



Semantic Web Fred

Agent Cooperation on the Semantic Web

Michael Stollberg

ISWC 2004 Demonstration, 10 November 2004

Contents

1. SWF Framework
2. SWF Use Case
 - Setting
 - Resources
3. SWF Demonstration
 - GG Discovery
 - GS Discovery
 - WW Discovery
 - Meeting Execution

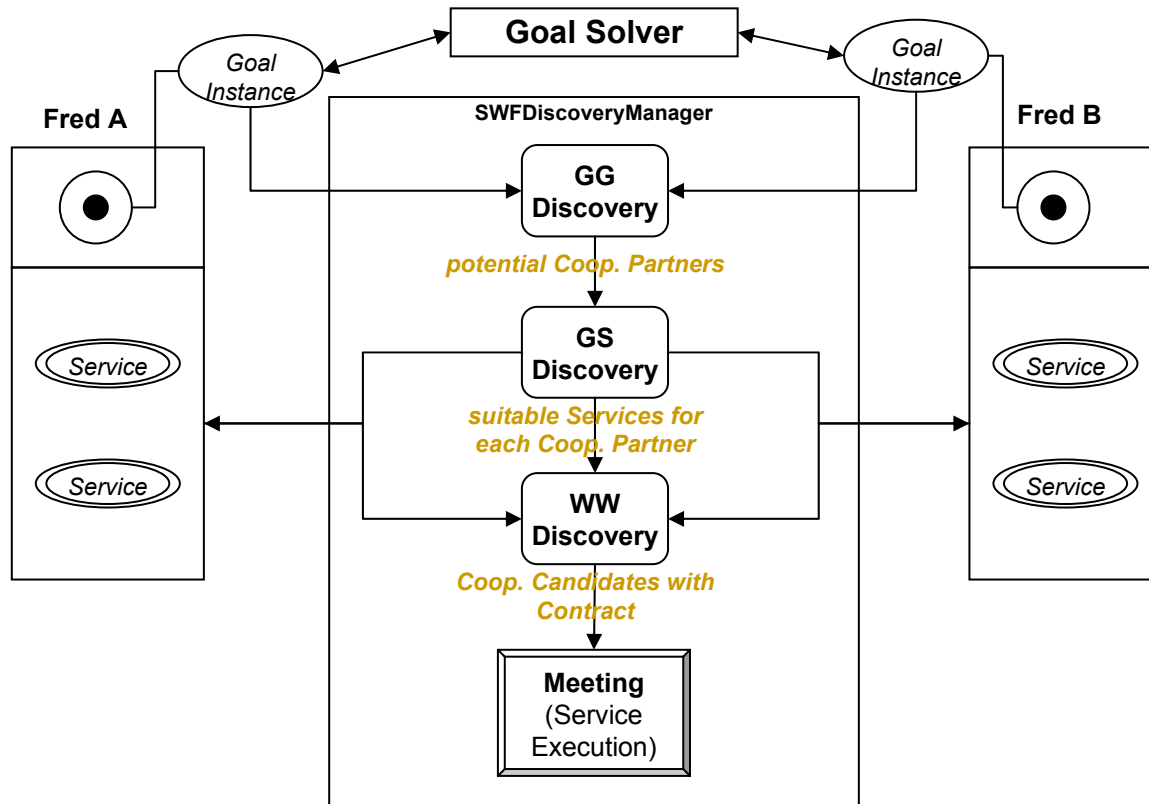


SWF Framework

SWF Framework

- An agent system
 - delegation of tasks to electronic representatives
 - automated resolution in agent cooperations
- Ontologies as data model
 - Smart Object technology for ontology management
 - support for Semantic Web ontology languages
- Semantic Technologies for dynamic cooperation partner and service detection
 - WSMO for describing resources
 - WSMO enabled discovery technologies

SWF Overview



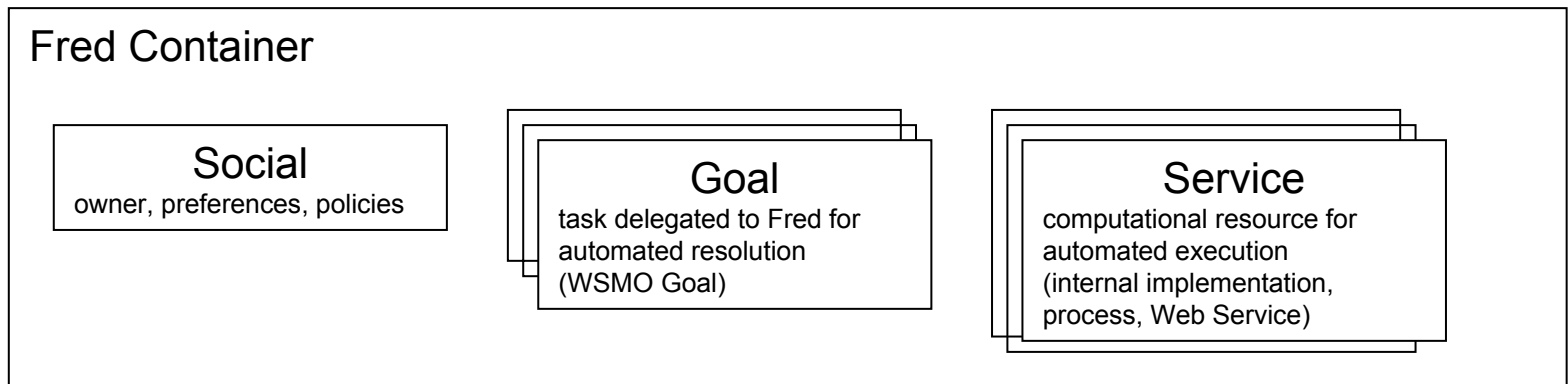
SWF Agents “Freds”

various definitions:

autonomous self-contained reactive pro-active
goal-orientated BDI (belief-desire-intention)
flexible temporal continuative adaptive
collaborative communicative

Freds

agents as electronic representatives for task delegation



FredBase

- Agent Runtime Environment
- Meeting Room
 - logical location where interactions between Freds take place
 - technical environment for service execution
 - monitoring, scheduling, and management of meetings
- FIPA ACL used for agent communication (i.e. services exchange messages via FIPA ACL communicative acts)
- Agent Management (Activation, Removal to Sleeping mode, etc.)

Ontologies

unambiguous definitions

conceptual model of a domain
(ontological theory)

formal, explicit specification of a shared conceptualization

machine-readability

commonly accepted
understanding

Smart Objects

- transformation of ontologies into Java Objects
- usage of conventional Java technologies for ontology handling
- expressiveness equivalent to OWL
- persistent storage & retrieval technology
- versioning and simple mapping techniques

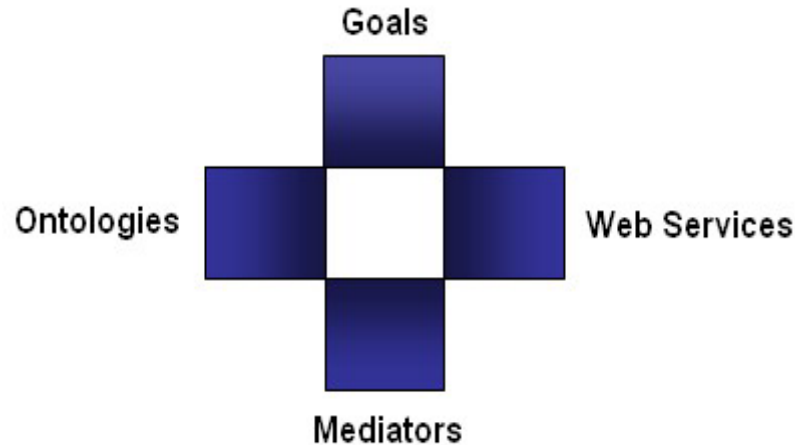
Ontology Usage in SWF

- Ontologies used as data (model & resources) throughout the whole system
- Smart Objects:
 - Expressiveness ~ OWL
 - “sufficient” ontology management
 - main benefit: ontology usage by conventional technologies
- Supported Ontology Languages
 - F Logic (imported from OntoEdit)
 - WSML
 - RDF
 - OWL partly

Semantic Web Services / WSMO

Objectives that a client may have when consulting a Web Service

Provide the formally specified terminology of the information used by all other components



Semantic description of Web Services:

- **Capability** (*functional*)
- **Interfaces** (*usage*)

Connectors between components with mediation facilities for handling heterogeneities

WSMO Components in SWF

Goal Template

*templates for desires to be delegated to Freds,
described as WSMO Goals*

nonFunctionalProperties **ofType** nonFunctionalProperties
importOntologies **ofTypeSet** ontology
usedMediators **ofTypeSet** {ooMediator, ggMediator}
postConditions **ofTypeSet** axiom
effects **ofTypeSet** axiom

WSMO Components in SWF

Goal Instance

concrete desires (instantiated Goal Templates) assigned to a Fred for automated resolution

instanceOf **ofType** goalTemplate

nonFunctionalProperties **ofType** nonFunctionalProperties,
timeConstraints, resourceConstraints, goalResolutionConstraints

owner **ofTypeSet** Fred-ID

submission **ofTypeSet** instance

postConditions **ofTypeSet** axiom

effects **ofTypeSet** axiom

status **ofType** {open, processing, solved, cancelled, not solved}

WSMO Components in SWF

Cooperative Goal

defines compatible Goal Templates (compatible on cooperation role and object of interest)

nonFunctionalProperties **ofType** nonFunctionalProperties
compatibleGoalGroup **ofTypeSet** goalTemplate
cooperativeGoalConstraints **ofTypeSet** ggMediator

Action Ontology

defines action compatibility for Goal Templates and Services

concept action

hasCompatibleAction **ofTypeSet** action

concept resource

hasAction **ofTypeSet** action

WSMO Components in SWF

Services

a computational resource available in the system, described as WSMO Web Services (Plans, Processes, external Semantic Web Services)

nonFunctionalProperties **ofType** wsNonFunctionalProperties

importOntologies **ofTypeSet** ontology

usedMediators **ofTypeSet** {ooMediator, wgMediator}

capability **ofType** capability

 preconditions **ofTypeSet** axiom

 assumptions **ofTypeSet** axiom

 postconditions **ofTypeSet** axiom

 effects **ofTypeSet** axiom

choreography **ofType** choreography

WSMO Components in SWF

Mediators

components connector with mediation facilities for resolving mismatches, described as WSMO Mediators (OO Mediator, GG Mediator, WG Mediator, WW Mediator)

nonFunctionalProperties **ofType** nonFunctionalProperties
importOntologies **ofTypeSet** ontology
source **ofTypeSet** {ontology, goal, webService, mediator}
target **ofType** {ontology, goal, webService, mediator}
mediationService **ofType** {goal, webService, wwMediator}



SWF Use Case

marketplace for purchasing furniture

- Marketplace Participants
 - Buyer
 - Seller
- Resources
 - Domain and Application Ontologies
 - Goal Templates and Instances for Marketplace Participants
 - Services for buyers and sellers

is realized in demonstration

complete use case definition available at: <http://swf.deri.org/usecase>

Funishing Ontology

This ontology describes the domain of furniture that can be found within rooms, and the relation between furniture and rooms; pieces of furniture are bought and sold by the participants of the virtual marketplace. The conceptual structure of this ontology is the following:

- taxonomy of furniture
- taxonomy of dwellings
- taxonomy of rooms
- taxonomy of relations between furniture and rooms

Marketplace Ontology

This ontology contains all notions concerned with buyers and sellers as participants of the virtual marketplace, products, purchasing, payment and delivery. The main building blocks of the Marketplace Ontology are:

- **marketplace participants:** defines buyers and seller with their contact / shipping / billing addresses
- **product:** describes products available in the marketplace
- **purchase contract:** defines a contract of purchase, which is the result of a successful purchase between a buyer and a seller
- **payment method:** specifies the payment methods accepted in the marketplace
- **delivery:** specifies types of delivery, i.e. how a buyer receives a purchased product

Location Ontology

This ontology describes locations and addresses with special regard to postal addresses in Austria (as the SWF virtual marketplace is settled in Austria). We use the Location Ontology as specified in [[WSMO Use Case](#)] with minor changes.

the main constructs of this ontology are:

- taxonomy of locations (country, state, city, etc.)
- postal addresses located in a location

Buyer Goal Template 1

- desire:** buying a single piece of furniture (furniture is the superconcept of all types of furniture), and get the furniture delivered to a ship address
- **postcondition:** a contract of purchase for a buyer for a specific piece of furniture, payment by creditcard
 - **effect:** delivery of the purchased piece of furniture to a specific shipping address in Austria

Buyer Goal Template 2

desire: purchasing a single piece of furniture, and get the furniture delivered (by drop ship or by self collection)

- **postcondition:** a contract of purchase for a buyer for a specific piece of furniture, payment by ANY PAYMENT METHOD
- **effect:** delivery of the purchased piece of furniture to a specific shipping address in Austria, or self collection by the buyer

Buyer Goal Template 3

desire: get detailed product information for a specific piece of furniture without purchasing

- **postcondition:** all information on a specific piece of furniture which can be purchased as a product from a company or store

- **effect:** none

Buyer Goal Template 4

desire: buy a piece of furniture from a private seller, i.e. not from a company or store

- **postcondition:** a contract of purchase for a buyer for a specific piece of furniture from a seller who is a private person, payment method unspecified (means: all payment methods are accepted)
- **effect:** the buyer collects the bought piece of furniture himself (self collection as special type of delivery)

Seller Goal Template 1

desire: sell a single piece of furniture, and deliver the furniture to the ship address of a buyer (in Austria only)

- **postcondition:** a contract of purchase for a seller for a specific piece of furniture, payment by any payment method
- **effect:** delivery of the purchased piece of furniture to the buyer's shipping address (restricted to Austria), or as self collection by the buyer

Seller Goal Template 2

desire: provide detailed information about products (pieces of furniture) available in a store

- **postcondition:** all information on a specific piece of furniture which can be purchased as a product from a company or store

- **effect:** none

Seller Goal Template 3

desire: a private seller sells a piece of furniture

- **postcondition:** a contract of purchase for a specific piece of furniture where the seller is a private person, payment method unspecified (means: all payment methods are accepted)
- **effect:** the buyer collects the bought piece of furniture himself (self collection)

Buyer Service 1

- **functionality:** buy a piece of furniture (i.e. receive a valid contract for purchase), and get the bought item delivered to a shipping address in Austria
- **SWF Service Type:** FRED Plan
- **capability**
 - **precondition:** required input is: [1] a piece of furniture to be bought, [2] buyer information (name, ship and bill address in Austria) , and [3] a payment method (types of payment methods are predefined in the ontology)
 - **assumption:** if payment method is credit card, the credit card has to be not expired
 - **postcondition:** a contract a purchase for the piece of furniture provided as input, with the buyer provided as input, and a payment method (all pre-defined payment methods are accepted)
 - **effect:** delivery of the purchased piece of furniture to the buyer's shipping address specified in the input, or via self collection by the buyer

Buyer Service 2

- **functionality:** gathers product information for pieces of furniture offered in the market, and returns them by email to the requester who is a buyer
- **comment:** for demonstration purpose, there is a suitable set instances of furniture defined which sellers can offer a (sub)set of); for a more advanced setting, this service has to (a) query the databases of stores in the market, and (b) collect all furniture offered for sale by private sellers.
- **SWF Service Type:** FRED Plan
- **capability**
 - **precondition:** required input is: [1] the piece of furniture that product information are requested for, and [2] buyer information (name, email-address)
 - **assumption:** the buyer has to be registered as marketplace participant
 - **postcondition:** returns product information for all piece of furniture available as sale offers in the marketplace which match the furniture description provided as input; sends this information as email to the requester (the buyer provided as input)
 - **effect:** none

Buyer Service 3

- **functionality:** buy a piece of furniture from a private seller with self-collection as the delivery method
- **SWF Service Type:** FRED Plan
- **Capabilit**
 - **precondition:** required input is: [1] the piece of furniture to be bought, [2] a buyer (name, bill address in austria)
 - **assumption:** there is at least 1 private seller in the marketplace who offers a piece of furniture that fits the desired furniture
 - **postcondition:** returns a contract of purchase between the buyer provided as input and a private seller for a piece of furniture that fits the desire; the accepted payment method are all those pre-defined in the ontology
 - **effect:** the delivery method for the bought piece of furniture is self-collection

Seller Service 1

- **functionality:** general service for selling a piece of furniture, returning a contract of purchase with delivery of the product as an effect (any type of delivery). All specific features can be defined by the service user (i.e. the seller that uses this Service).
- **provider / usage permission:**
 - marketplace owner / all 'Seller Freds'
 - specific services are realized in the demonstration for companies: IKEA, Leiner, Kika (furniture stores)
- **SWF Service Type:** FRED Process
- **capability**
 - **precondition:** required input is: [1] piece of furniture to be sold, [2] a seller (registered marketplace participant, located in Austria), [3] accepted payment methods are credit card or invoice
 - **assumption:** the desired piece of furniture has to be available as a product offered by the seller
 - **postcondition:** returns a contract of purchase for a piece of furniture that fits the desire between a buyer and the seller provided as input, with accepted payments credit card or invoice.
 - **effect:** delivery of bought piece of furniture by the a Delivery Service (not a Web Service), to buyer shipping addresses in Austria only; or the buyer gets the furniture by self-collection

Seller Service 2

- **functionality:** general service for providing detailed information on a piece of furniture that can be bought in the marketplace
- **comment:** this service is implemented as a simple search of the 'marketplace product repository' which holds all pieces of furniture that can be purchased in the marketplace: each seller signs in for those instances of furniture that he wants to sell; so, restricted search on furniture that are offered by a specific seller or a group of sellers (e.g. private sellers) can be performed. In a more elaborated version, this service searches (a) the product repositories of stores in the market, and (b) all furniture offered for sale by private sellers
- **provider / usage permission:** marketplace owner / all 'Seller Freds'
- **SWF Service Type:** FRED Plan
- **capability**
 - **precondition:** required input is: [1] the piece of furniture that product information are requested for, [2] the seller or type of seller (e.g. private seller only) whose offers to be searched for
 - **assumption:** if product information from a specific seller are requested, this seller has to be registered in the marketplace
 - **postcondition:** returns product information (incl. item, price, seller) for pieces of furniture that are existing as sales items in the market by the seller specified or all sellers of the seller type specified
 - **effect:** none

Seller Service 3

- **functionality:** general service for selling a piece of furniture; only private sellers are allowed to use this service. The service is provided by the marketplace owner as a generic "contract of purchase" service; the delivery type is always self-collection
- **provider / usage permission:** marketplace owner / all 'Seller Freds' that represent a private seller
- **SWF Service Type:** FRED Process
- **capability**
 - **precondition:** required input is: [1] piece of furniture to be sold , [2] a private seller (registered marketplace participant, located in Austria), [3] accepted payment methods all payment methods pre-defined in the ontology
 - **assumption:** [1] the desired piece of furniture has to be available in the marketplace, [2] if the payment method is credit card, then the credit card has to be valid (i.e. not expired)
 - **postcondition:** returns a contract of purchase for a piece of furniture that fits the desire between the buyer and the private seller provided as input, with the payment method provided as input
 - **effect:** the buyer gets the furniture by self-collection

Action Ontologies

- Cooperative Knowledge Ontology
 - defines groups of compatible Goal Templates
 - used as preselection in GG Discoverer
- Service Ontology
 - defines action compatibility between Goal Templates and Services
 - used as preselection in GSDiscoverer

preliminary versions for an integrated action ontology

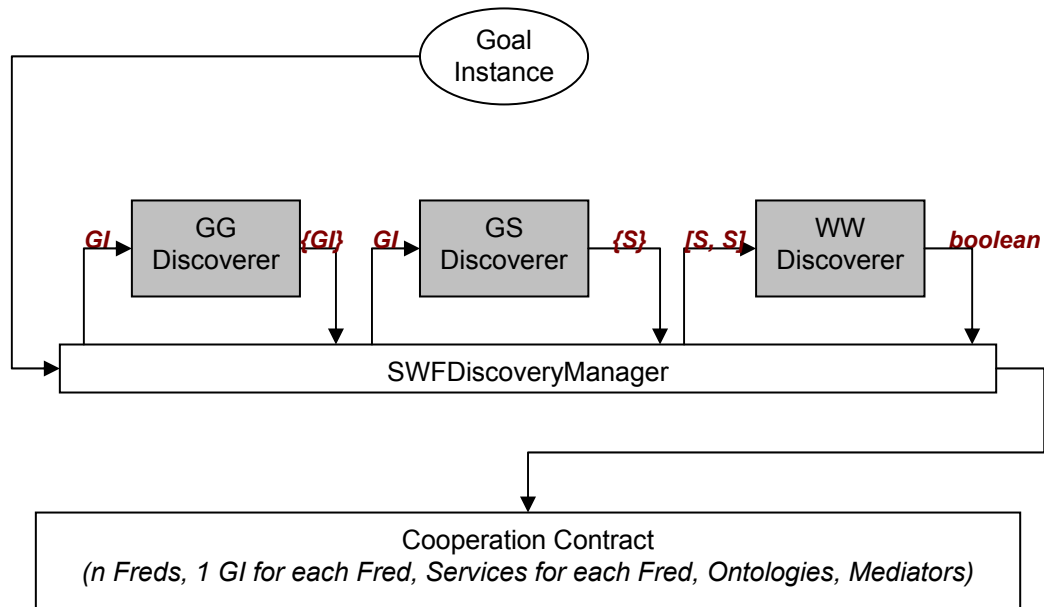


SWF Demonstration

Overview

- What is final aim:
 - a User Interface for:
 - editing Freds (Agents as electronic representatives)
 - creation and assignment of Goal Instances
 - monitoring of resolution process
 - Cooperative Goal Resolution Process completely automated
- What we show & why:
 - selection of Freds with open Goal Instances
=> finally done by a Selection Engine
 - Walk Thru of Cooperation Establishment & Execution
=> finally done by SWF Discovery Manager

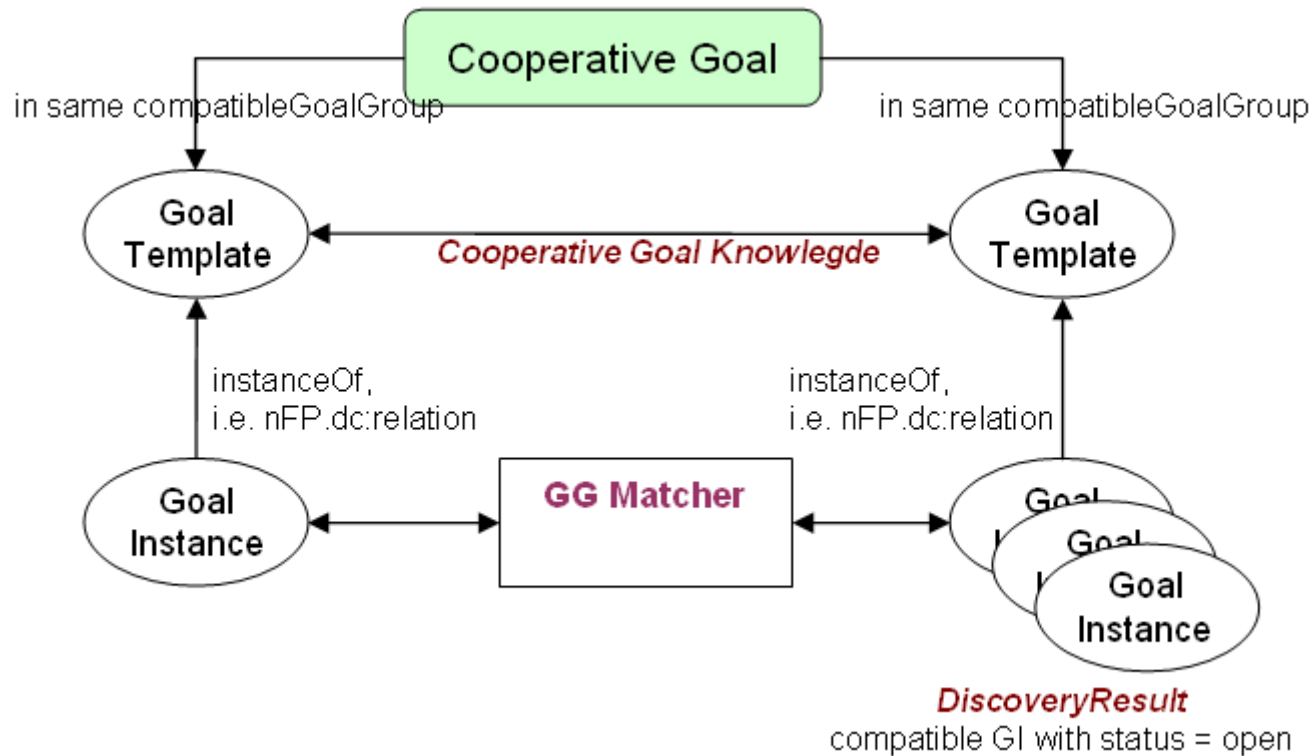
SWF Discovery Manager



1. Select a Fred-Agent

- This is an agent of some user
 - has a open Goal Instance assigned
 - discovered by Selection Engine (**in automated version**)
- A Fred has:
 - owner
 - a Goal Instance (concrete desire + submission)
 - usage permission for Services
- The open Goal Instance of this Fred represents the initiator of the Goal Resolution process

2. GG Discoverer



Cooperative Knowledge Filter

Results for Use Case

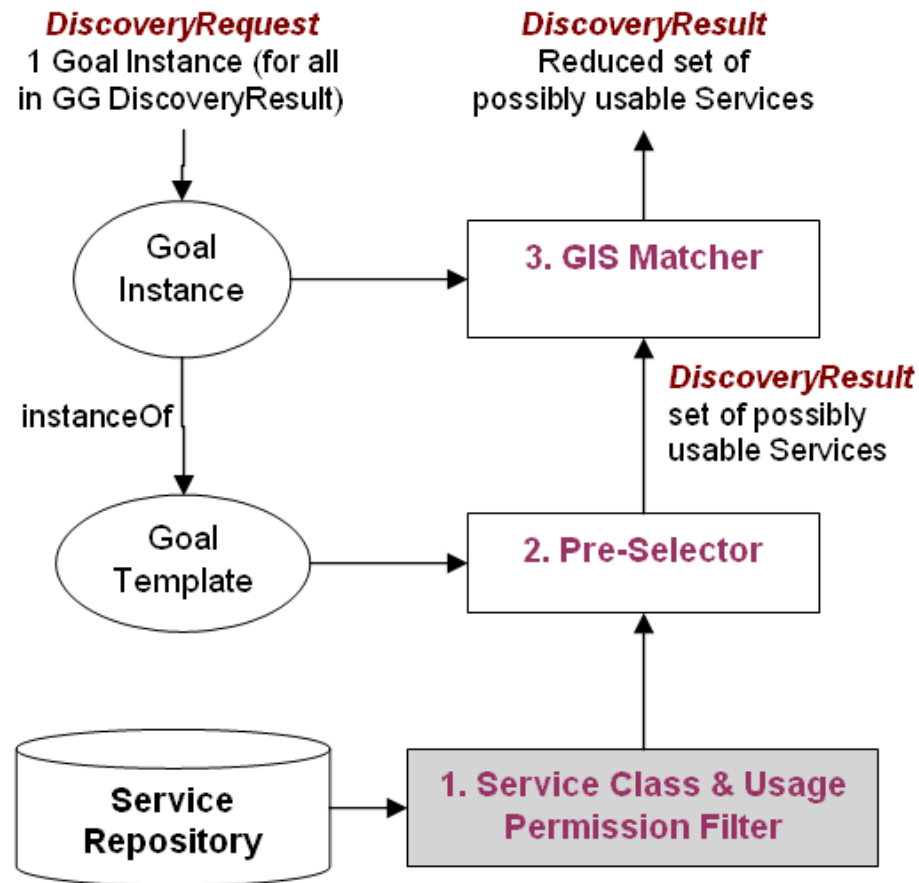
	BGT1	BGT2	BGT3	BGT4	SGT1	SGT2	SGT3
BGT1					X		
BGT2					X		X
BGT3						X	
BGT4							X
SGT1	X	X					
SGT2			X				
SGT3		X		X			

GG Matcher

Results for Use Case

	111	112	113	114	121	1212	1213	122	123	1232
111					X	X	X			
112					X	X	X		X	
113								X		
114										X
121	X	X								
1212	X	X								
1213	X	X								
122			X							
123		X								
1232				X						

3. GS Discoverer



GS Pre-Selector

Results for Use Case

	BGT1	BGT2	BGT3	BGT4	SGT1	SGT2	SGT3
Buyerservice1	X	X		X			
Buyerservice2			X				
Buyerservice3		X		X			
Sellerservice1					X		
Sellerservice2						X	
Sellerservice3							X
Sellerservice IKEA					X		
Sellerservice LEINER					X		
Sellerservice KIKA					X		

GG Matcher

Results for Use Case

	111	112	113	114	121	1212	1213	122	123	1232
111					X	X	X			
112					X	X	X		X	
113								X		
114										X
121	X	X								
1212	X	X								
1213	X	X								
122			X							
123		X								
1232				X						

4. WW Discoverer

- aim: determine compatibility of services with regard to their interaction behavior
- for all possible combinations of services for potential cooperation partners
- choreography checked on basis of compatibility patterns
- not realized yet

5. Meeting Execution

- After cooperation establishment, the Freds along with the discovered resources are sent into a ‘meeting’
- The SWF Meeting Room Manager invokes the services and monitors, controls the execution
 - services (Plans) communicate via FIPA ACL constructs
 - Meeting Room technology currently for FRED internal usage only
- after a successful meeting:
 - the service execution results are handed to the Freds / their owner
=> here: create PDF or “product not available” message
 - the Goal Instances of the participating Freds are solved
 - the Freds are released from the meeting and can take over the next task