf:projects rdf:about="&cdf_ex;P11">

<cdf:project-name>Gaaki Secondary School - Construction of Laboratory

</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">500000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">50000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">10000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationB</cdf:location>

<cdf:prev-imp>yes</cdf:prev-imp>

<cdf:project-status>on-going</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 12

<cdf:projects rdf:about="&cdf_ex;P12">

<cdf:project-name>Muthinga Police Post - Construction and Equipping of Police Lines</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">250000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">20000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">5000</cdf:target-size>

<cdf:project-area>administrative</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationB</cdf:location>

<cdf:prev-imp>yes</cdf:prev-imp>

<cdf:project-status>on-going</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 13

<cdf:projects rdf:about="&cdf_ex;P13">

<cdf:project-name>Gathuthi primary school - Rennovation of classrooms

</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">600000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">30000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">5000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationD</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 14

<cdf:projects rdf:about="&cdf_ex;P14">

<cdf:project-name>Hubu-ini dispensary - Construction and installation

of equipment</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">1300000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">30000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">5000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationD</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 15

<cdf:projects rdf:about="&cdf_ex;P15">

<cdf:project-name>Ndugamano police post - Rennovation work</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">400000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">20000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">3000</cdf:target-size>

<cdf:project-area>administrative</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">800</cdf:geo-coverage>

<cdf:location>LocationD</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 16

<cdf:projects rdf:about="&cdf_ex;P16">

<cdf:project-name>Miagayu-ini primary school - Rennovation of classrooms

</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">600000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">20000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">3000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">1000</cdf:geo-coverage>

<cdf:location>LocationE</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 17

<cdf:projects rdf:about="&cdf_ex;P17">

<cdf:project-name>Zaina dispensary - Construction and equiping of dispensary</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">1300000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">50000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">5000</cdf:target-size>

<cdf:project-area>Health</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">1000</cdf:geo-coverage>

<cdf:location>LocationE</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 18

<cdf:projects rdf:about="&cdf_ex;P18">

<cdf:project-name>Kihingo primary school - Construction of two new classrooms</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">600000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">10000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">3000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationE</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 19

<cdf:projects rdf:about="&cdf_ex;P19">

<cdf:project-name>Kigogo-ini primary sch - Rehabillitation of classrooms

</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">600000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">20000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">3000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationF</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 20

<cdf:projects rdf:about="&cdf_ex;P20">

<cdf:project-name>Githaka primary school - Rennovation of classrooms

</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">1000000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">10000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">4000</cdf:target-size>

<cdf:project-area>education</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">700</cdf:geo-coverage>

<cdf:location>LocationF</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 21

<cdf:projects rdf:about="&cdf_ex;P21">

<cdf:project-name>Tetu Chiefs camp - Rennovation works</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">500000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">10000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">5000</cdf:target-size>

<cdf:project-area>administrative</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">700</cdf:geo-coverage>

<cdf:location>LocationF</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

Project Offer 22

<cdf:projects rdf:about="&cdf_ex;P22">

<cdf:project-name>Kiandu police station(Reallocation)- Construction and equipping of a police post</cdf:project-name>

<cdf:imp-cost rdf:datatype="&xsd;integer">1000000</cdf:imp-cost>

<cdf:op-cost rdf:datatype="&xsd;integer">20000</cdf:op-cost>

<cdf:target-size rdf:datatype="&xsd;integer">10000</cdf:target-size>

<cdf:project-area>administrative</cdf:project-area>

<cdf:geo-coverage rdf:datatype="&xsd;integer">2000</cdf:geo-coverage>

<cdf:location>LocationC</cdf:location>

<cdf:prev-imp>no</cdf:prev-imp>

<cdf:project-status>different</cdf:project-status>

<cdf:presence_of_similar_initiatives>no</cdf:presence_of_similar_initiatives>

<cdf:complementary_proposal>no</cdf:complementary_proposal>

</cdf:projects>

```
</rdf:RDF>
```

rojects Rule File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rulebase SYSTEM "http://10.2.21.44/cdf_projects/dr-device.dtd"

    <!ENTITY cdf "http://10.2.21.44/cdf_projects/cdf.rdf#">
    <!ENTITY cdf_rb "http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#">

<rulebase direction="forward" rdf_export="export-cdf.rdf" rdf_export_classes="cheapest passed_criteria"
rdf_import="http://10.2.21.44/cdf_projects/2007_2008/008_projects.rdf" xmlns:cdf="&cdf;" xmlns:cdf_rb="&cdf_rb;">
<_rbaselab>
    <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#" type="defeasible">cdf-rules</ind>
</_rbaselab>
<competing_rules c_rules="r15 r16 r17">
    <_crlab><ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#cr1">cr1</ind></_crlab>
</competing_rules>
<imp>
    <_rlab ruleID="r1" ruletype="defeasiblerule">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r1">r1</ind>
    </_rlab>
    <_head>
        <atom>
            <_opr><rel>acceptable</rel></_opr>
            <_slot name="projects"><var>x</var></_slot>
        </atom>
    </_head>
    <_body>
        <atom>
            <_opr><rel href="cdf:projects"></rel></_opr>
            <_slot name="cdf:project-name"><var>x</var></_slot>
        </atom>
```

```
</_body>

<_rlab ruleID="r2" ruletype="defeasiblerule" superior="r1">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r2">r2</ind>
</_rlab>
<_head>
        <neg><atom>
                <_opr><rel>acceptable</rel></_opr>
                <_slot name="projects"><var>x</var></_slot>
        </neg></atom>
</_head>
<_body>
        <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:allocation"><var>z</var></_slot>
                <_slot name="cdf:imp-cost">
                        <_and>
                                <var>y</var>
                                <function_call name=">">
                                        <var>y</var>
                                        <function_call name="/">
                                                <var>z</var><ind>2</ind>
                                        </function_call>
                                </function_call>
                        </_and>
                </_slot>
        </atom>
</_body>
```

```
<_rlab ruleID="r3" ruletype="defeasiblerule" superior="r1">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r3">r3</ind>
</_rlab>
<_head>
        <neg>
                <atom>
                        <_opr><rel>acceptable</rel></_opr>
                        <_slot name="projects"><var>x</var></_slot>
                </atom>
        </neg>
</_head>
<_body>
        <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:project-status"><ind>"different"</ind></_slot>
                <_slot name="cdf:prev-imp"><ind>"yes"</ind></_slot>
        </atom>
</_body>
</p>
<p>

<_rlab ruleID="r4" ruletype="defeasiblerule" superior="r1">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r4">r4</ind>
</_rlab>
<_head>
        <neg>
                <atom>
                        <_opr><rel>acceptable</rel></_opr>
                        <_slot name="projects"><var>x</var></_slot>
                </atom>
        </neg>
</_head>
<_body>
```

```
        <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:op-cost"><var>y</var></_slot>
                <_slot name="cdf:imp-cost">
                        <_and>
                                <var>z</var>
                                <function_call name=">">
                                        <var>y</var>
                                        <function_call name="*">
                                                <var>z</var><ind>0.3</ind>
                                        </function_call>
                                </function_call>
                        </_and>
                </_slot>
        </atom>
    </_body>
</imp>
<imp>
    <_rlab ruleID="find_acceptable" ruletype="defeasiblerule" superior="r1">
    <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#find_acceptable">find_acceptable</ind>
    </_rlab>
    <_head>
        <calc>
                <function_call name="bind">
                        <var>p</var>
                        <function_call name="round">
                                <function_call name="*">
                                        <ind>100</ind>
                                        <function_call name="/">
                                                <var>m</var><var>d</var>
                                        </function_call>
                                </function_call>
```

```
                    </function_call>
                </function_call>
            </calc>
            <calc>
                <function_call name="bind">
                    <var>c</var>
                    <function_call name="-">
                        <var>m</var><ind>0</ind>
                    </function_call>
                </function_call>
            </calc>
            <calc>
                <function_call name="bind">
                    <var>t</var>
                    <function_call name="-">
                        <var>w</var><ind>0</ind>
                    </function_call>
                </function_call>
            </calc>
            <atom>
                <_opr><rel href="passed_criteria"></rel></_opr>
                <_slot name="projects"><var>x</var></_slot>
                <_slot name="percentage_allocation"><var>p</var></_slot>
                <_slot name="cost"><var>c</var></_slot>
                <_slot name="target"><var>t</var></_slot>
            </atom>
    </_head>
    <_body>
            <and>
                <atom>
                    <_opr><rel>acceptable</rel></_opr>
                    <_slot name="projects"><var>x</var></_slot>
                </atom>
```

```
                <atom>
                        <_opr><rel href="cdf:projects"></rel></_opr>
                        <_slot name="cdf:project-name"><var>x</var></_slot>
                        <_slot name="cdf:imp-cost"><var>m</var></_slot>
                        <_slot name="cdf:target-size"><var>w</var></_slot>
                        <_slot name="cdf:allocation"><var>d</var></_slot>
                </atom>
        </and>
</_body>
imp>
mp>

<_rlab ruleID="r6" ruletype="defeasiblerule" superior="r1">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r6">r6</ind>
</_rlab>
<_head>
        <neg>
                <atom>
                        <_opr><rel>acceptable</rel></_opr>
                        <_slot name="projects"><var>x</var></_slot>
                </atom>
        </neg>
</_head>
<_body>
        <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:target-size">
                        <_and>
                                <var>z</var>
                                <function_call name="&lt;">
                                        <var>z</var><ind>5000</ind>
                                </function_call>
                        </_and>
```

```
                </_slot>
        </atom>
</_body>
imp>
imp>
<_rlab ruleID="r7" ruletype="defeasiblerule" superior="r1">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r7">r7</ind>
</_rlab>
<_head>
        <neg>
                <atom>
                        <_opr><rel>acceptable</rel></_opr>
                        <_slot name="projects"><var>x</var></_slot>
                </atom>
        </neg>
</_head>
<_body>
        <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:presence_of_similar_initiatives"><ind>"yes"</ind></_slot>
                <_slot name="cdf:complementary_proposal"><ind>"no"</ind></_slot>
        </atom>
</_body>
imp>
mp>
<_rlab ruleID="r8" ruletype="defeasiblerule" superior="r1">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r8">r8</ind>
</_rlab>
<_head>
        <neg>
                <atom>
```

```
                    <_opr><rel>acceptable</rel></_opr>
                    <_slot name="projects"><var>x</var></_slot>
                </atom>
            </neg>
        </_head>
        <_body>
            <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:project-area"><ind>"education"</ind></_slot>
                <_slot name="cdf:allocation"><var>z</var></_slot>
                <_slot name="cdf:imp-cost">
                    <_and>
                        <var>y</var>
                        <function_call name=">">
                            <var>y</var>
                            <function_call name="*">
                                <var>z</var>
                                <ind>0.15</ind>
                            </function_call>
                        </function_call>
                    </_and>
                </_slot>
            </atom>
        </_body>
</imp>

<imp>
    <_rlab ruleID="r9" ruletype="defeasiblerule" superior="r1">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r9">r9</ind>
    </_rlab>
    <_head>
        <neg>
```

```
                <atom>
                        <_opr><rel>acceptable</rel></_opr>
                        <_slot name="projects"><var>x</var></_slot>
                </atom>
        </neg>
</_head>
<_body>
        <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:project-area"><ind>"sports"</ind></_slot>
                <_slot name="cdf:allocation"><var>z</var></_slot>
                <_slot name="cdf:imp-cost">
                        <_and>
                                <var>y</var>
                                <function_call name=">">
                                        <var>y</var>
                                        <function_call name="*">
                                                <var>z</var><ind>0.2</ind>
                                        </function_call>
                                </function_call>
                        </_and>
                </_slot>
        </atom>
</_body>
</imp>

<imp>
<_rlab ruleID="r10" ruletype="defeasiblerule" superior="r1">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r10">r10</ind>
</_rlab>
<_head>
        <neg>
```

```
                <atom>
                    <_opr><rel>acceptable</rel></_opr>
                    <_slot name="projects"><var>x</var></_slot>
                </atom>
            </neg>
        </_head>
        <_body>
            <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:project-area"><ind>"monitoring_evaluation"</ind></_slot>
                <_slot name="cdf:allocation"><var>z</var></_slot>
                <_slot name="cdf:imp-cost">
                    <_and>
                        <var>y</var>
                        <function_call name=">">
                            <var>y</var>
                            <function_call name="*">
                                <var>z</var><ind>0.2</ind>
                            </function_call>
                        </function_call>
                    </_and>
                </_slot>
            </atom>
        </_body>
    </imp>
    <imp>
        <_rlab ruleID="r11" ruletype="defeasiblerule" superior="r1">
            <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r11">r11</ind>
        </_rlab>
        <_head>
            <neg><atom>
                <_opr><rel>acceptable</rel></_opr>
```

```
                <_slot name="projects"><var>x</var></_slot>
            </atom></neg>
        </_head>
        <_body>
            <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:project-area"><ind>"environmental"</ind></_slot>
                <_slot name="cdf:allocation"><var>z</var></_slot>
                <_slot name="cdf:imp-cost">
                    <_and>
                        <var>y</var>
                        <function_call name=">">
                            <var>y</var>
                            <function_call name="*">
                                <var>z</var><ind>0.2</ind>
                            </function_call>
                        </function_call>
                    </_and>
                </_slot>
            </atom>
        </_body>
</imp>
<imp>
        <_rlab ruleID="r12" ruletype="defeasiblerule" superior="r1">
            <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r12">r12</ind>
        </_rlab>
        <_head>

            <calc>
                <function_call name="bind">
                    <var>t</var>
                    <function_call name="-">
```

```
                    <var>z</var><ind>0</ind>
                </function_call>
            </function_call>
        </calc>
        <atom>
            <_opr><rel>over_7000_target_offer</rel></_opr>
            <_slot name="projects"><var>x</var></_slot>
            <_slot name="target"><var>t</var></_slot>
        </atom>
    </_head>
    <_body>
        <and>
            <atom>
                <_opr><rel href="passed_criteria"></rel></_opr>
                <_slot name="projects"><var>x</var></_slot>
            </atom>
            <atom>
                <_opr><rel href="cdf:projects"></rel></_opr>
                <_slot name="cdf:project-name"><var>x</var></_slot>
                <_slot name="cdf:allocation"><var>y</var></_slot>
                <_slot name="cdf:target-size">
                    <_and>
                        <var>z</var>
                        <function_call name=">">
                            <var>z</var><ind>7000</ind>
                        </function_call>
                    </_and>
                </_slot>
            </atom>
        </and>
    </_body>
</imp>
```

```
<_rlab ruleID="r13" ruletype="defeasiblerule" superior="r12">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r13">r13</ind>
</_rlab>
<_head>
        <calc>
                <function_call name="bind">
                        <var>a</var>
                        <function_call name="round">
                                <function_call name="+">
                                        <function_call name="*">
                                                <var>y</var><ind>0.25</ind>
                                        </function_call>
                                        <function_call name="*">
                                                <ind>200000</ind>
                                                <function_call name="/">
                                                        <ind>100</ind>
                                                        <function_call name="-">
                                                                <var>z</var><ind>7000</ind>
                                                        </function_call>
                                                </function_call>
                                        </function_call>
                                </function_call>
                        </function_call>
                </function_call>
        </calc>

        <atom>
                <_opr><rel>poverty_reduction_offer</rel></_opr>
                <_slot name="projects"><var>x</var></_slot>
                <_slot name="amount"><var>a</var></_slot>
        </atom>
</_head>
```

```
<_body>
        <and>
                <atom>
                        <_opr><rel href="cdf:projects"></rel></_opr>
                        <_slot name="cdf:project-name"><var>x</var></_slot>
                        <_slot name="cdf:project-area"><ind>"poverty_reduction"</ind></_slot>
                        <_slot name="cdf:allocation"><var>y</var></_slot>
                        <_slot name="cdf:target-size">
                                <_and>
                                        <var>z</var>
                                        <function_call name=">">
                                                <var>z</var><ind>7000</ind>
                                        </function_call>
                                </_and>
                        </_slot>
                </atom>
                <atom>
                        <_opr><rel href="passed_criteria"></rel></_opr>
                        <_slot name="projects"><var>x</var></_slot>
                </atom>
        </and>
</_body>
</imp>

<imp>
<_rlab ruleID="find_cheapest" ruletype="defeasiblerule">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#find_cheapest">find_cheapest</ind>
</_rlab>
<_head>
        <calc>
                <function_call name="bind">
                        <var>b</var>
                        <function_call name="-">
```

```
                        <var>z</var><ind>0</ind>
                    </function_call>
                </function_call>
        </calc>
        <atom>
                <_opr><rel href="cheapest"></rel></_opr>
                <_slot name="projects"><var>x</var></_slot>
                <_slot name="cost"><var>b</var></_slot>
        </atom>
</_head>
<_body>
        <and>
                <atom>
                        <_opr><rel href="passed_criteria"></rel></_opr>
                        <_slot name="projects"><var>x</var></_slot>
                </atom>
                <atom>
                        <_opr><rel href="cdf:projects"></rel></_opr>
                        <_slot name="cdf:project-name"><var>x</var></_slot>
                        <_slot name="cdf:imp-cost"><var>z</var></_slot>
                </atom>
                <naf>
                        <and>
                                <atom>
                                        <_opr><rel>acceptable</rel></_opr>
                                        <_slot name="projects">
                                                <_and>
                                                        <var>y</var>
                                                        <_not><var>x</var></_not>
                                                </_and>
                                        </_slot>
                                </atom>
                                <atom>
```

```
                                        <_opr><rel href="cdf:projects"></rel></_opr>
                                <_slot name="cdf:project-name"><var>y</var></_slot>
                                        <_slot name="cdf:imp-cost">
                                                <_and>
                                                        <var>w</var>
<function_call name="&lt;">
                                                        <var>w</var><var>z</var>
                                                        </function_call>
                                                </_and>
                                        </_slot>
                                </atom>
                        </and>
                </naf>
        </and>
    </_body>
</imp>

<imp>
<_rlab ruleID="find_largest_target" ruletype="defeasiblerule">
    <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#find_largest_target">find_largest_target</ind>
</_rlab>
    <_head>
        <calc>
            <function_call name="bind">
                <var>t</var>
                <function_call name="-">
                    <var>z</var><ind>0</ind>
                </function_call>
            </function_call>
        </calc>
        <calc>
            <function_call name="bind">
                <var>c</var>
```

```
                    <function_call name="-">
                        <var>m</var><ind>0</ind>
                    </function_call>
                </function_call>
        </calc>
        <atom>
                <_opr><rel href="largesttarget"></rel></_opr>
                <_slot name="projects"><var>x</var></_slot>
                <_slot name="target"><var>t</var></_slot>
                <_slot name="cost"><var>c</var></_slot>
        </atom>
</_head>
<_body>
        <and>
                <atom>
                        <_opr><rel href="passed_criteria"></rel></_opr>
                        <_slot name="projects"><var>x</var></_slot>
                </atom>
                <atom>
                        <_opr><rel href="cdf:projects"></rel></_opr>
                        <_slot name="cdf:project-name"><var>x</var></_slot>
                        <_slot name="cdf:imp-cost"><var>m</var></_slot>
                        <_slot name="cdf:target-size"><var>z</var></_slot>
                </atom>
                <naf>
                        <and>
                                <atom>
                                        <_opr><rel>acceptable</rel></_opr>
                                        <_slot name="projects">
                                                <_and>
                                                        <var>y</var>
                                                        <_not><var>x</var></_not>
                                                </_and>
```

```
                    </_slot>
                </atom>
                <atom>
                    <_opr><rel href="cdf:projects"></rel></_opr>
                    <_slot name="cdf:project-name"><var>y</var></_slot>
                    <_slot name="cdf:target-size">
                        <_and>
                            <var>w</var>
                            <function_call name=">">
                                <var>w</var><var>z</var>
                            </function_call>
                        </_and>
                    </_slot>
                </atom>
            </and>
        </naf>
    </and>
</_body>
</imp>

<imp>
<_rlab ruleID="r15" ruletype="defeasiblerule">
    <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r15">r15</ind>
</_rlab>
<_head>
    <atom>
        <_opr><rel>implement</rel></_opr>
        <_slot name="projects"><var>x</var></_slot>
    </atom>
</_head>
<_body>
    <and>
        <atom>
```

```
                    <_opr><rel href="passed_criteria"></rel></_opr>
                    <_slot name="projects"><var>x</var></_slot>
                </atom>
                <atom>
                    <_opr><rel href="cdf:projects"></rel></_opr>
                    <_slot name="cdf:project-name"><var>x</var></_slot>
                </atom>
            </and>
        </_body>
</imp>

<imp>
        <_rlab ruleID="r16" ruletype="defeasiblerule" superior="r15">
            <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r16">r16</ind>
        </_rlab>
        <_head>
            <atom>
                <_opr><rel>implement</rel></_opr>
                <_slot name="projects"><var>x</var></_slot>
            </atom>
        </_head>
        <_body>
            <and>
                <atom>
                    <_opr><rel href="cdf:projects"></rel></_opr>
                    <_slot name="cdf:project-name"><var>x</var></_slot>
                    <_slot name="cdf:project-area"><ind>"poverty_reduction"</ind></_slot>
                </atom>
                <atom>
                    <_opr><rel href="passed_criteria"></rel></_opr>
                    <_slot name="projects"><var>x</var></_slot>
                </atom>
            </and>
```

```
        </_body>
</imp>
<imp>
        <_rlab ruleID="r17" ruletype="defeasiblerule" superior="r15 r16">
                <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#r17">r17</ind>
        </_rlab>
        <_head>
                <atom>
                        <_opr><rel>implement</rel></_opr>
                        <_slot name="projects"><var>x</var></_slot>
                </atom>
        </_head>
        <_body>
                <and>
                        <atom>
                                <_opr><rel href="cdf:projects"></rel></_opr>
                                <_slot name="cdf:project-name"><var>x</var></_slot>
                                <_slot name="cdf:project-area"><ind>"poverty_reduction"</ind></_slot>
                        </atom>
                        <atom>
                                <_opr><rel>cheapest</rel></_opr>
                                <_slot name="projects"><var>x</var></_slot>
                        </atom>
                </and>
        </_body>
</imp>
<imp>
        <_rlab ruleID="find_implementable" ruletype="defeasiblerule" superior="r1">
        <ind href="http://10.2.21.44/cdf_projects/cdf-rbase.ruleml#find_implementable">find_implementable</ind>
        </_rlab>
        <_head>
                <calc>
                        <function_call name="bind">
```

```
                    <var>c</var>
                    <function_call name="-">
                        <var>m</var><ind>0</ind>
                    </function_call>
                </function_call>
            </calc>
            <atom>
                <_opr><rel href="passed_implement"></rel></_opr>
                <_slot name="projects"><var>x</var></_slot>
                <_slot name="cost"><var>c</var></_slot>
            </atom>
        </_head>
        <_body>
            <and>
                <atom>
                    <_opr><rel>implement</rel></_opr>
                    <_slot name="projects"><var>x</var></_slot>
                </atom>
                <atom>
                    <_opr><rel href="cdf:projects"></rel></_opr>
                    <_slot name="cdf:project-name"><var>x</var></_slot>
                    <_slot name="cdf:imp-cost"><var>m</var></_slot>
                </atom>
            </and>
        </_body>
</imp>
</rulebase>
```