

SENG 697 - Agent-based Software Engineering  
Development Project

DCNet: An Agent Approach to Securing and Optimizing Oil and Gas Rig  
Resources

J. Mark Dochstader  
ID: 00318459  
For: Dr. Behrouz Far  
Due: 31 October 2006

## Abstract

Western Canada has seen record growth in the demand for drilling rigs capable of pursuing oil and gas properties. Securing these 3<sup>rd</sup> party resources however, is complicated by individual organizational priorities, environmental concerns and market forces. This paper describes the design of The Drilling Contractor Network (DCNet), a system intended to simplify the act of securing drilling rigs and allow contracting organizations to work together in optimizing the value extracted from these resources. DCNet uses an agent approach based on the Prometheus design methodology and is designed for use by drilling contractors and petroleum development companies.

## Introduction

The increasing volume of petroleum exploration activities in Western Canada has led to skyrocketing demand for drilling resources capable of recovering oil and gas. These activities are characterized by a number of exasperating factors:

- Targeted reserves are relatively well defined when compared with newer global fields. This means a large volume of proposed wells are infill operations of short duration - schedule optimization can have huge benefits.
- Seasonal weather restricts drilling to specific time periods due to accessibility and environmental impact.
- Potential drilling activities are defined by market factors, leading to specific organizational priorities.
- Drilling operations are generally undertaken by 3<sup>rd</sup> party drilling contractors who own the drilling rig but do not have an ownership stake in the target property. This can lead to conflicting goals between the contracting developer and the rig contractor.

The Drilling Contractor Network (DCNet) is proposed as an agent-based solution to these industry issues, with two specific goals:

1. Create an efficient mechanism for organizations to secure drilling resources based on complex demands, schedules and priorities.
2. Enable organizations to collaborate in maximizing their return on secured drilling resources by sharing relevant information.

This paper outlines the design and specification of DCNet using the Prometheus methodology for developing intelligent agents and includes the artifacts created by this agent-based software engineering approach. The intent is to produce a set of documentation that could be utilized to implement the DCNet system.

## System Overview

DCNet represents two main industry-based entities: Drilling Contractors and Oil and Gas (O&G) Developers. Each encompasses a set of independent attributes and goals.

Drilling Contractors own the physical drilling rig and personnel which they make available on a contract basis. These assets have a high capital cost, making utilization maximization the overall goal. Sub goals such as minimizing move time between clients and using the correct rig type based on the target job also play a key role in achieving maximum utility.

O&G Developers view the drilling activity as a pure cost in retrieval of petroleum products, thus the desired outcome is to minimize rig usage, diametrically positioned to the goal of Drilling Contractors. Developers may also need to account for some or all of the following factors:

- Drill wells in the order dictated by corporate strategies or regulatory approval (example: sufficient reservoir drainage)
- Minimize inter-job move time while under contract
- Correctly order multiple operations on the same well
- Minimize fixed or flat rate costs (example: move and setup)

The system follows these general steps:

1. Drilling Contractors post availability, rate, location and other relevant attributes describing the rig<sup>1</sup>
2. O&G Developers compare the jobs on their drilling schedules with rig availability. Jobs include these key factors:
  - a. Location
  - b. Duration
  - c. Required Rig Type
  - d. Priority
  - e. Precedence (relies on another job) example: A well must be drilled before it can be completed.
3. Schedules change due to:
  - a. External Events (System Environment) such as weather or global markets
  - b. Internal Actions (Agent) like shifting priorities or complications during job activities
4. Drilling Contractors revise availability of rigs
5. O&G Developers must:
  - a. Secure resources to cover new schedule gaps
  - b. Release resources no longer needed:
    - i. Back to the drilling contractor
    - ii. To other O&G Developers

---

<sup>1</sup>Rigs may be contracted based on a daily, hourly or depth metric but scheduling is always completed on a calendar basis.

6. Rig schedules for Drilling Contractors and O&G Developers are updated and the process repeats.

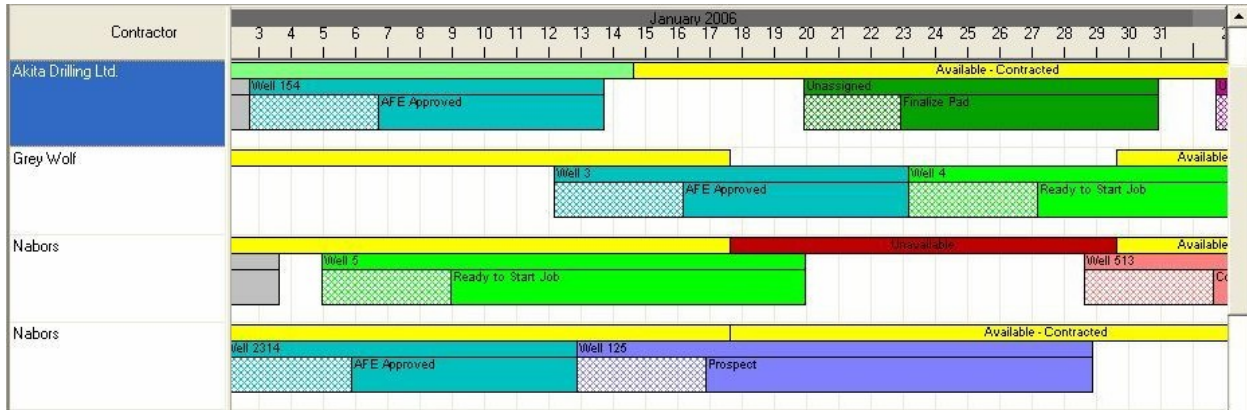


Figure 1 - Rig Schedule Visualization

## System Specification

The Prometheus methodology initiates the agent system design with the overall system specification. This involves defining the agent functionalities from the viewpoint of the entire system using functional descriptors. Later steps in the process will group these descriptors into distinct agent units. *Appendix A - Functional Descriptors* lists the complete definitions for the following system functionalities:

Function ID - Name	Description
001-Determine Rig Availability	Defines how rig availability is determined.
002-Post Rig Availability	Allows a drilling contractor to publish the availability of their drilling resources.
003-Negotiate Rig Contract	Process of reaching agreement for the contracting of a drilling rig.
004-Job Schedule Optimizer	Given a list of proposed jobs, optimization criteria and constraints, determines the target job schedule.
005-Retrieve Rig Availability	Allows an O&G developer to review the published rig availability as posted by drilling contractors.
006-Post Excess Rig Capacity	Allows an O&G developer to post excess contracted drilling resources as available for other developers.
007-Request Collaboration Partnership	Allows an O&G developer to post specific information regarding activities to find other developers looking to cooperate on specific projects.
008-Update Rig Availability and Contracts	Allows a drilling contractor to update their rig availability schedule and inform developers who are impacted by these changes.

The other substantial component of the system specification under the Prometheus methodology is the set of use case scenarios describing the system in action.

Appendix B - Use Case Scenarios includes the illustrations of the system functionality using the suggested Prometheus notation.

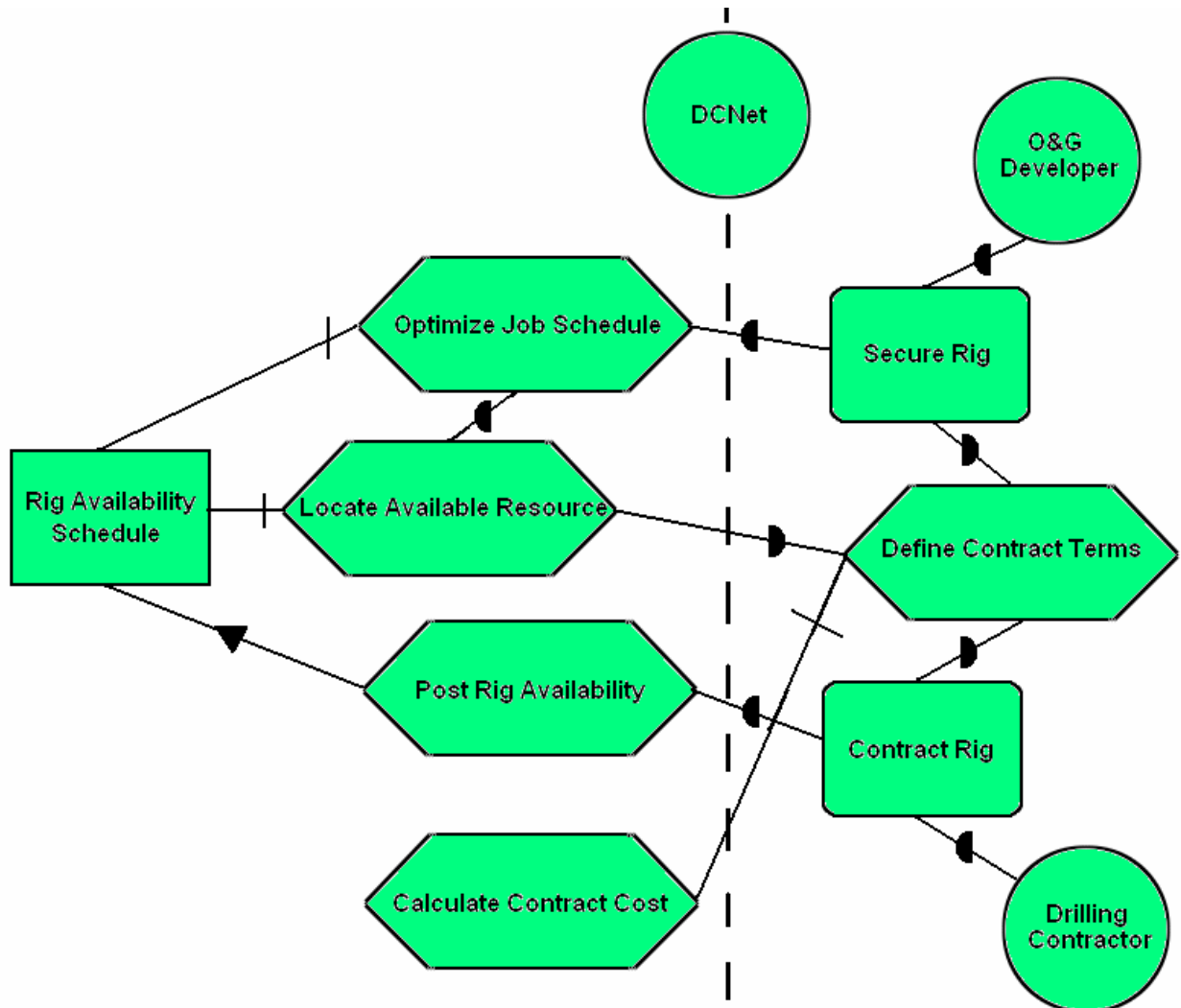


Figure 2 - Use Case Scenario 003 - Negotiate Rig Contract

## Architecture Design

The architecture phase of the Prometheus methodology focuses on determining which agents should exist and how identified functions should be allocated to the target agents. Each agent is represented by an agent descriptor which includes the agent identification, description, cardinality, included functionalities, read/write data and interactions. Shared data sources are also defined and the system overview diagram is created to illustrate relationships between agents, events and shared data sources.

DCNet divides the defined functionality into three agents exhibiting high cohesion and limited coupling: Developer Agent, Contractor Agent and Helper Agent. The descriptors are presented below:

<b>Developer Agent</b>	Submits job schedules, requests & secures rig resources, posts/searches partnership opportunities
Cardinality:	1 per O&G Developer, Instantiated when the user registers with system
Functions:	job schedule submission, rig resource requestor, contract negotiator, excess rig capacity submission
Reads Data:	Excess Capacity Schedule, Partnering Opportunities
Writes Data:	Excess Capacity Schedule, Partnering Opportunities
Interactions:	Contractor Agent, Helper Agent

<b>Contractor Agent</b>	Submits rig availability schedules, completes contract negotiations, updates developers when schedule changes
Cardinality:	1 per Drilling Contractor, Instantiated when the user registers with system
Functions:	Rig availability submission, rig availability updates, schedule change notifications, contract negotiator
Reads Data:	Rig Availability Schedule
Writes Data:	Rig Availability Schedule
Interactions:	Developer Agent, Helper Agent

<b>Helper Agent</b>	
Cardinality:	1 per system, Instantiated on system startup
Functions:	Job Schedule Optimizer, Locate Rig Resources, Cost Calculator
Reads Data:	Rig Availability Schedule, Excess Capacity Schedule
Writes Data:	Rig Availability Schedule, Excess Capacity Schedule
Interactions:	Contractor Agent, Developer Agent

### **Shared Data Sources**

Shared data sources are used to provide the opportunity for collaboration in scheduling rigs as well as partnering opportunities between oil and gas developers.

<b>Rig Availability Schedule</b>	Lists rig resource details and availability
Readers:	Developer Agent, Helper Agent
Writers:	Contractor Agent

<b>Excess Capacity Schedule</b>	Lists excess capacity rig resource that have already been secured but are no longer needed/wanted
Readers:	Developer Agent, Helper Agent
Writers:	Developer Agent

<b>Partnering Opportunities</b>	Lists job information and/or rig resource details and availability for shared development of specific target properties
Readers:	Developer Agent
Writers:	Developer Agent

**System Overview Diagram**

The System Overview diagram combines the agents, events and shared data sources. Prometheus relies very heavily on the system overview to provide a single artifact that effectively communicates the entire general operation of the system under development.

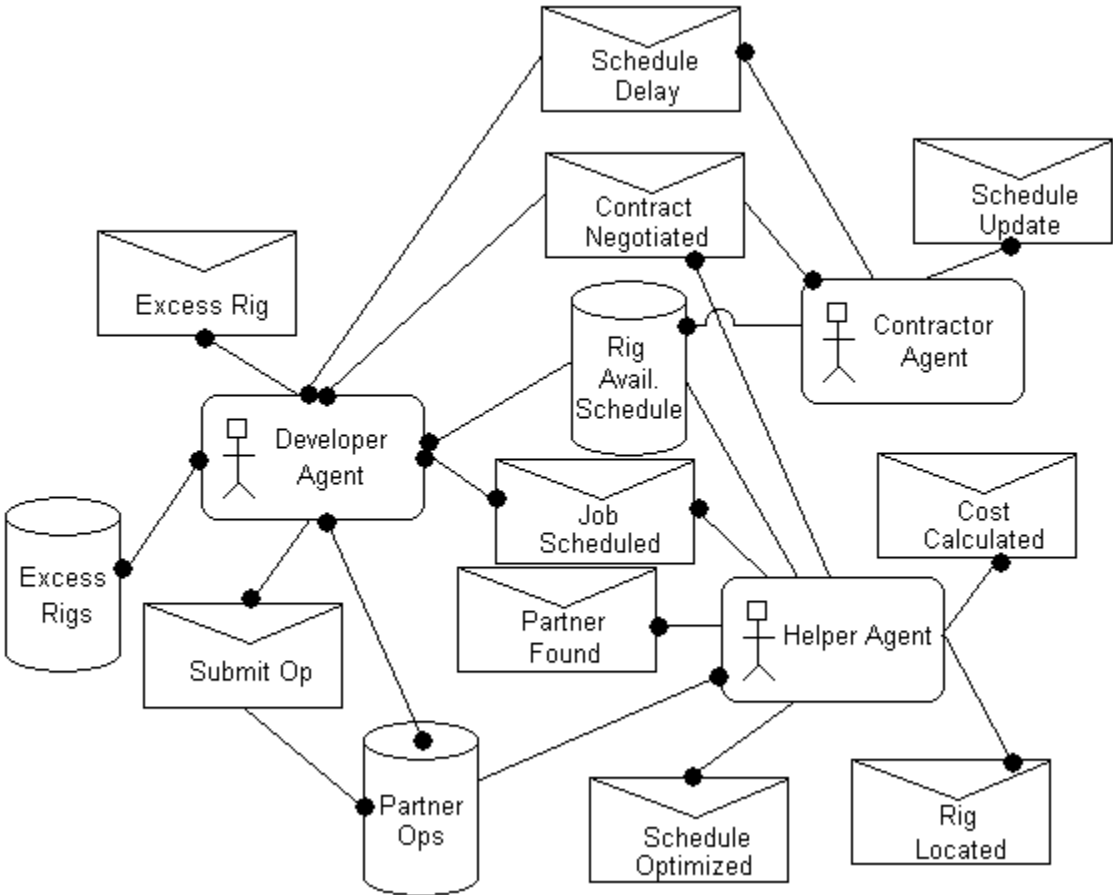


Figure 3 - DCNet System Overview Diagram

## Agent Interactions

Coupled with the System Overview Diagram, the Interaction Diagrams and Protocols provide a flow analysis similar to the previously defined Use Case scenarios. The Prometheus interaction constructs are borrowed directly from object oriented design. The major instances for the DCNet system are presented below:

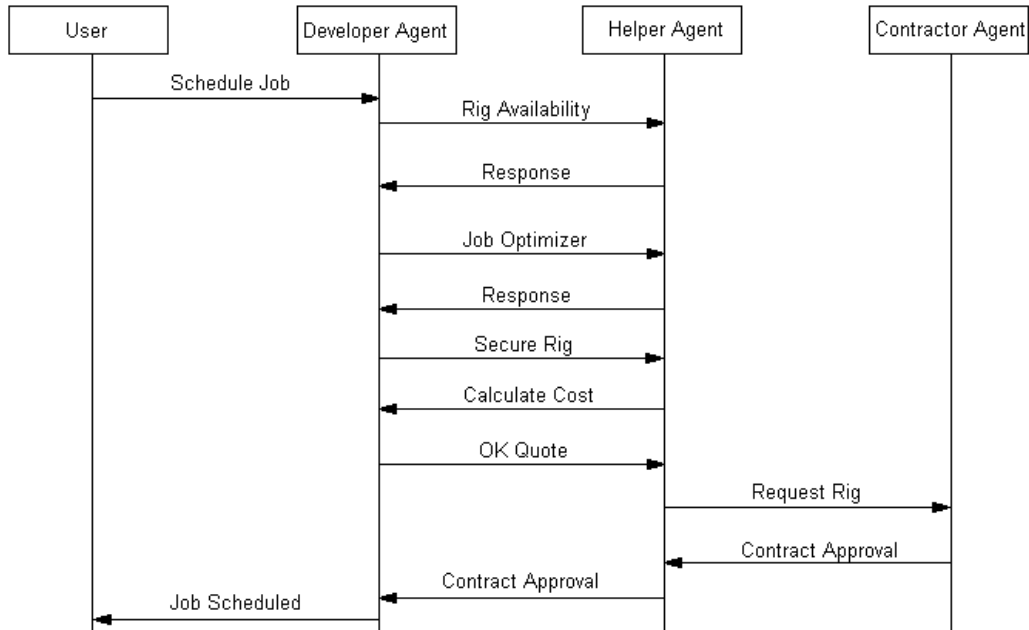


Figure 4 - Interaction Diagram - Schedule Job

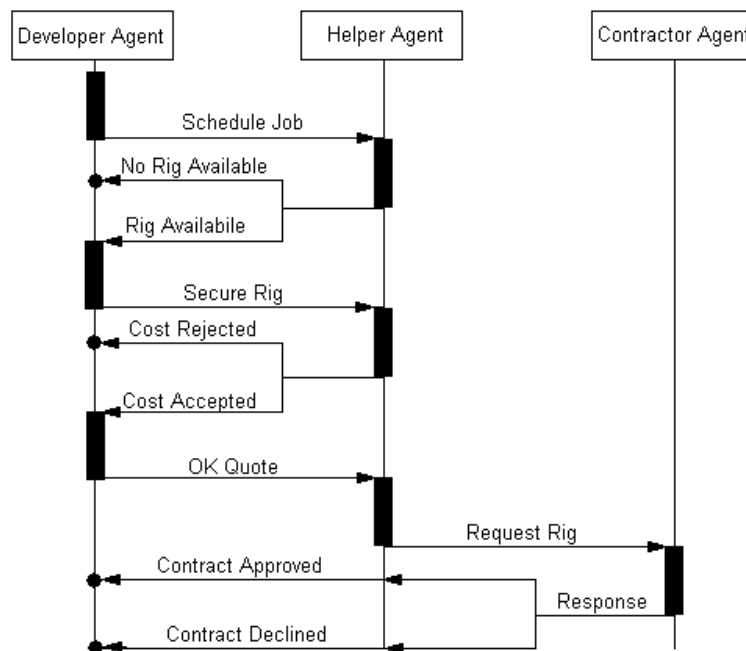


Figure 5 - Interaction Protocol - Schedule Job

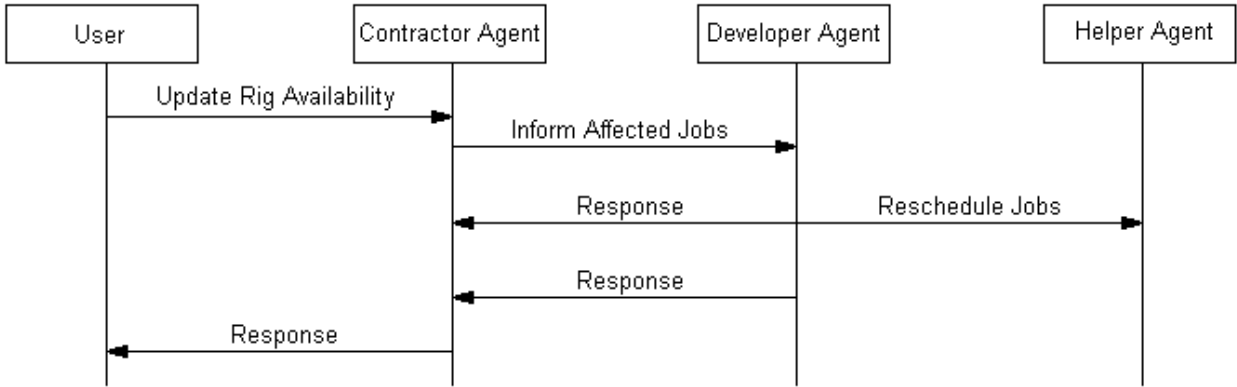


Figure 6 - Interaction Diagram - Update Rig Availability

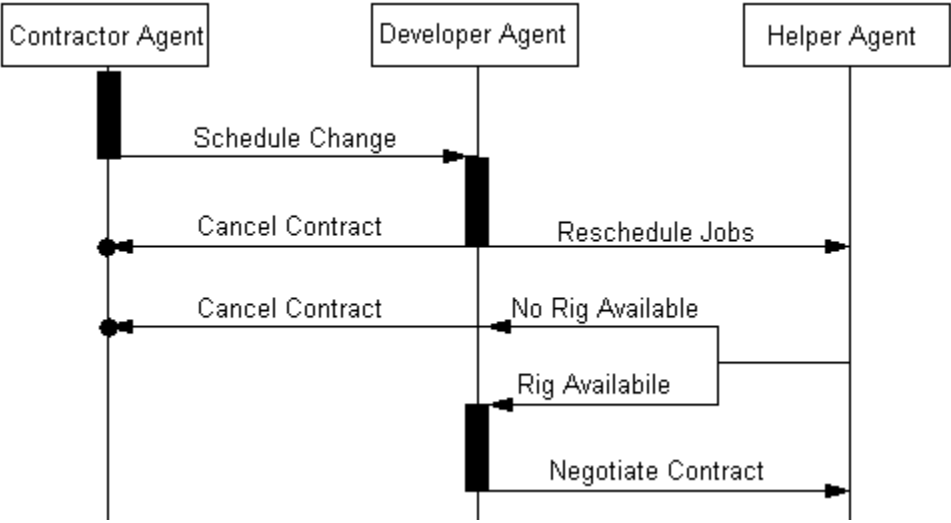


Figure 7 - Interaction Protocol - Update Rig Availability

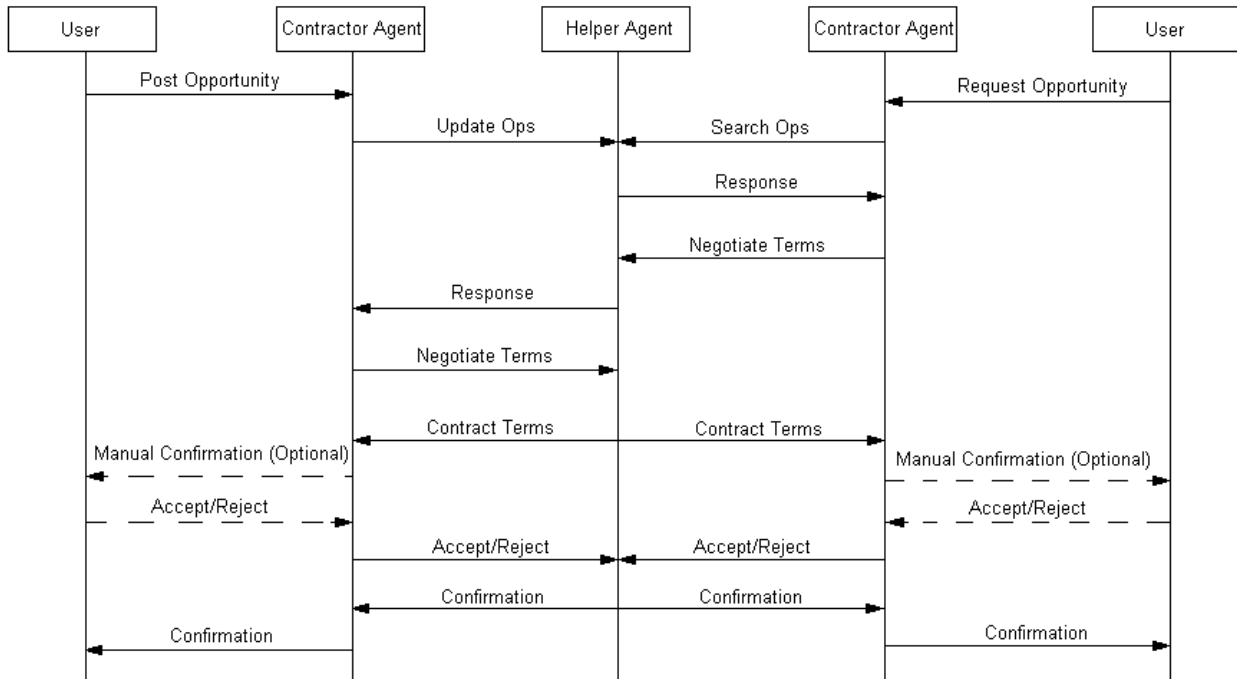


Figure 8 - Interaction Diagram - Collaboration Opportunity

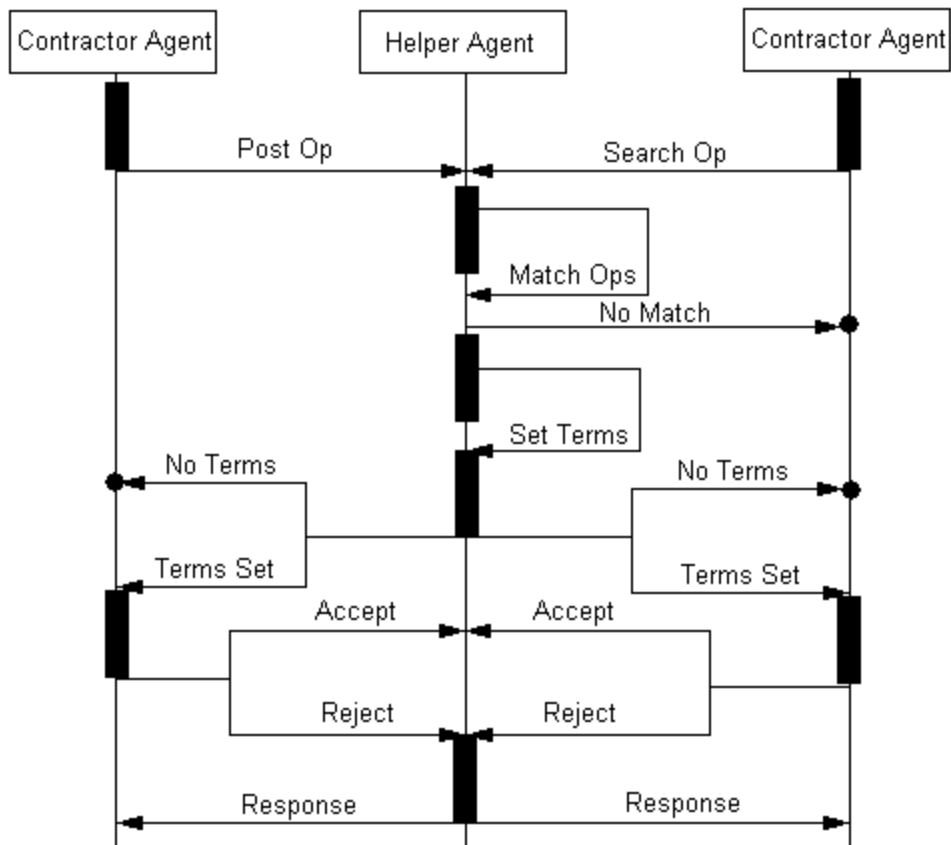


Figure 9 - Interaction Protocol - Collaboration Opportunity

## Detailed Design

The detailed design phases focuses on the internal structure of each agent and the implementation of the functionalities contained. The Prometheus methodology suggests an iterative refinement of the agent’s capabilities, events, plans and detailed data structures. The Agent Overview Diagram is produced to illustrate a top level view of the agent internals, including functions and/or services.

DCNet contains three agents with the overview diagrams presented below:

### Developer Agent

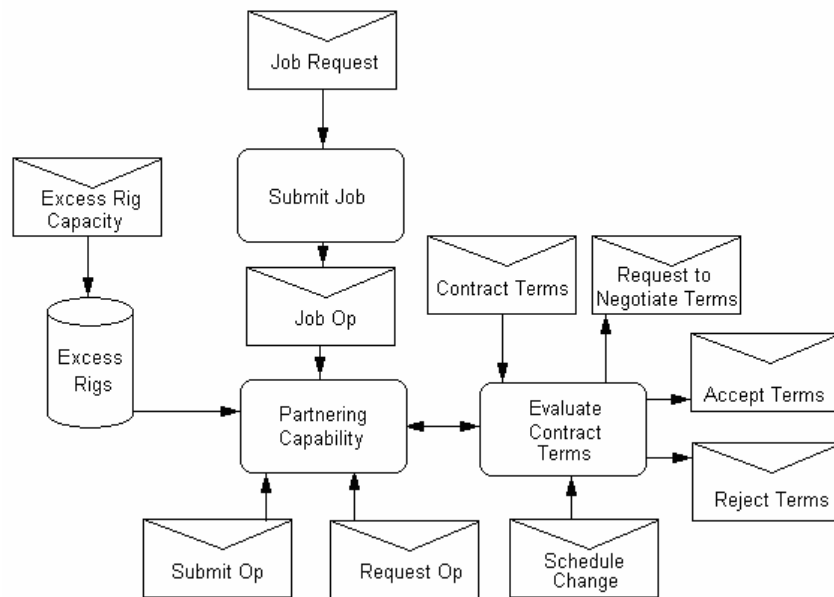


Figure 10 - Agent Overview - Developer Agent

### Contractor Agent

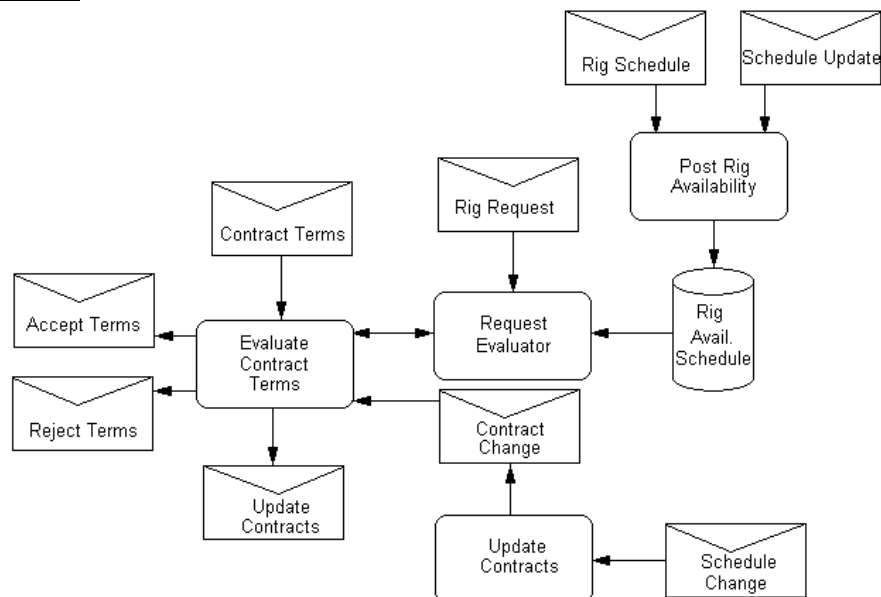


Figure 11 - Agent Overview - Contractor Agent

## Helper Agent

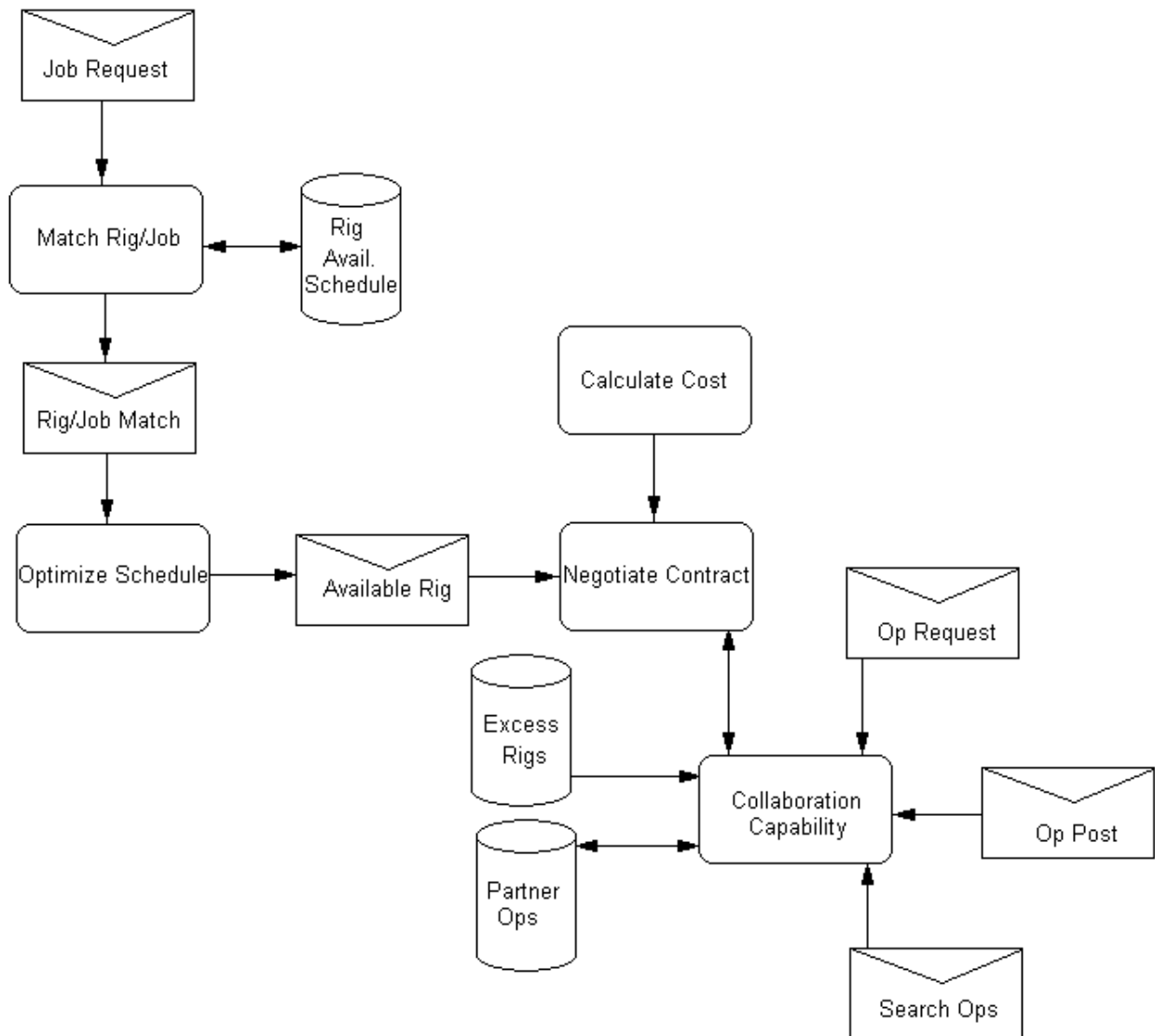


Figure 12 - Agent Overview - Helper Agent

## Data and Knowledge Sharing

DCNet uses a set of shared data sources (defined during the architecture design phase) to share knowledge between agents. A novel aspect of the system is the dual roles agents can exhibit they potentially compete for limited drilling resources and collaborate on drilling projects simultaneously. The judicious sharing of select information allows for this apparent paradox to exist within the system as agents autonomously cooperate or compete as the user's goals warrant.

## **Communication Specification**

DCNet is intended to utilize an established standard messaging protocol such as XML messages delivered via SOAP over HTTP to reflect the distributed nature of the system. The message sequences will adhere to the interaction and protocol diagrams defined during the Architectural Design phase, as per the Prometheus development process. The actual contents and transmission of the messages is left to the actual development of the system so as not to preemptively restrict the implementation utilized.

## **Summary and Further Work**

DCNet describes an agent-based system that proposes to solve two key problems currently experienced in the Western Canadian oil and gas industry:

1. How do exploration and development organizations located and secure scarce drilling resources?
2. Once resources have been secured, How do organizations share selected information in an attempt to maximize their return on investment?

The potential benefits of agent-based sourcing and securing of drilling rigs as illustrated by DCNet would easily be recognized by potential adopters. In terms of cooperation, the petroleum industry is traditionally very competitive and a collaborative approach to improving the returns from investment in drilling operations a new concept. It is however an industry where the financial bottom line greatly influences decisions and DCNet provides a viable framework for controlling and sharing information that can directly impact profitability.

Prometheus appears to be a suitable framework for developing agent systems such as DCNet. A greater abstraction from the typical object-oriented paradigm allowed agents to be defined as independent entities that both compete and cooperate as environmental conditions warrant. The artifacts produced by Prometheus also lend themselves to interpretation and utilization by non-experts in the agent field. Next steps in the development of DCNet would be to take the documents created by the design process and implement the system.

## Appendix A - Functional Descriptors

<b>001-Determine Rig Availability</b>	Defines how rig availability is determined.
Actions:	Compares available resources with matching criteria, returns one of three responses: <ul style="list-style-type: none"> <li>• Unavailable: no resources available</li> <li>• Uncontracted - Available: rig available for contract</li> <li>• Contracted - Available: excess capacity available for sub contract</li> </ul>
Percepts:	Rig Availability Schedule, Excess Capacity Schedule
Data Access:	Geographic location, planned duration, rig type, cost range
Interactions:	Cost Calculation

<b>002-Post Rig Availability</b>	Allows a drilling contractor to publish the availability of their drilling resources.
Actions:	Drilling contractor updates rig schedule, system replies with success or error
Percepts:	Drilling contractor accesses system
Data Access:	Writes Rig Availability Schedule
Interactions:	None

<b>003-Negotiate Rig Contract</b>	Process of reaching agreement for the contracting of a drilling rig.
Actions:	O&G Developer passes job information to optimizer, resource finder matches jobs with rig availability, passes to contractor, calculates costs and forwards contract agreement to O&G developer and drilling contractor
Percepts:	O&G Developer accesses system
Data Access:	Job Optimizer, Reads Rig Availability Schedule
Interactions:	O&G Developer, Drilling Contractor

<b>004-Job Schedule Optimizer</b>	Given a list of proposed jobs, optimization criteria and constraints, determines the target job schedule.
Actions:	O&G Developer forwards job information: type, duration, location, priority, precedence
Percepts:	O&G Developer accesses Job Schedule Optimizer
Data Access:	Job Schedule Information
Interactions:	None

<b>005-Retrieve Rig Availability</b>	Allows an O&G developer to review the published rig availability as posted by drilling contractors.
Actions:	O&G Developer retrieves published rig availability schedule
Percepts:	O&G Developer requests rig schedule
Data Access:	Reads Rig Availability Schedule
Interactions:	None

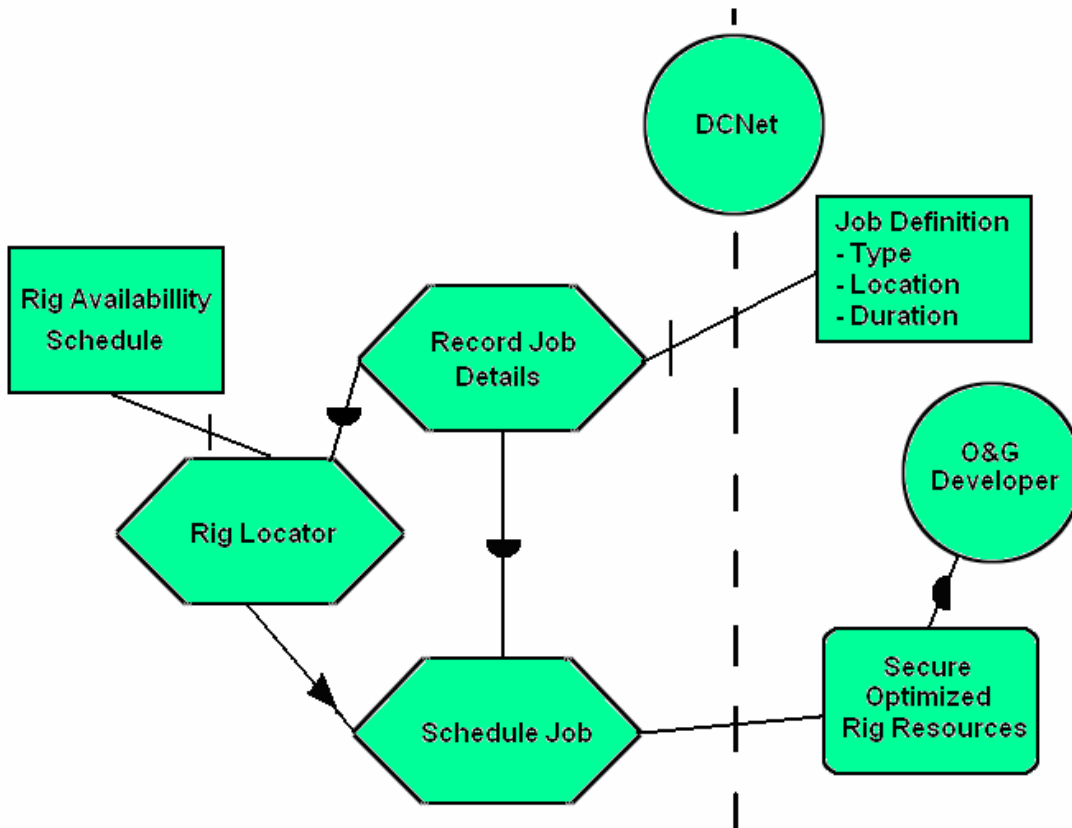
<b>006-Post Excess Rig Capacity</b>	Allows an O&G developer to post excess contracted drilling resources as available for other developers.
Actions:	O&G Developer submits unneeded contracted rig resource, updates excess capacity schedule
Percepts:	O&G developer accesses system
Data Access:	Writes Excess Rig Capacity Schedule
Interactions:	None

<b>007-Request Collaboration Partnership</b>	Allows an O&G developer to post specific information regarding activities to find other developers looking to cooperate on specific projects.
Actions:	O&G Developer submits information regarding specific jobs: location, duration, type; Post to partnership opportunities; O&G developer reviews partnerships against internal priorities and matches own needs with other O&G developer's excess capacity or own excess capacity with other O&G developer's needs, create development agreement between developers
Percepts:	Partnership opportunity posted, Partnership opportunity accepted
Data Access:	Read/Write Partnership Opportunities, Read/Write Excess Rig Capacity Schedule
Interactions:	O&G Developer (1 : Many)

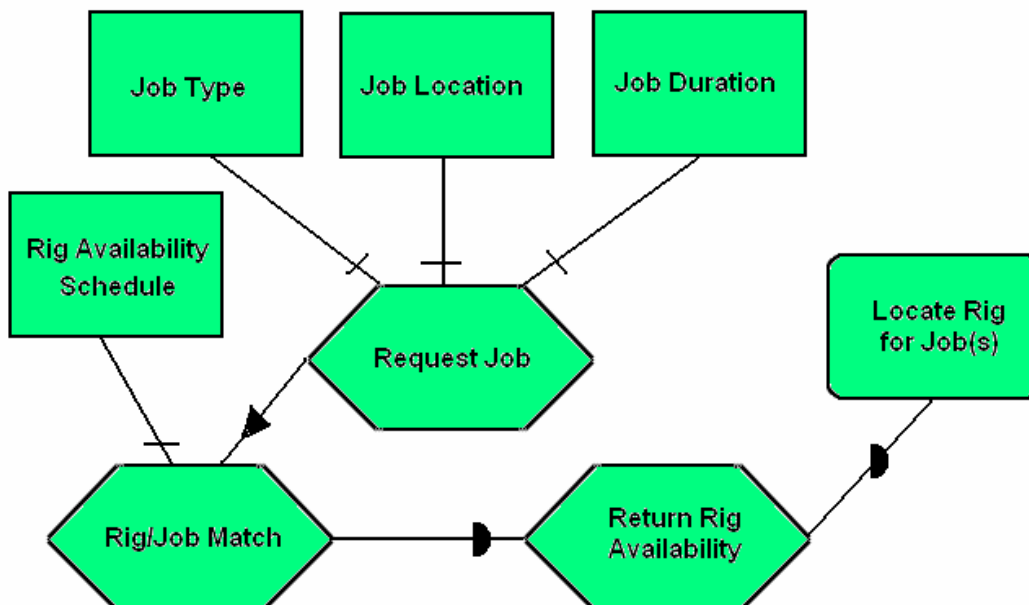
<b>008-Update Rig Availability and Contracts</b>	Allows a drilling contractor to update their rig availability schedule and inform developers who are impacted by these changes.
Actions:	Drilling Contractor updates Rig Availability Schedule, Impacted O&G Developers informed of changes, triggers scheduling function as applicable
Percepts:	Updated Rig Availability Schedule
Data Access:	Read/Write Rig Availability Schedule
Interactions:	Drilling Contractor, O&G Developers

## Appendix B - Use Case Scenarios

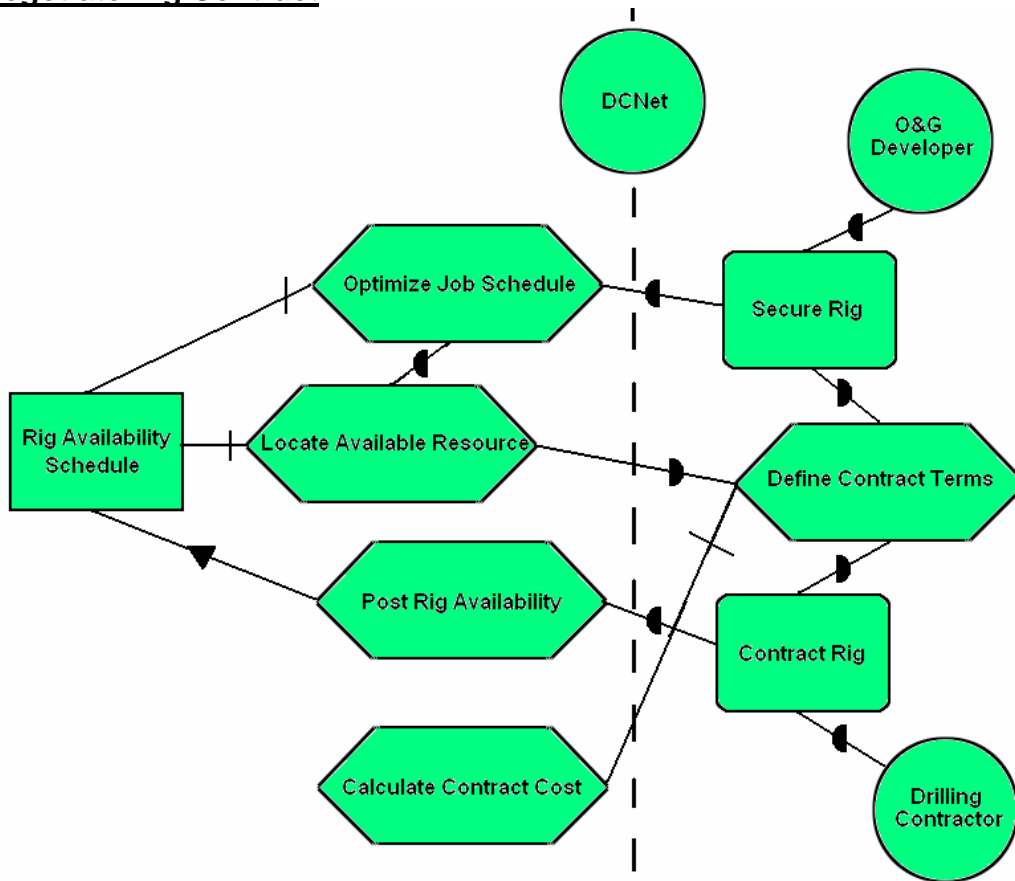
### 001 - Optimize Job Schedule



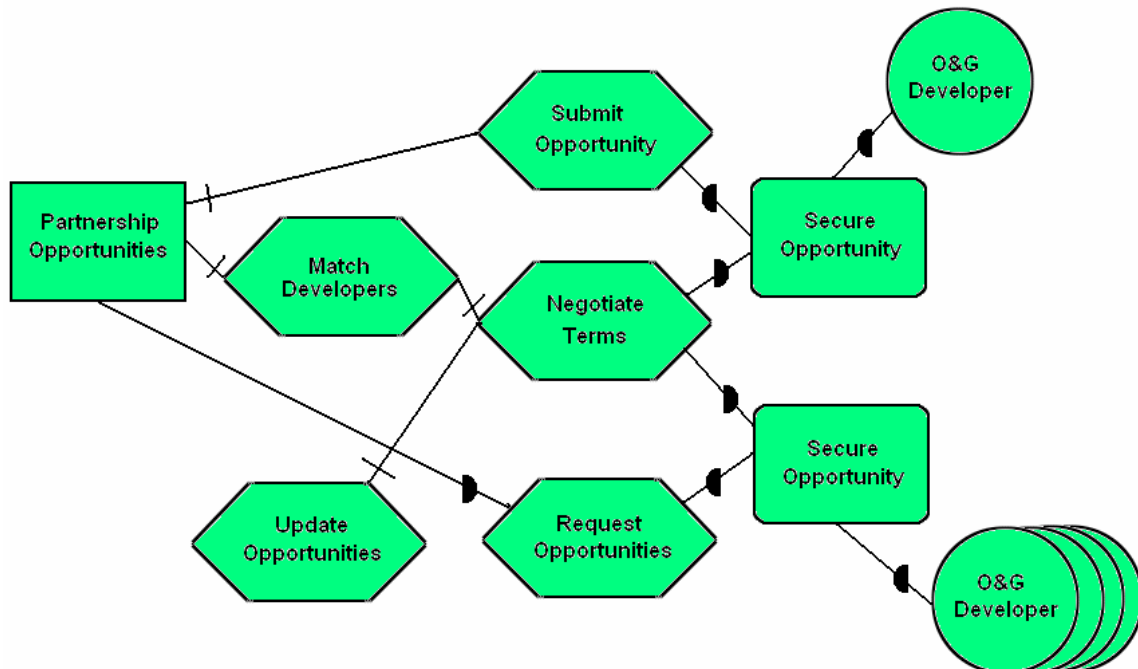
### 002 - Locate Available Rig Resource



### 003 - Negotiate Rig Contract



### 004 - Request Collaboration Partnership



## References

[Padgham2002] Lin Padgham and Michael Winikoff, Prometheus: A Methodology for Developing Intelligent Agents. Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 2002.