Distributed Intelligent Agents for Decision Making at Local Distributed Energy Resource (DER) Levels

U.S. DEPARTMENT OF ENERGY

Electric Distribution Transformation Program Peer Review

October 28-30, 2003

David Cohen, Chief Technical Officer, Infotility

Principal Investigator

Agenda

- Infotility Overview
  - → Team Experience
- Project Vision, Objectives, and Deliverables
  - → Relevance to Problems & Needs
  - → Design Principles
  - → Technical Challenges
  - → Technical Approach
  - → Project Timeline
  - → FY03 Progress and Accomplishments
- Planned Activities for FY04 (Phase II Effort)
- Impacts and Benefits
- Interactions & Collaborations

Infotility Overview

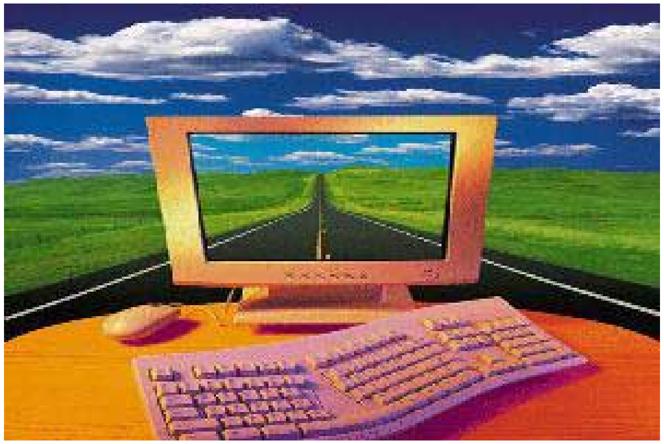
- Located in the San Francisco Bay area, with offices in Boulder, CO.
  - →Infotility provides a real-time energy information platform which enables subscription and publishing of highly relevant information when and where it is needed most
  - →The architecture supports both server-based analytic applications and distributed intelligence which allows automatic business rule processing, on-the-fly analytics, and data integration at distributed end points (meters, gateways, DERs)
  - →Platform is built natively using Web Services Architecture

Team Experience

Principal Investigator	Project Team
Pioneered the development of a utility-scale, commercially available **DER networking, communications, and control** software platform (2000)	Developed software agents for delivering **real-time pricing** and automated **demand response** services (Infotility, 2001)
Early application of **artificial intelligence** methods to energy applications: neural network-based energy forecasting applications (1990); self-organizing maps for pattern recognition (1997)	Developed a **real-time software** platform with dynamic analytic capabilities for managing energy information services (Infotility, 2001)
Developer and creator of several commercially available software applications: (Eg.) DGen Pro software model the **technical and economic performance of DER technologies** (1995)	Pioneered the use of the **publish/subscribe** paradigm and software architecture for energy information communications and control (Infotility, 2000)
Technical Committee member of FIPA's **Agent UML** (AUML), a notational system for specifying agent-based systems (parts implemented in UML 2.0 specification) (2003)	Developed NetFIT: A Windows-based **neural network** training and prediction tool (Peter Curtis)
Contributor to development of several commercially available **enterprise software** suites: (1) ICF Energy Vision (1998); (2) Silicon Energy's Enterprise Energy Management, EEM Suite application (2001)	Wrote Chapter 7 - Principles of **Control of Distributed Generation Systems**, *Distributed Generation: The Power Paradigm for the New Millennium*, A. M. Borbely and J. F. Kreider, ed., CRC Press, 2001. (Peter Curtiss)

Project Vision

- This research involves the development of intelligent software components which run at distributed locations on the energy network to improve the reliability, efficiency, security, and stability of the U.S. electrical transmission and distribution network.



Infotility

More Power To You

Project Objectives

- Develop an adaptive, intelligent agent-based information system enabling collaboration between distributed, local DER system nodes.
- Build a collection of reusable intelligent agents that will interoperate within the many interfaces and devices on the distribution level of the power delivery infrastructure.
- Build support for multiple operational criteria, with emphasis on analysis and response to electrical grid contingencies. Provide for coordination with power electronics and grid protection schemes to enhance grid reliability.

Infotility

More Power To You

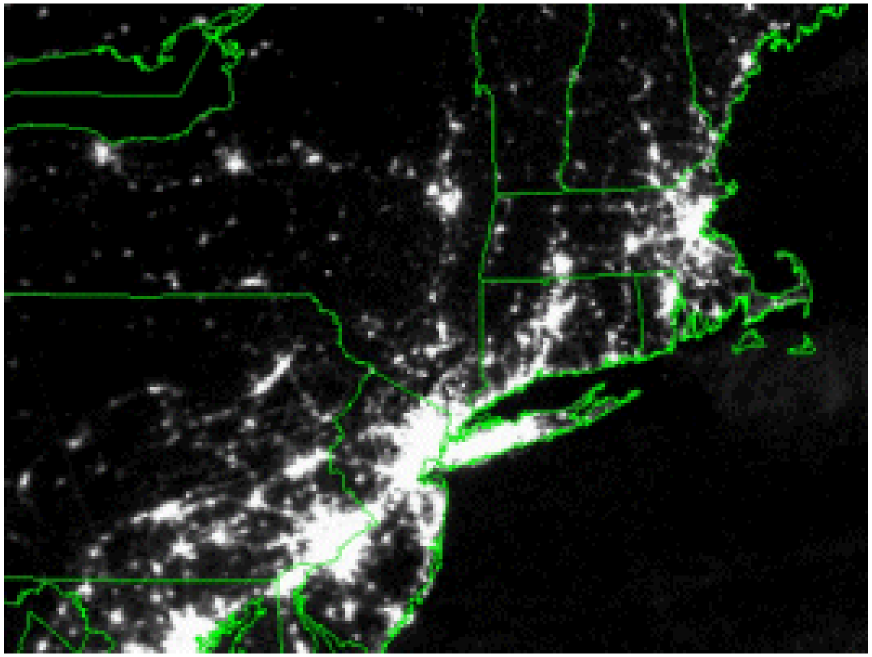
Project Deliverables

- The results of the Phase I research will be a detailed Software Development Plan and Software Architecture documents
- Results of Phase I will be used in Phase II to create a commercialized version of the software and Proof-of-Concept Testing in a real-world environment with actual DER devices connected to a major U.S. electrical system.
- Phase I started in mid **September 2003**, work complete in **April 2004**. Phase II, if funded, will start in **Mid 2004**.

Problem: Contingencies

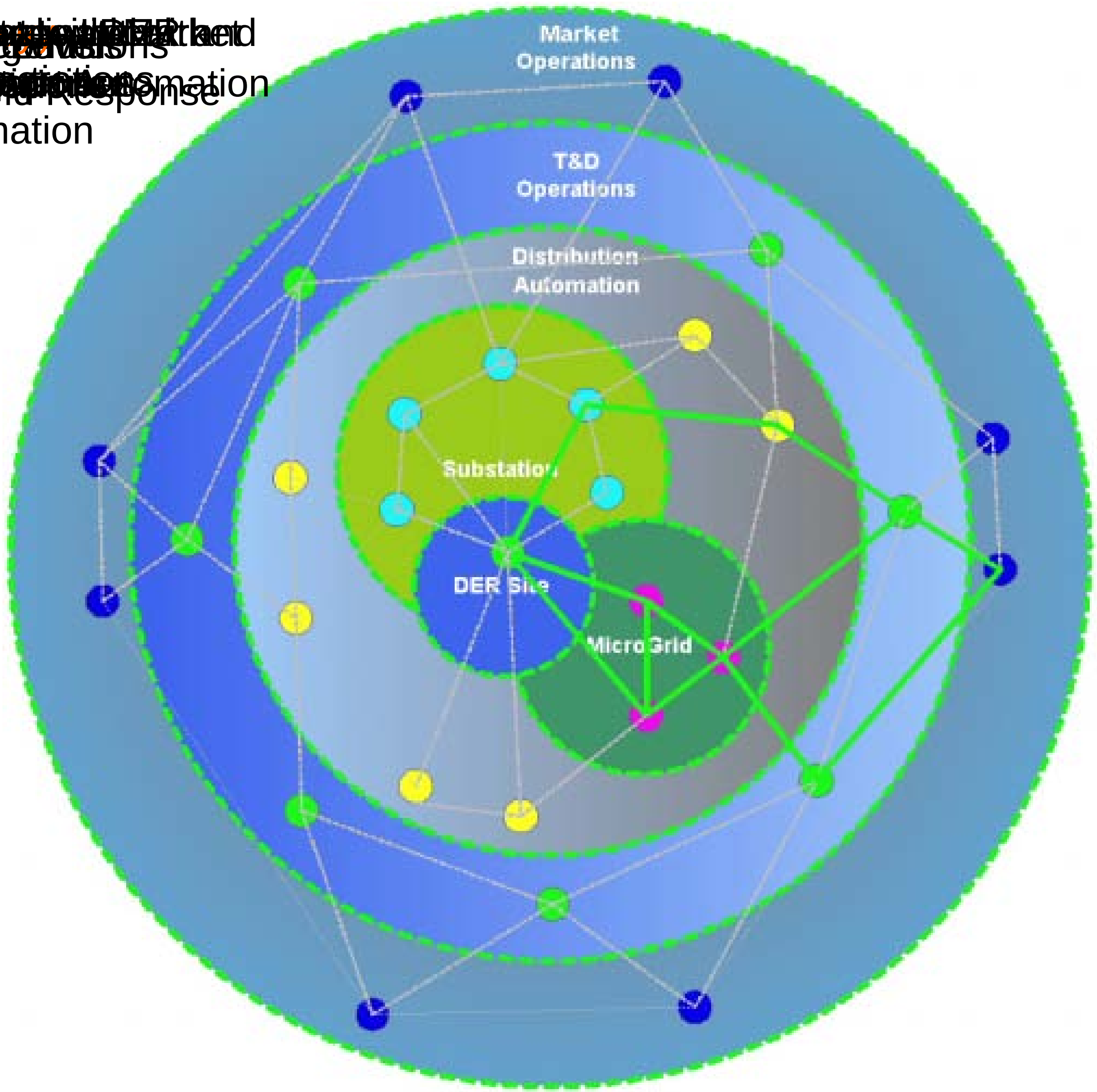
August 14, 2003
Power Outage

Before and
After Images



Part of Solution: Agents at DER levels can assist in breaking the system up into smaller islands that are self-regulating and can respond "locally" to contingencies. This supports the concept of agility and robustness, "Survivability" vs. eliminating contingencies.

Automation



Market Operations
T&D Operations
Distribution Automation
Substation
DER Site
Micro Grid

Design Principles

- Build stakeholder team with broad industry representation
  - →Requirements analysis, collaboration, review
- Build upon existing work in this area
  - →Leverage existing standards and architectures
- Design for interoperability
  - →Open communications, common data models, Semantic Web

Advisory Team

- Murray Davis – DTE Energy
- Dave Hawkins – California ISO
- Robert Burke – New England ISO
- Susan Covino – PJM
- Steve Sanders – Consultant (from GPU)
- Rick Weston - Regulatory Assistance Project
- Richard Deblasio – NREL
- Jim McCray – Siebel
- John Stevens -- SANDIA
- Steve Widegren – PNNL
- Todd Quayle -- Envenergy
- Brendan Kirby – ORNL
- Ozman Sezgen – LBL
- Chris Marnay – LBL
- Abbas Akhil – SANDIA
- Danny Julian – ABB
- Eric Wong – Cummins
- Scott Castelaz – Encorp
- Scott Samuelson – UC Irvine, NFRC
- Rob Klashner – NJ Institute of Technology
- Mukesh Kattar – Oracle
- Eric Lightner – DOE

Build Upon Existing Work

Web Services | Security | Complex Networks | Real-time Systems

AOSE | BPEL | AUML | DAML

Software Engineering

DOE C&C Program | PNNL Energy Web | SANDIA Agent R&D | RAP

CEC Agent R&D | CERTS/NERC | EPRI Agent R&D | DOD/NSF

R&D

CEIDS (ADA DER) | IECSA | GridWise | W3C

UCA/IEC 68150 | IEEE P1547 | ANSI C12.19 | BACnet / LonWorks

Standards and Guidelines

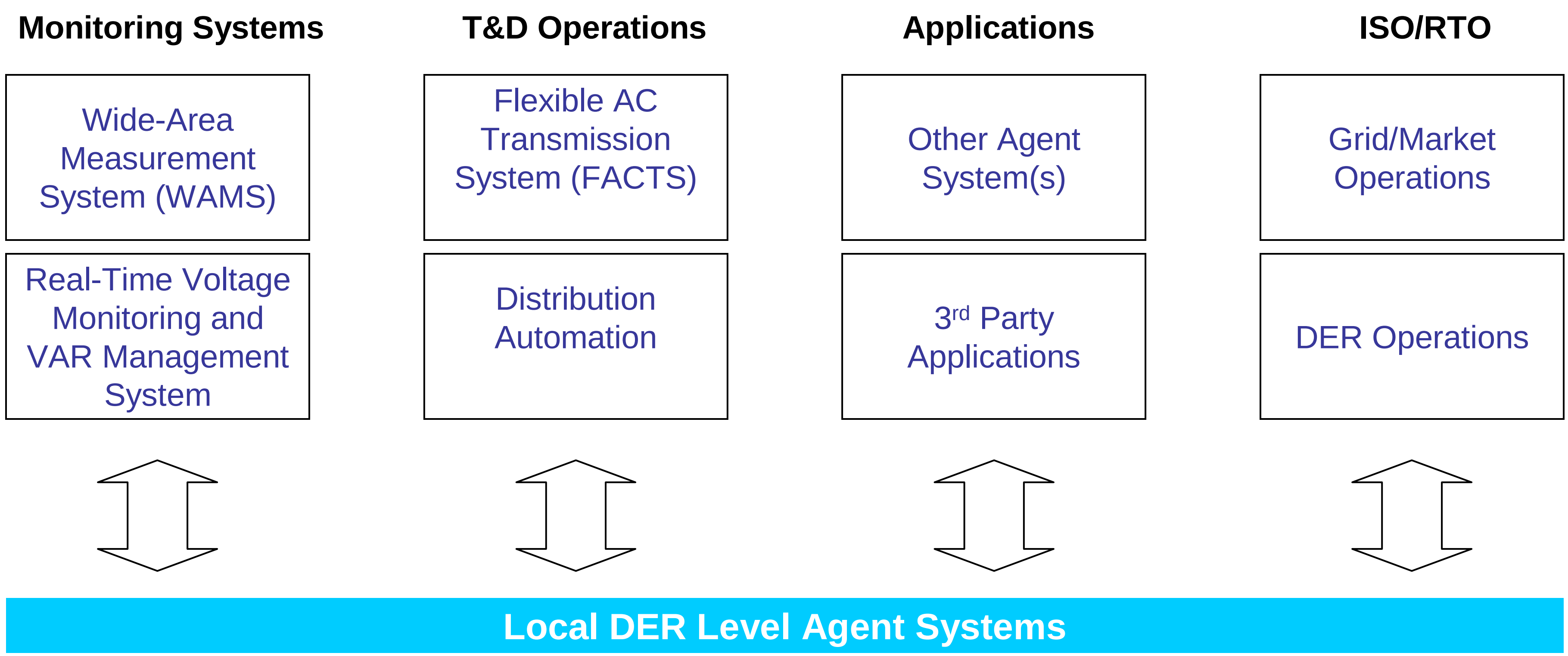
Leverage Existing Standards

- The current design will rely on a number of external technical specifications, including:
  - →Standard Internet Protocols (IP suite), XML, HTTP v1.1., SOAP, DAML-S, BPEL4WS
  - →Utility Communications Architecture (UCA), CIM, ADA/DER Object Models, ANSI C12.19
  - →Foundation for Intelligent Physical Agents (FIPA) Specifications (FIPA ACL Message Structure Specification, and others)
- Use where appropriate!

Infotility

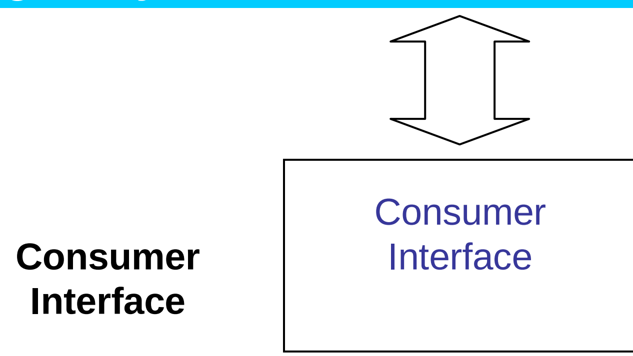
More Power To You

Interoperability



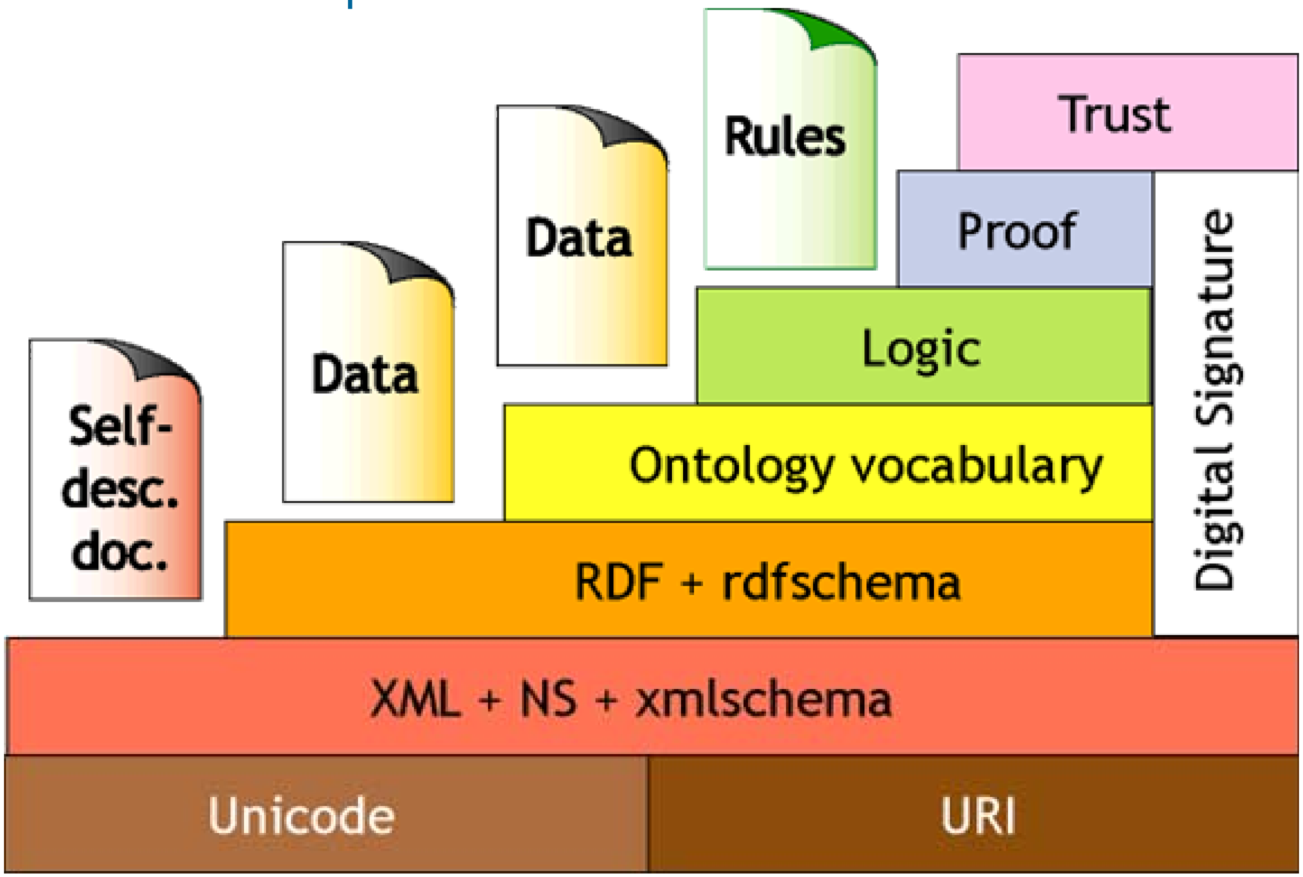
Communications Support

1. Inter-agent communications 2-way
2. Information passing 1-way
3. Messaging and control 2-way



Semantic Web

- W3C Project: Software agents navigating web of descriptions and ontologies, making inferences about data collected, communicating via partial understanding. Requires machine-readable statements about resources/relationships.



Technical Challenges

- Electrical Grid Infrastructure Issues
  - → Inherent complexity of Grid operations
  - → Centralized nature of operations and planning
  - → Grid Contingencies (500 alarms per minute)
  - → Reaction time issues
- Restructuring Issues
  - → Non-uniform solution deployment capability
  - → Moving target for regulatory requirements related to reliability, interconnections, market rules, and Federal mandates
  - → Lack of market signals and appropriate tariff designs
- System Ownership Issues:
  - → Diverse ownership of T&D assets in electric grid
  - → Large number of disparate systems which do not "speak" the same language

Technical Challenges

- Software Engineering Issues
  - → Security
  - → Immaturity of AOSE methods
  - → Network constraints (reaction time)
  - → Existing agents acts as programmed, but don't respond well to actual situation
- Economic Viability Issues
  - → Cost/benefit of reliability gains at local levels
  - → Cost to add computational resources at local levels
- Existing Paradigm Issues
  - → Operator: "I use the Bat phone to do this now, why change?"
  - → Lack of confidence in allowing autonomy at local DER levels based on real issues

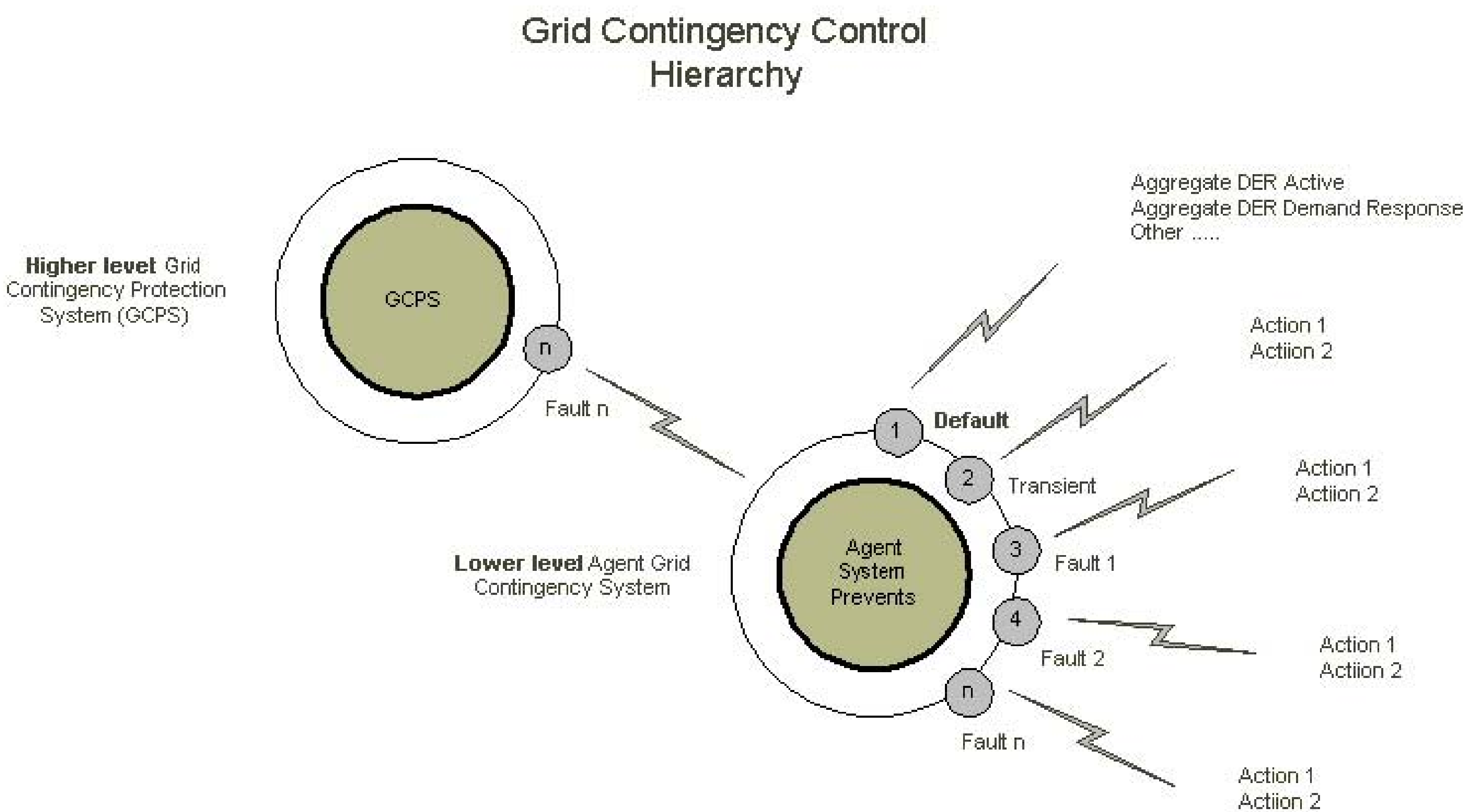
Technical Approach

- Develop a collection of agents that can collaborate towards achieving joint objectives
- Allow agents to act locally (semi-autonomous) with coordination with central control
- Use an open architecture which supports relevant standards to maintain interoperability
- Use an architecture that supports complex, real-time domains (bounded response times)
- Build-in the ability to specify domain specific, customizable business rules

Technical Approach

- Support communications with other agent platforms (FIPA-compliant)
- Support open communications using Web Service architectures (XML, SOAP, WSDL, UDDI)
- Support development of reusable, modular agent components with a “standardized interface” (automatic updates)
- Support a requirements-driven approach, standard notation methods (UML), and hybrid AOSE design methodologies
- Future-proofing the architecture by using appropriate methods/standards

Contingency Control



Agent Location Issues

- Timing issues, memory requirements, computation ability, and cost all contribute to the logical location to place agents
  - → Timing Issues: bounded response time?, critical or non-critical process?
  - → Memory requirements to run application?
  - → Computing ability at location? (embedded system, RTOS, Pentium, Server Cluster, Grid computing)
  - → Incremental cost vs. functionality or benefit

DER Services

- Building Control Level (System Reliability)
  - Aggregate demand response (1000s)
    - Eg. Carrier Comfort Choice (thermostats)
    - Eg. LBL's IBECS load controls (lighting)

- DER Level (Aggregate Services)
  - Active generation, demand response, load following, spinning reserve, emergency, voltage/VAR support

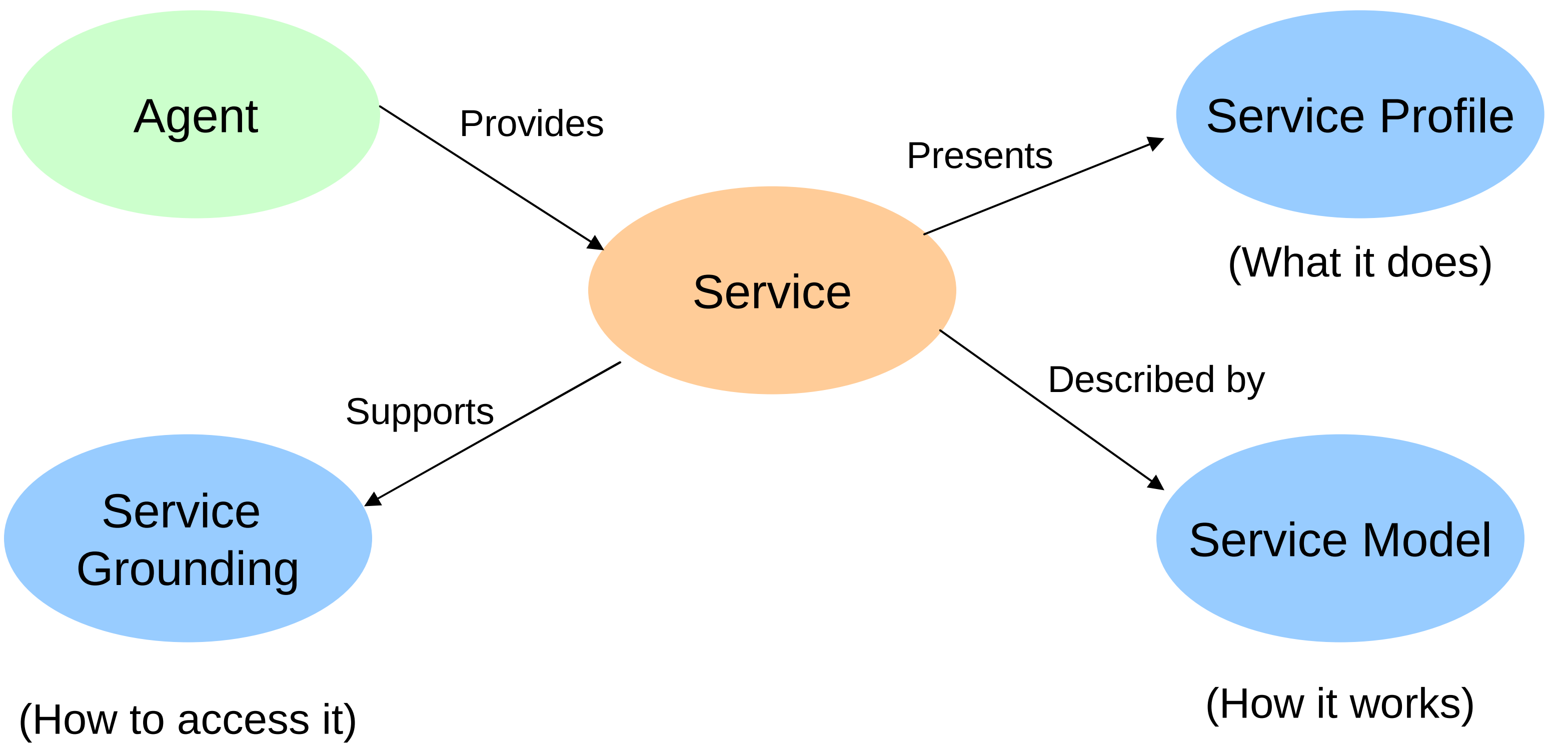
DER Services

- Distribution System Level
  - →Local circuit switching, fault isolation, clearing
  - →Provision of local reliability services
  - →Backup generation for self-supporting islands (campus, microgrids, etc.)
  - →Request for aggregate services at DER and Building levels
  - →Monitor feeders in real-time to optimize voltage and minimize reactive power flow
  - →Deliver routine status information to upstream applications (central operations)

Service Delivery

DAML-S

Automatic Discovery, invocation, composition, interoperation, execution monitoring



Project Tasks – Phase I

- **Task 1**: Develop End User Requirements
- **Task 2**: Develop Intelligent Agent Requirements and Architecture
- **Task 3**: Determine the Data Model and Communication Requirements
- **Task 4**: Determine the Decision Logic, Analytics, and Business Rules Requirements
- **Task 5**: Write Software Requirements, Engineering Specifications, and Final Report

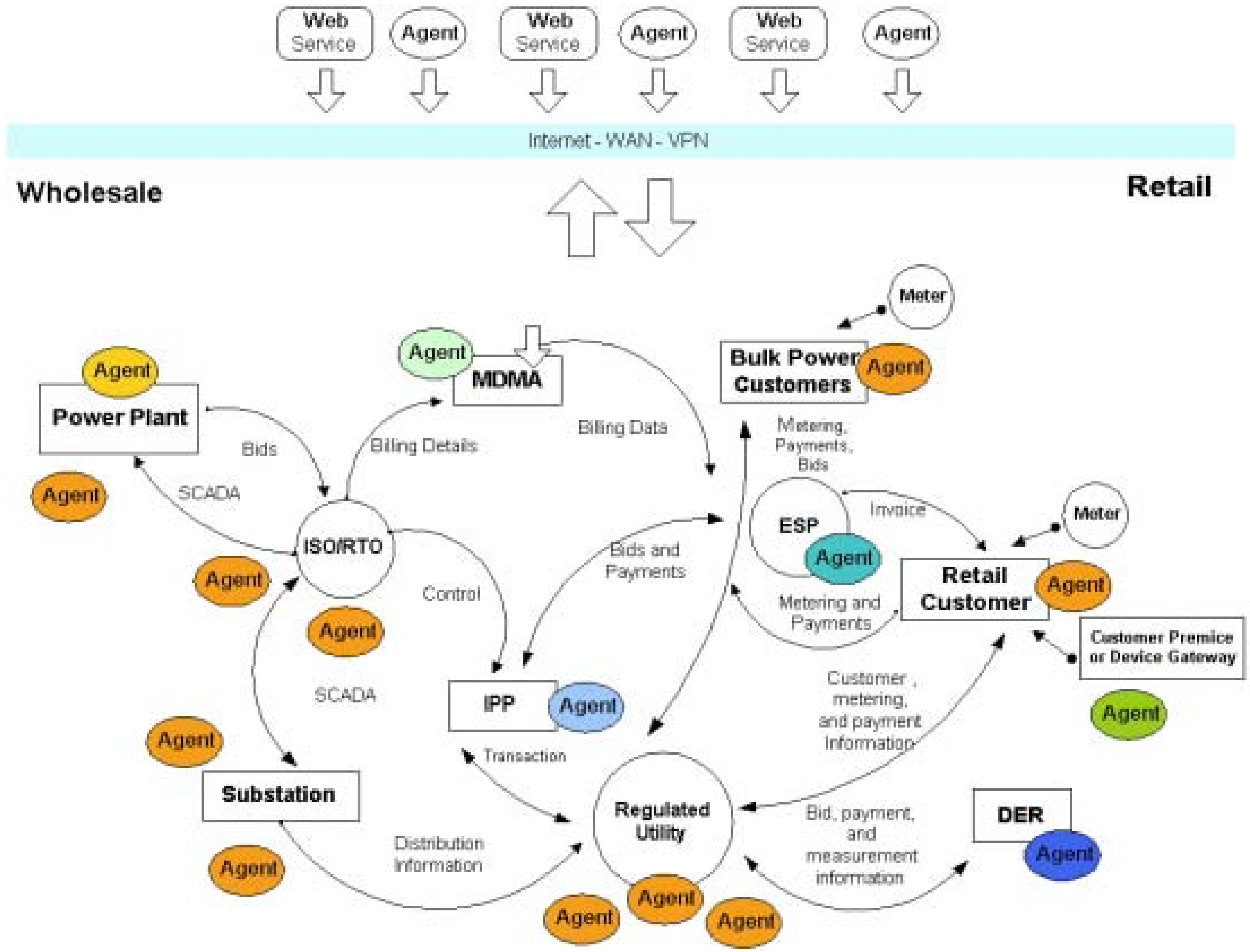
Task 1

- **Task 1: Develop End User Requirements**
  - → Identify a set of key end users and stakeholders
    - ❖ Create Stakeholder Team
    - ❖ Software requirements survey
    - ❖ Interview stakeholders to create enhance preliminary requirements
  - → Review Prior work on agents (EPRI, CEC/AESC, SANDIA
  - → Use industry accepted *Rational Unified Process* for requirements development
  - → Use best available AOSE analysis/design methodologies for MAS requirements development

Task 1

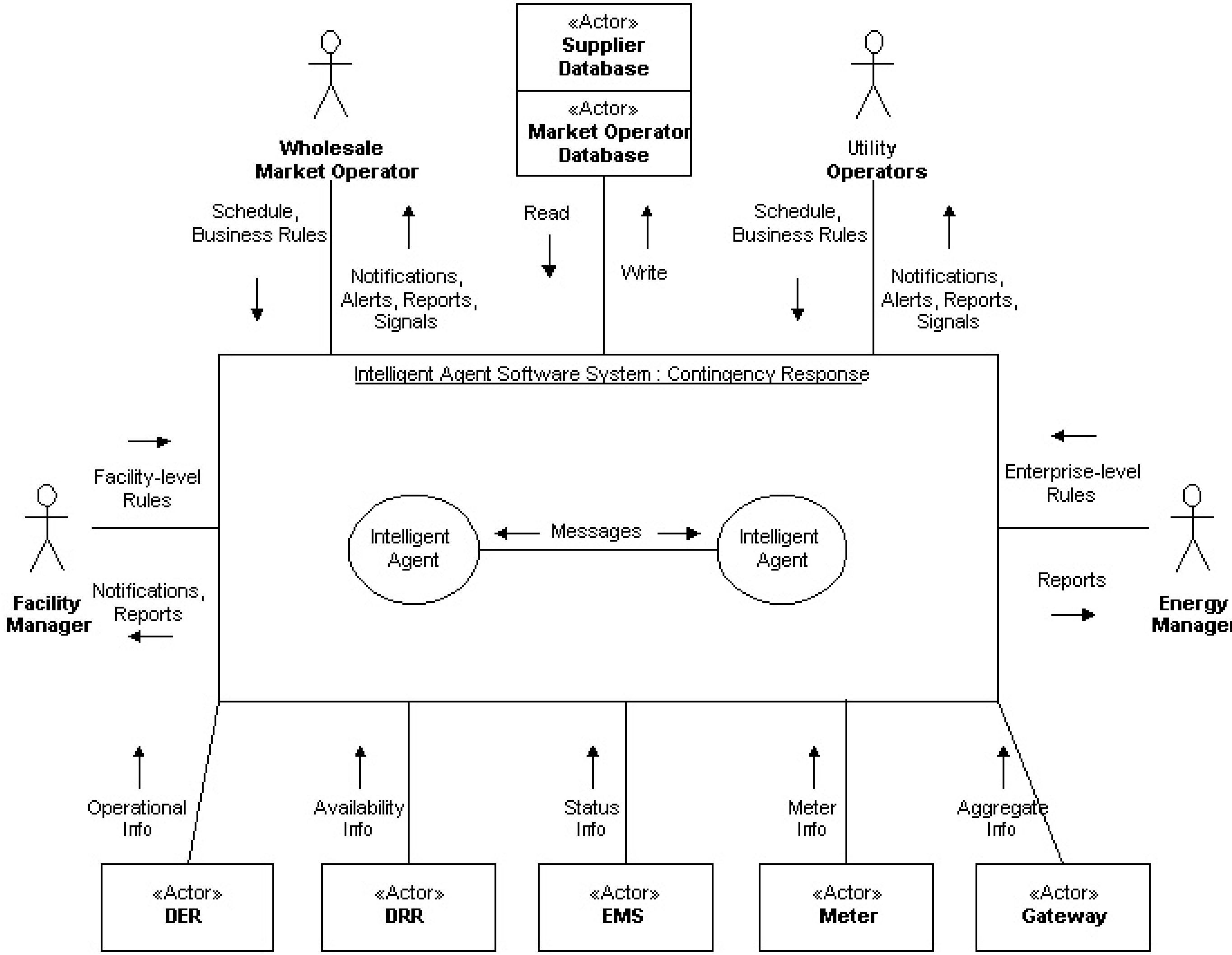
- **Task 1: Develop End User Requirements**
  - → Preliminary "Problem" Analysis
    - ❖ End-users
    - ❖ Utilities and Market Operators
  - → Preliminary User Needs (business process, organization chart, goals)
  - → System View: Actors and Stakeholders, Non-actor
    - ❖ Non-actor Stakeholders
  - → UML Context Diagram for Intelligent Agent-based DER Services (Grid Contingency Response)
  - → Preliminary Project Constraints

System View



Web Service
Agent
Internet - WAN - VPN
Wholesale
Retail
Meter
Agent
MDMA
Bulk Power Customers
Power Plant
Bids
Billing Details
Billing Data
Metering, Payments, Bids
SCADA
ISO/RTO
ESP
Invoice
Bids and Payments
Control
Metering and Payments
Retail Customer
Customer Premice or Device Gateway
IPP
Customer, metering, and payment Information
Transaction
Substation
Regulated Utility
Bid, payment, and measurement information
DER
Distribution Information

UML Context Diagram



Task 2

- **Task 2: Develop Intelligent Agent Requirements and Architecture**
  - →Build upon and utilize well known, widely accepted, next-generation Agent-oriented Software Engineering (AOSE) methods
    - ❖Incorporate methods from software engineering, distributed systems, and artificial intelligence to unify architectural approach
  - →Design for integration with Web Services, Workflow Management, and Service Discovery as part of overall architecture design

Task 2

- **Task 2: Develop Intelligent Agent Requirements and Architecture**
  - → Extensive literature review performed on the following:
    - ❖ Current Intelligent Agent Books (7), research publications (>20), software engineering analysis and design methodologies (>6)
  - → Principal Investigator is a Working Group Member for Agent UML (AUML)
  - → Significant work completed on AOSE Analysis and Preliminary Design for to support the architecture for this project.

Infotility
More Power To You

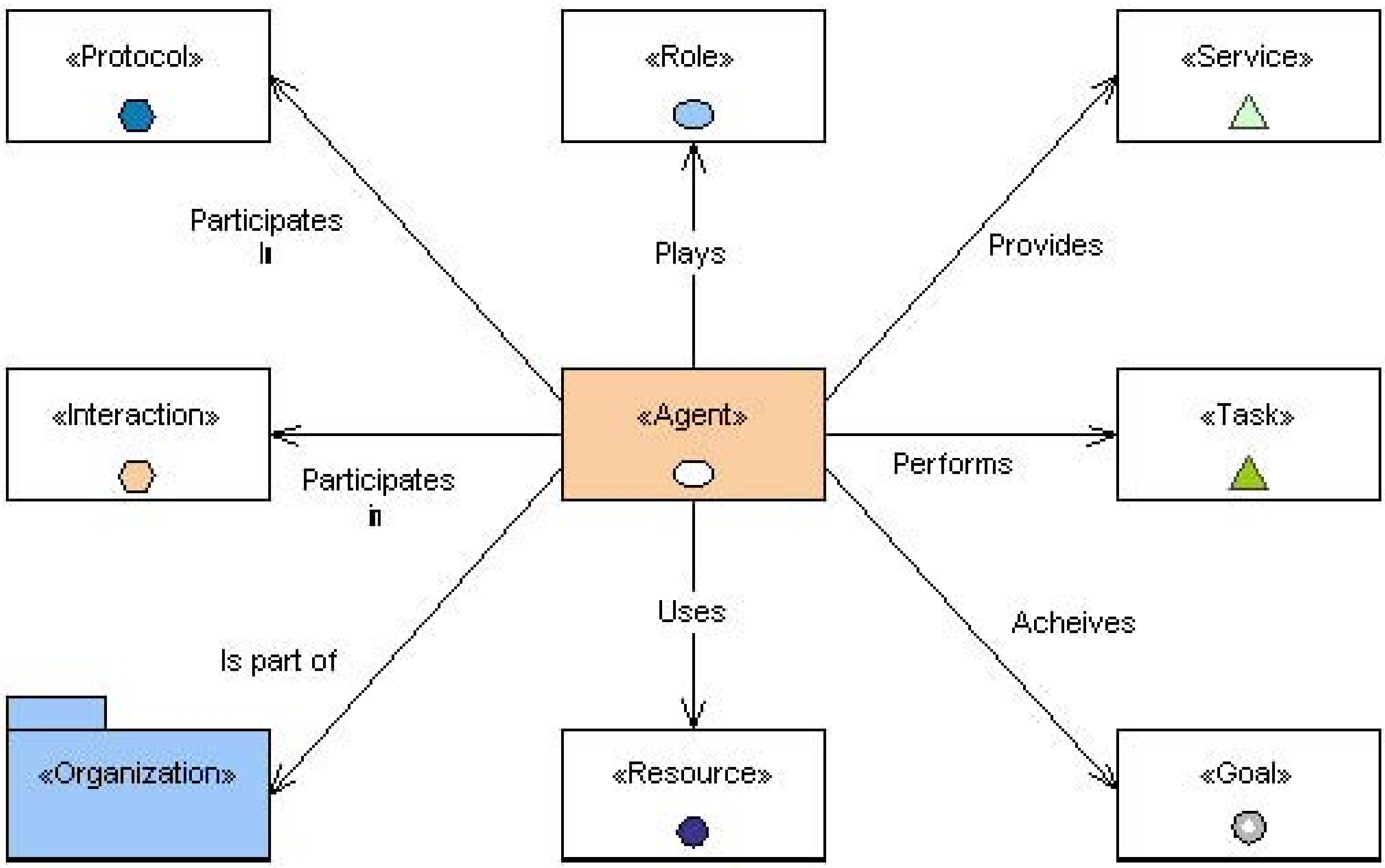
Task 2

- **Task 2: Develop Intelligent Agent Requirements and Architecture**
  - → Agent Meta-Model (UML)
  - → Develop Agent "Views" in order to understand the System, its components, interactions, and goals (Use UML 2.0 and AUML)
    - ❖ System Views – Architecture/Organization/Environment
    - ❖ Resource Views
    - ❖ Use Case and Actor Views
    - ❖ Agent and Agent Role Views
    - ❖ Goal/Task Views
    - ❖ Service and Workflow Views
    - ❖ Interaction Views (Sequence, Interaction, Collaboration, Timing)

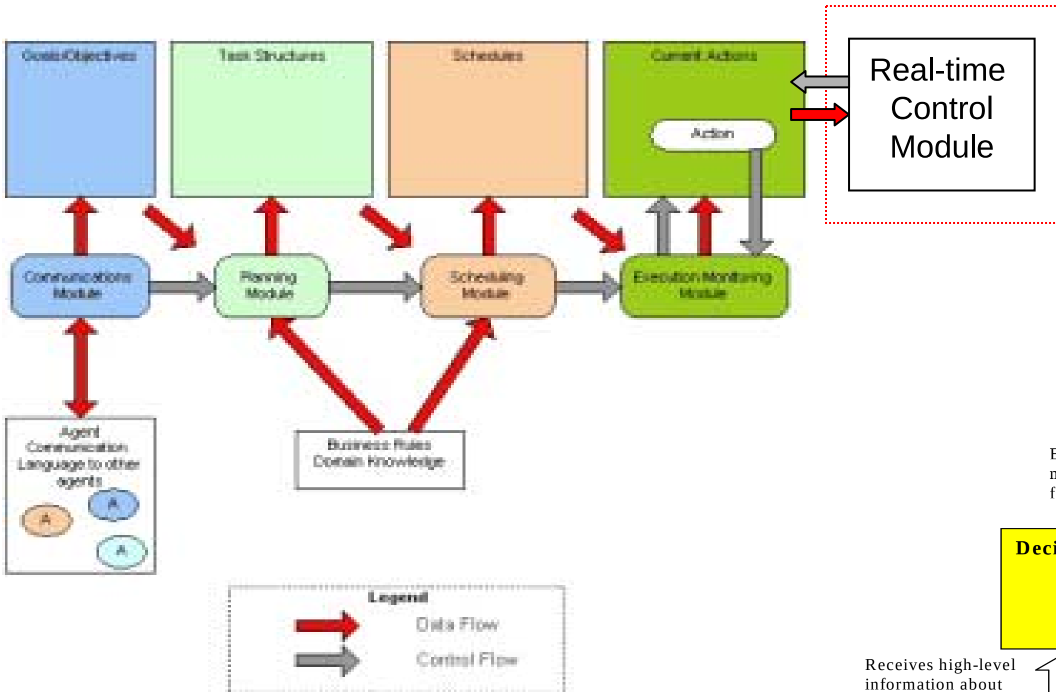
Task 2

- **Task 2: Develop Intelligent Agent Requirements and Architecture**
  - →Architecture assumptions for project domain
  - →Defined as a "Social Real-Time Domain"
    - ❖Collection of autonomous agents collaborating to achieve a common long-term goal
    - ❖Agents periodically send messages with no adverse effects on goal
    - ❖Domain is "Time-Bounded"
    - ❖Domain has "Unreliable Communications"
      - ❖Transmission reliability, or bandwidth limits

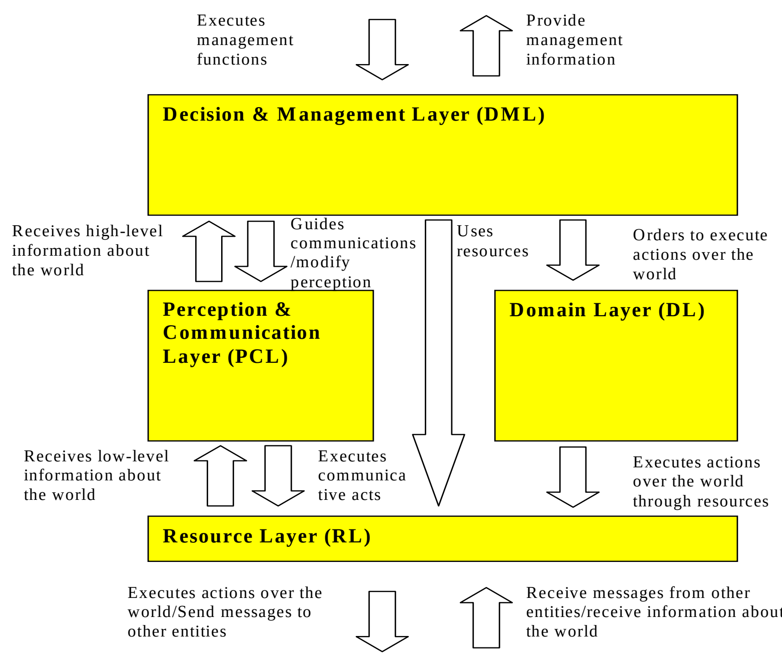
Agent Architecture Meta-Model



Agent Architecture View

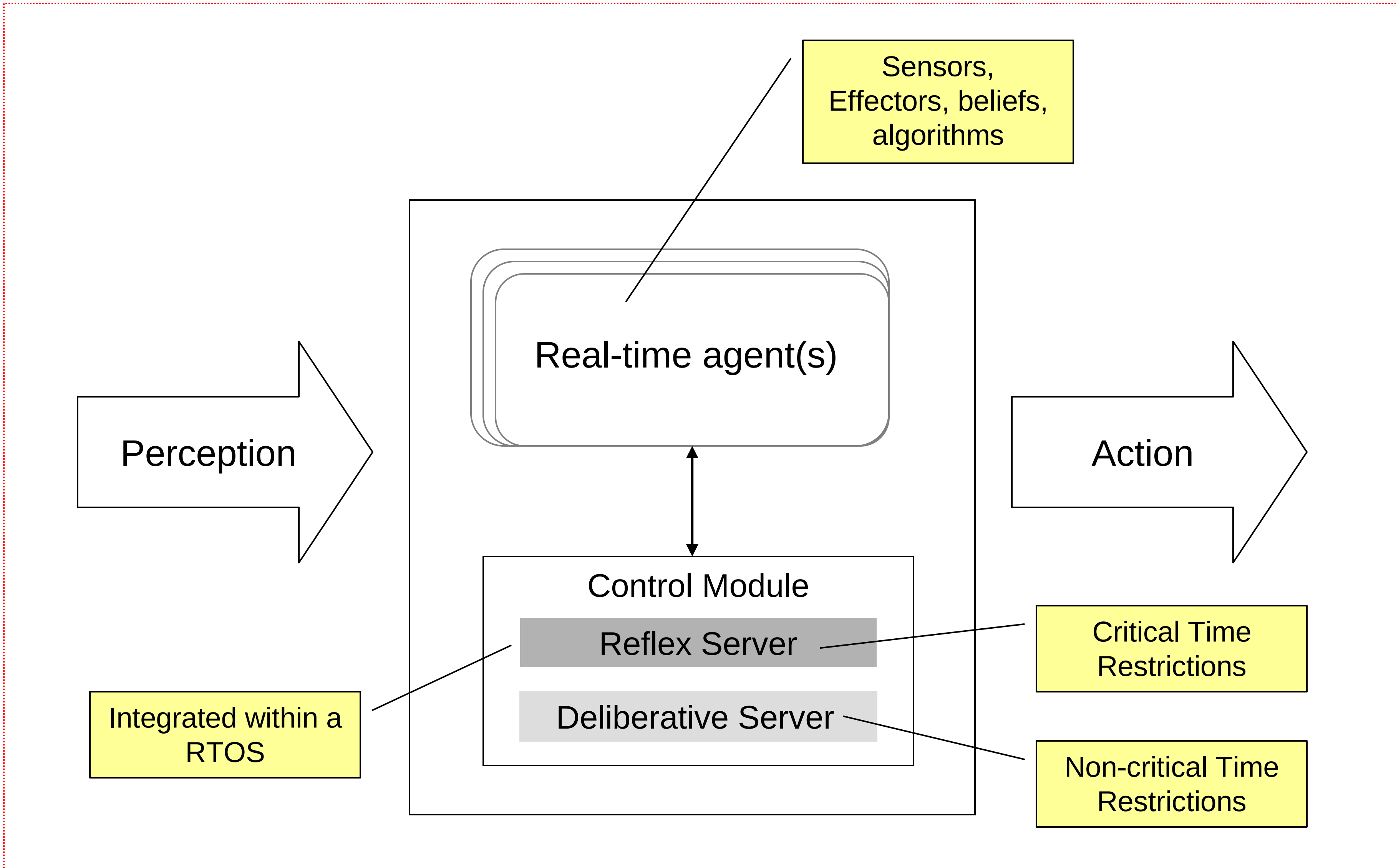


Agent Module View



Agent Layer View

Real-time Architecture



MAS Architecture

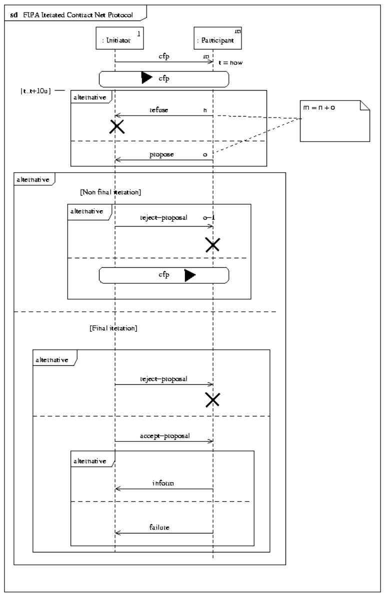
- Real-time Architecture
  - →Which agents require real-time architecture?
  - →Based on critical temporal restrictions
- Distributed Collaboration Architecture
  - →Specification Sharing Agent System
    - ❖Agents decompose and share (all agents equal)
  - →Contract Nets
    - ❖Agents negotiate tasks (all agents equal)
  - →Federated System
    - ❖Agents interact with a facilitator agent (Master or Middle agent controls behavior)

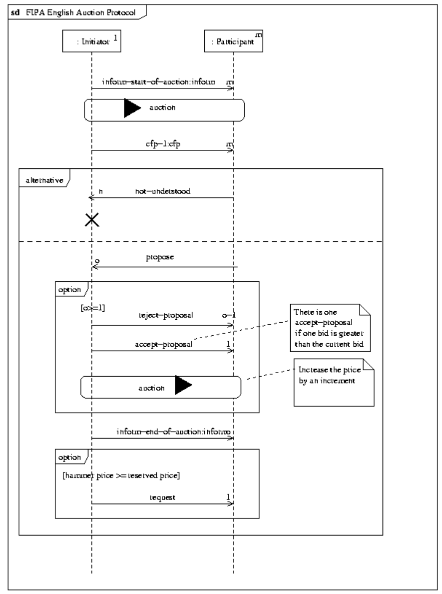
MAS Architecture

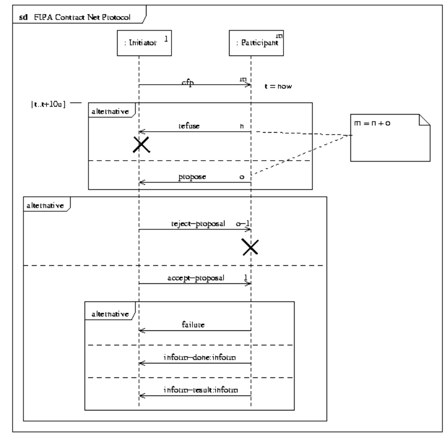
- FIPA-compliant
  - →Allow for future inter-communication with agents from other platforms
  - →FIPA-compliant- implements ACL and Agent Management specifications
- Choose Operating System(s)
- Integrated Development Environment (IDE)
- Potential Testing Environments (AgentCities Network, Sandia, UCI, NREL)

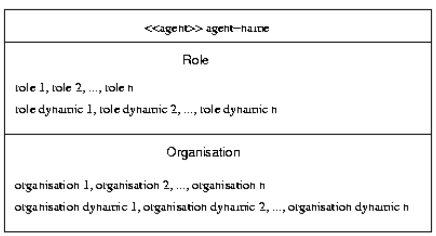
Infotility
More Power To You

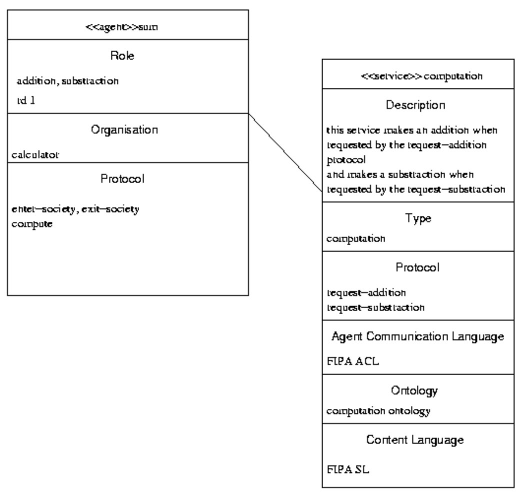
AUML Artifacts

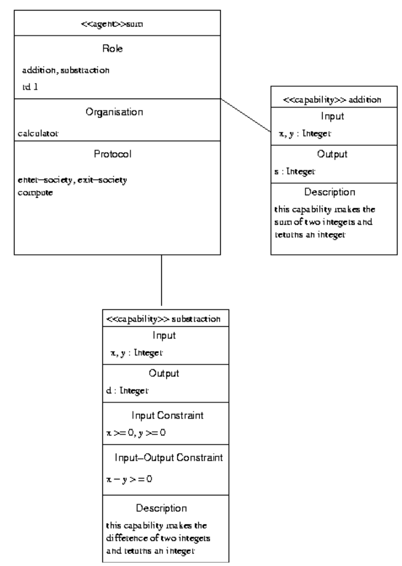




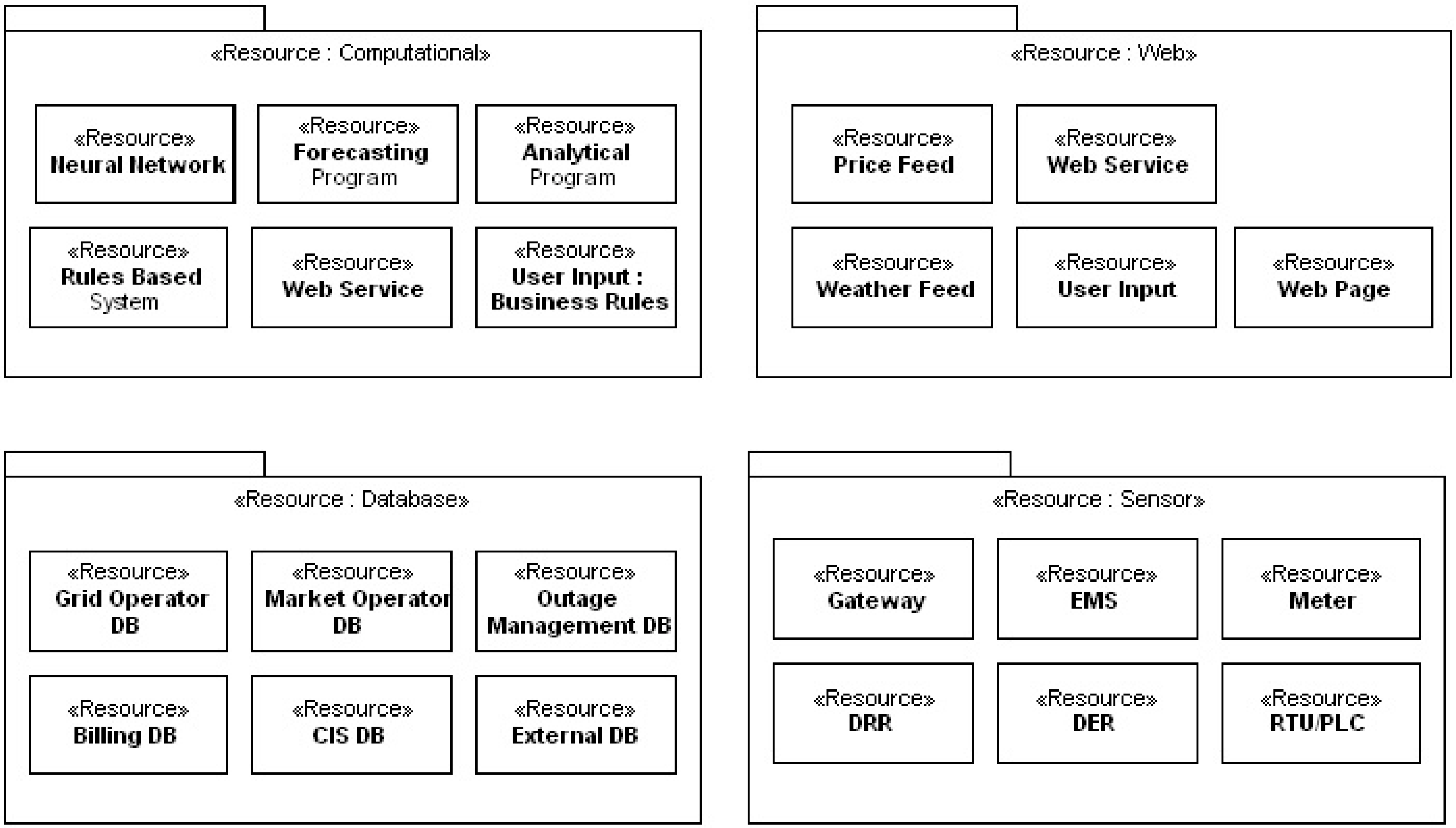




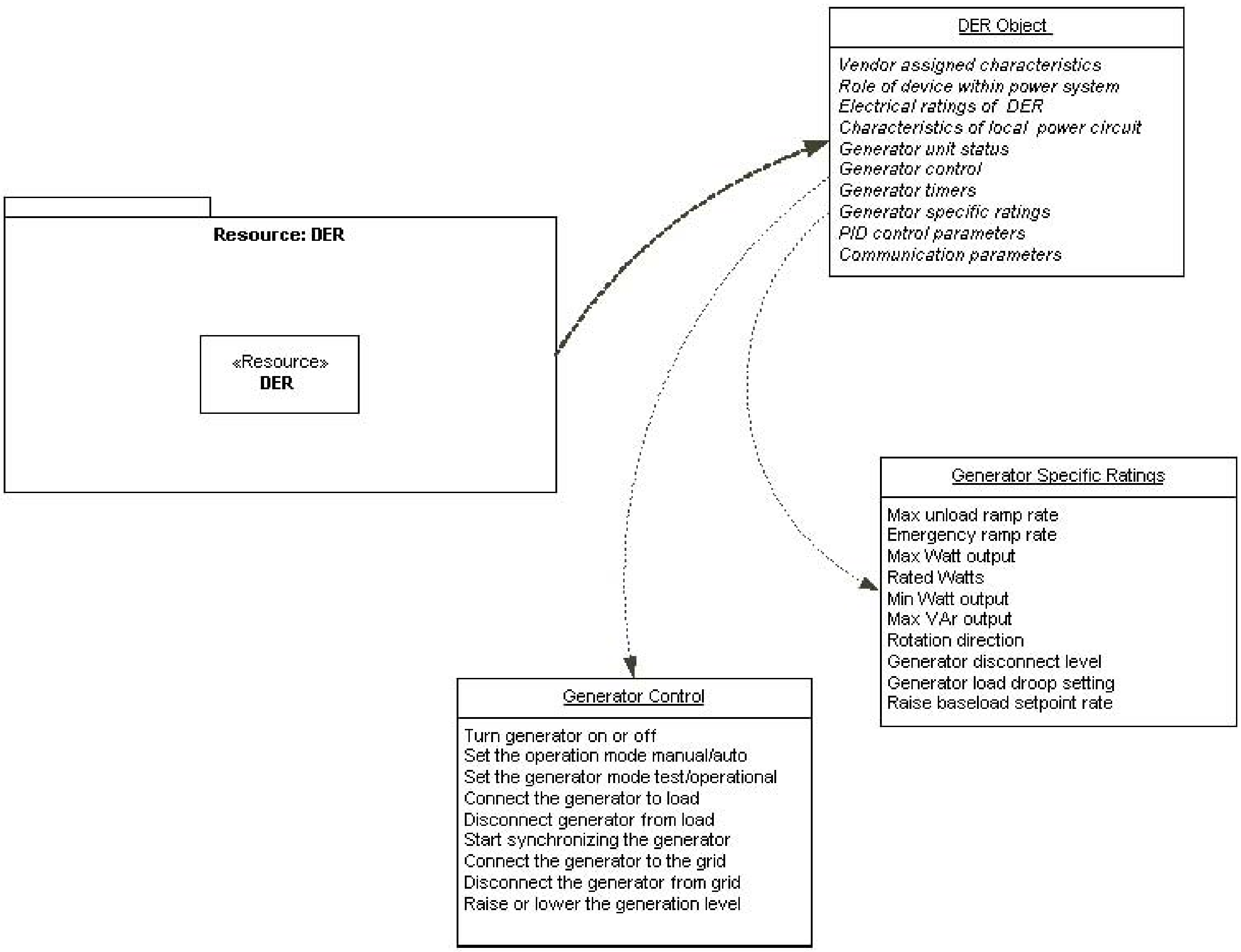




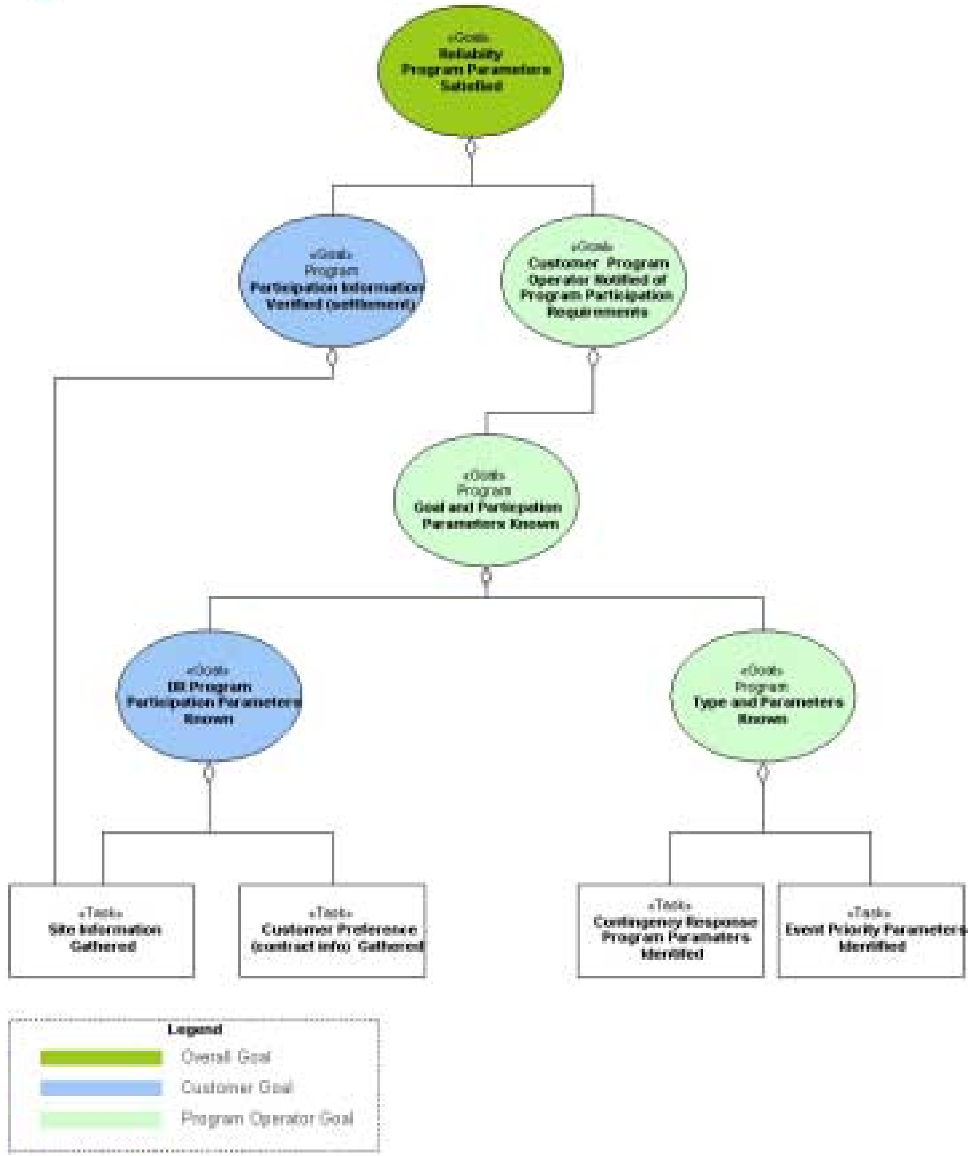
Resource View



Resource View: DER Object Models



Goal View



«Goal» Reliability Program Parameters Satisfied
«Goal» Program Participation Information Verified (settlement)
«Goal» Customer Program Operator Notified of Program Participation Requirements
«Goal» Program Goal and Participation Parameters Known
«Goal» DR Program Participation Parameters Known
«Goal» Program Type and Parameters Known
«Task» Site Information Gathered
«Task» Customer Preference (contract info) Gathered
«Task» Contingency Response Program Parameters Identified
«Task» Event Priority Parameters Identified
Legend
Overall Goal
Customer Goal
Program Operator Goal

Agent Properties

- Agent Task Environment Properties
  - → Fully observable, partially observable
  - → Deterministic vs. Stochastic
  - → Episodic vs. Sequential
  - → Static vs. Dynamic
  - → Discrete vs. Continuous
- Agent Types (Broker, Task, Resource, Interface)
  - → Subtypes (All can be turned into learning agents)
    - ❖ Reflex Agent, Model-based Reflex Agent, Goal-based Agent, Utility-based Agent, Learning Agent
- Agent Components
  - → Type, Role, Tasks, Goals, Services, Interactions, Constraints

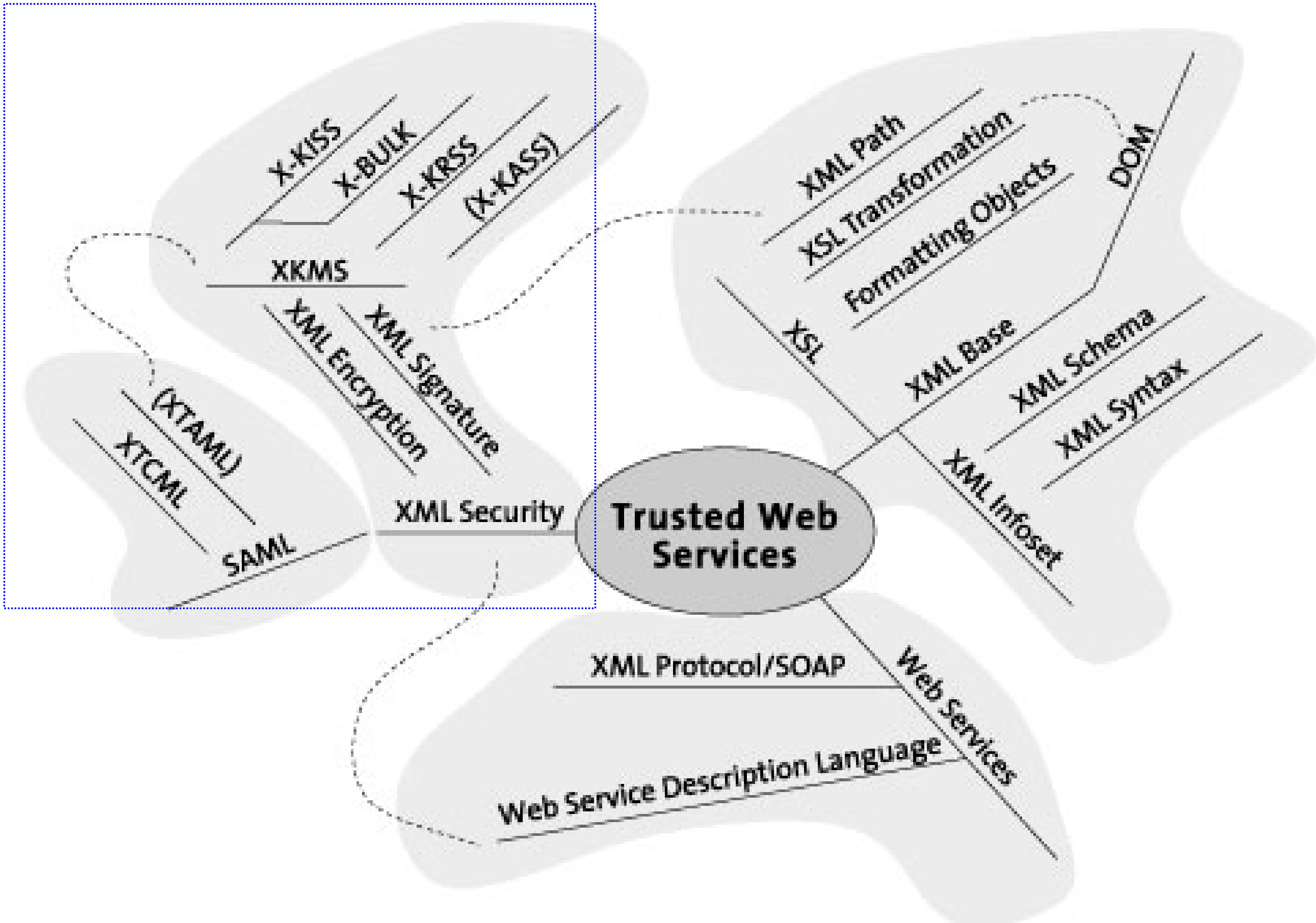
Task 3

- **Task 3: Determine the Data Model and Communication Requirements**
  - → Develop data model architecture to captures the data communications requirements needed by the intelligent agents.
  - → Review Utility Communications Architecture (UCA) Version 2.0
    - ❖Review ANSI C12.19 Metering Data Models (Include in Meter object model)
    - ❖Review CIEDS/E2I Common Object Models for DER (Advanced Distribution Automation)
    - ❖Review CIM/XML efforts
    - ❖Review of Web Services Communication Methods (XML and SOAP)

Task 3

- **Task 3: Determine the Data Model and Communication Requirements:**
  - → For locating services within agent platform and registering with others (eg. WSDL, DAML-S)
  - → For Agent Communication Language (ACL) for *internal* agent-to-agent communications
  - → For XML Schema for *external* agent-to-other communications (eg. XML over SOAP). (discovery, abstraction, interactions)
  - → For mapping ACL to XML protocol
  - → For domain "Ontology" (Register with FIPA)
  - → For use of Business Process Execution Language for Web Services (BPEL4WS) to manage workflows
  - → For Integration Issues
  - → For Security Issues (Web Service, Distributed Network, SCADA)

Web Service Security

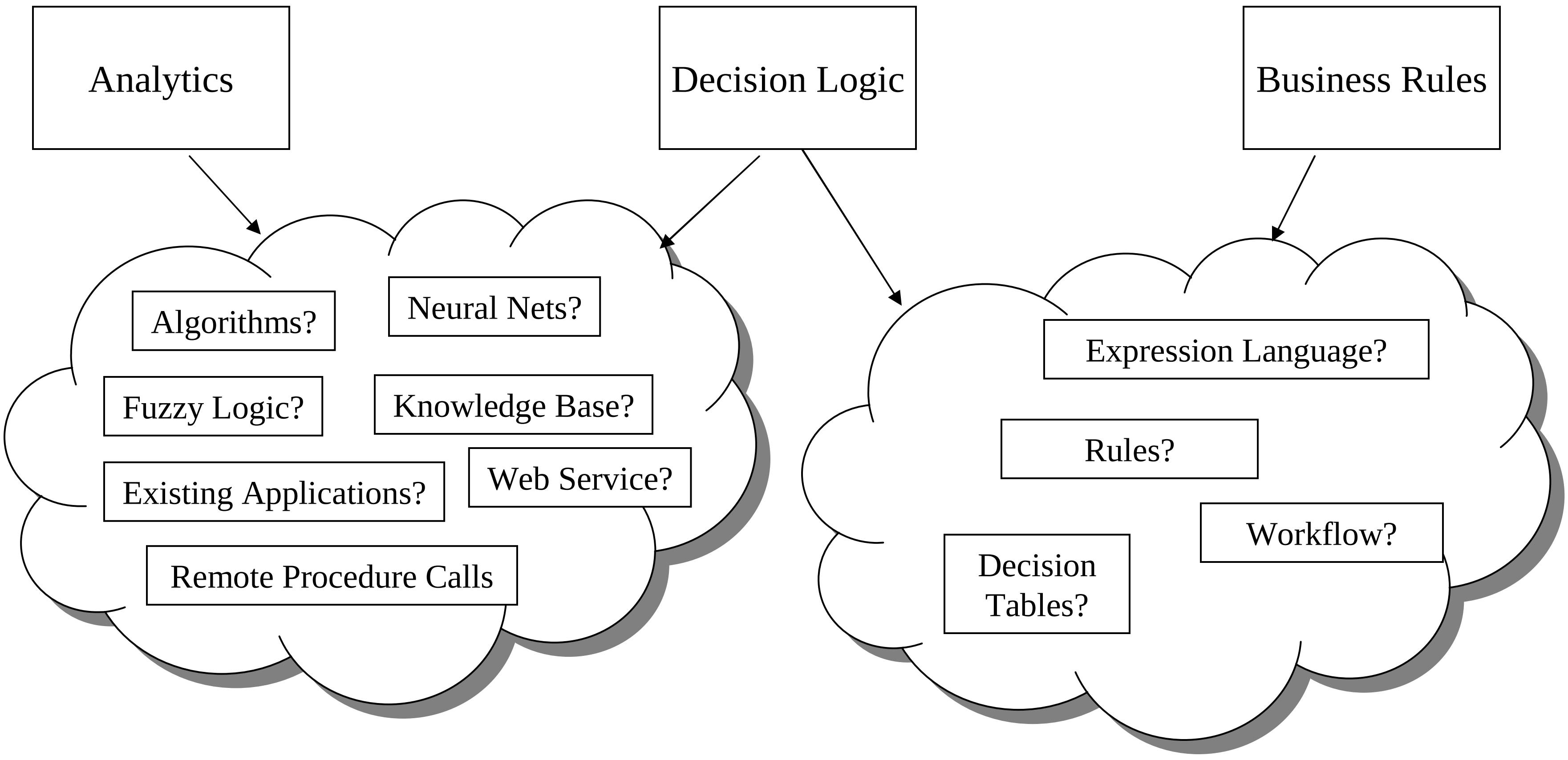


Task 4

- **Task 4: Determine the Decision Logic, Analytics, and Business Rules Requirements**
  - → The agents will need to interpret various information sources in order to perform the reliability and other decision-making functions as part of the overall requirements.
  - → Results based on Requirements Analysis (Task 1), Tasks and Services work
  - → What Resources do they need to accomplish to support the Services and Tasks defined above?

Task 4

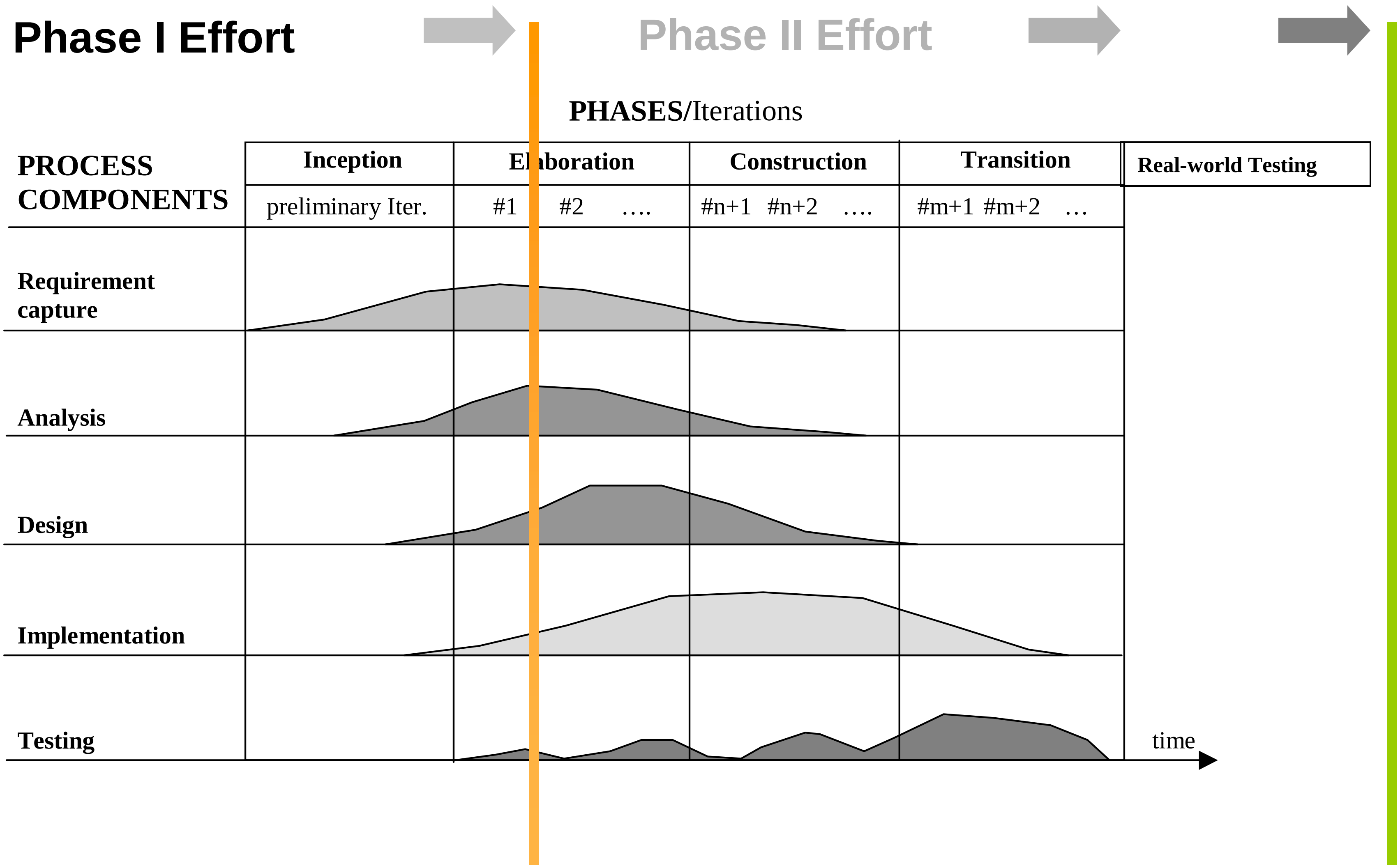
- **Task 4: Determine the Decision Logic, Analytics, and Business Rules Requirements**



Task 5

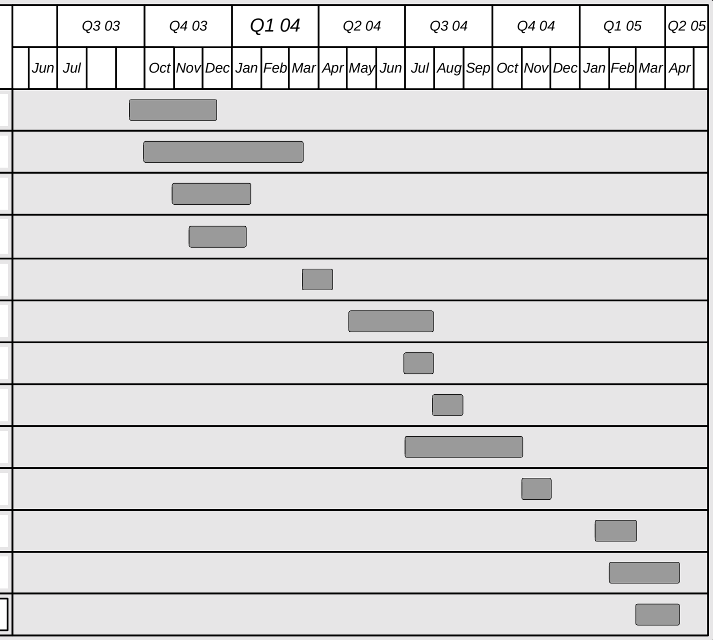
- **Task 4: Write Software Requirements, Engineering Specifications, and Final Report**
  - →**Data Glossary**
  - →**Preliminary Ontology Requirements**
  - →**UML Views and Diagrams**
  - →**AUML Views and Diagrams**

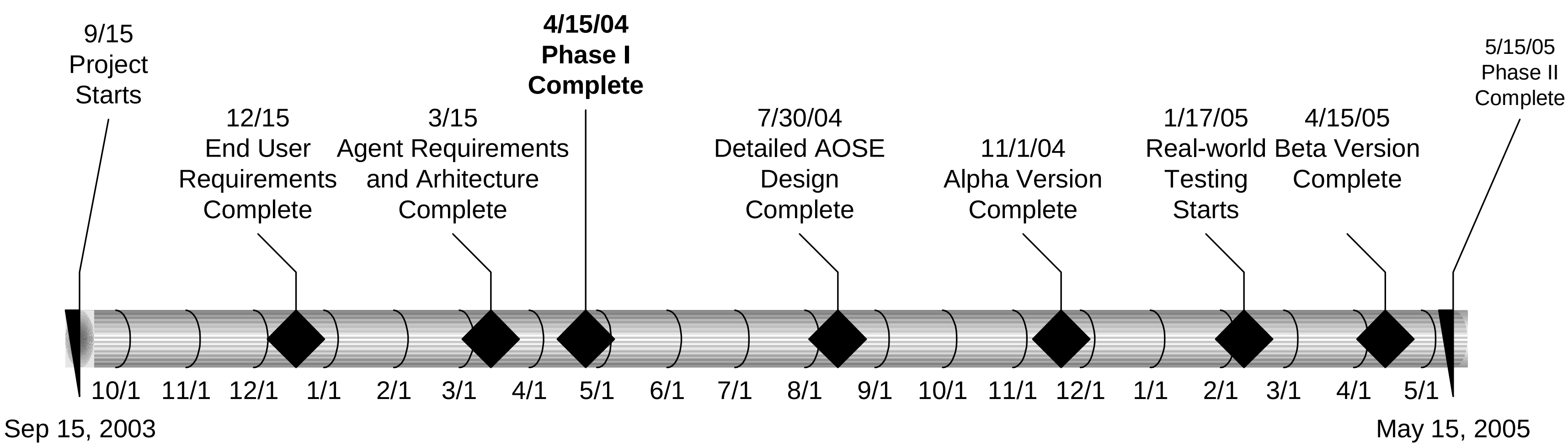
Project Timeline



Project Timeline

ID	Task Name
1	Task 1 Develope End-user Requirements
2	Task 2 Develop Intelligent Agent Architecture and Requirements
3	Task 3 Determine Data Model and Communication Requirements
4	Task 4 Decision Logic, Analytics, Business Rules Requirements
5	**Task 5 Write Final Report (Phase I Complete)**
6	Task 6 Detailed AOSE Design
7	Task 7 MAS System Prototyping
8	Task 8 Agent Sub-system Verification and Testing
9	Task 9 Develop MAS System Alpha Version
10	Task 10 Detailed System Testing and Verification
11	Task 11 Proof-of-Concept Testing with Real-world Testbed
12	Task 12 Develop Beta Version of Software (MAS System)
13	**Task 13 Final Report (Phase II Complete)**





FY04 Activities: Phase II

- Detailed AOSE Design, Elaboration, Construction, Transition, Testing, Demonstrations
- Prototyping with stakeholder feedback
- Agent System Testing and Verification
- Develop Alpha Version
- Detailed Agent Testing and Verification
- Proof-of-Concept testing within major utility distribution system with multiple DER technologies connected to a major ISO (Possible AOSE network integration testing)
- Develop Beta Version
- Cost/benefit analysis

Impacts and Benefits

- Local agents can help deal with large-scale contingencies by reducing response time
- Supports targeted PQ and reliability such as spinning reserves, reactive power supply, and voltage regulation Injecting reactive power at point of load (greater impact than having capacitors installed at substations)
- Local DER-level response supports the concept of agility and robustness, “Survivability” vs. eliminating contingencies

Impacts and Benefits

- Automatically adapting to new conditions, enables system to takes on “Self Healing” characteristics helping contingency response
- Decentralizing the intelligence allows the communication network to operate at a much slower speed
- Establishing interfaces to mainstream software technologies, and building on existing open standards helps to “future proof” the application
- Using an open architecture (common data models, open communication standards, and component-based development) reduces integration cost and increases overall interoperability

Impacts and Benefits

- Self-organization, self-description, and self-configuration based on semantic information of the agents contributes to grid resiliency
- Flexible interaction capabilities of MAS increase the ability to use market signals for dynamic resource scheduling
- Supports DOE Transmission Reliability (DER integration, Real-time Grid Reliability Management, Load as a Resource, and Smart, switchable network) programs

Infotility
More Power To You

Interactions/Collaborations

- Phase II work will involve collaboration of the following:
  - →Major distribution utility(s)
  - →RTOs and ISOs
  - →R&D Labs
  - →Equipment Vendors
  - →Consultants
  - →Testing Facilities

Contact Info

Principal Investigator

David Cohen
Infotility
CTO
Email: Dave@Infotility.com

Corporate Office

3003 Executive Parkway
San Ramon, CA 94583
Phone: 925-275-3185

Boulder Office

7136 Petursdale Ct.
Boulder, CO 80301
Mobile: **303-808-3329**