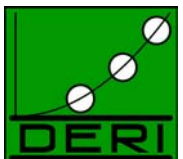
Semantic Web Fred:

Goal and Service Description Language

Michael Stollberg

- 05 June 2004 -

Net Dynamics

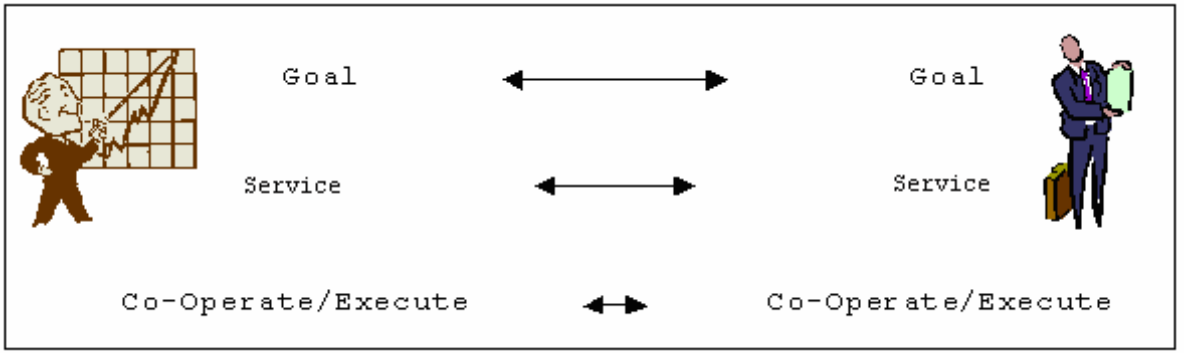


Content

1. Recall: SWF Aims & Architecture
2. Description Language
   - Goals
   - SWF Services
3. Usage in SWF Mechanisms
4. Summary

Cooperation Model

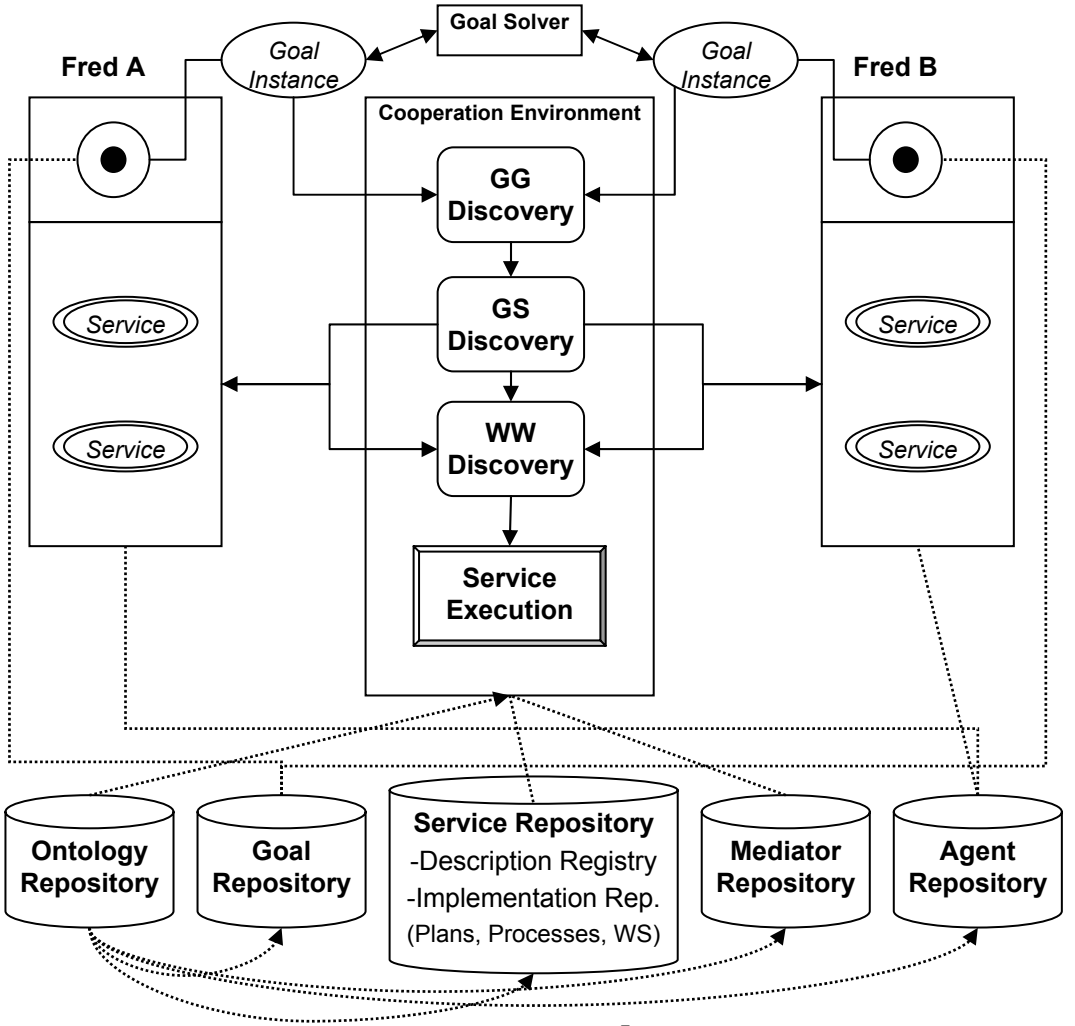
Aim: Map real world Cooperation Model into Software



=> **Symmetry** of cooperating parties:

- Why:
  - Cooperative Goals can only be achieved by cooperation of several parties (e.g. the CEO needs salesman for increasing company's success, salesman needs CEO for increase sale rate)
  - Every party is requester and provider at the same time
- Implication for System Architecture: **all potential partners need to have Goals and Services**

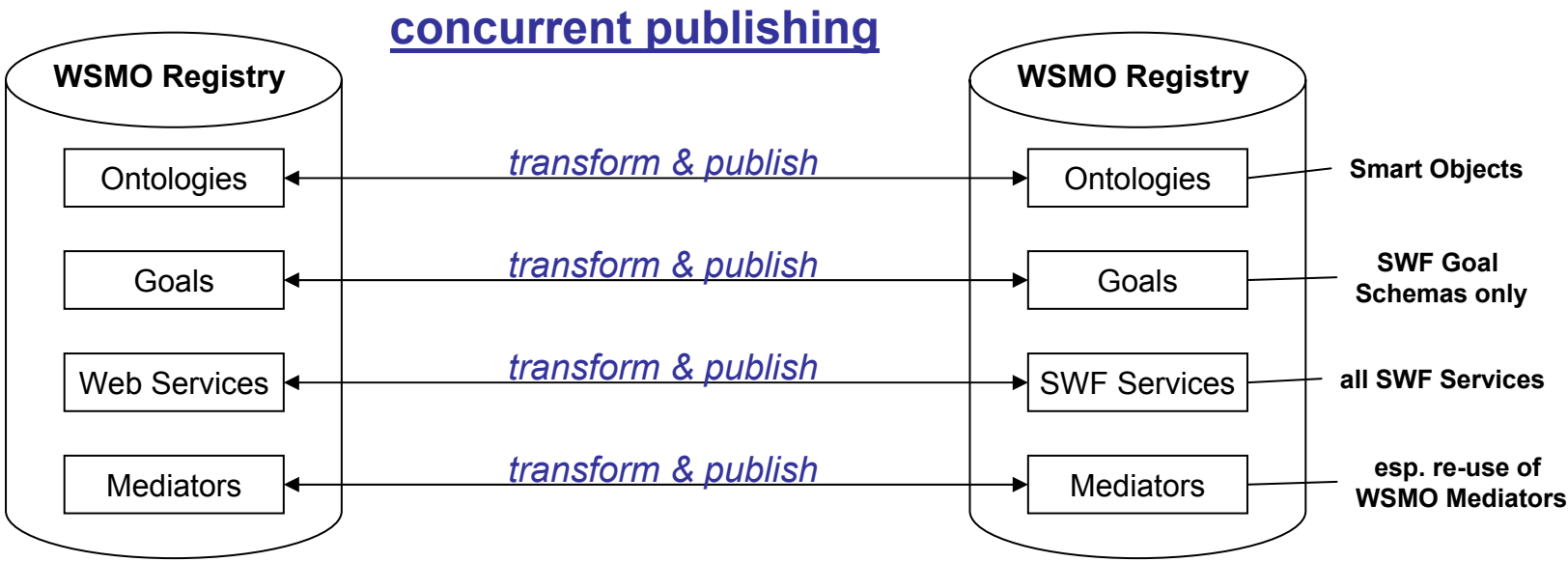
SWF Architecture



– Recall: SWF Aims & Architecture –

WSMO - SWF Repository Alignment

all SWF Goals / Web Services / Mediators, etc also in WSMO Repository & vice versa



- Repository structure identical, see WSMO D10
- transformation between descriptions needed

=> create a nice repository of use case data

Aims & Objectives

- **We only need semantic descriptions for Goals & SWF Services**
  - Ontologies are handled by SMO technology, also external ontologies (e.g. from WSMO) are transformed into SMOs => internal handling
  - Mediators:
    - "connection facility" is transformed in SWF technology
    - "mediation facility" is a normal Web Service / SWF Service
  - Agents (Freds) do not need semantic descriptions

- **Aim: SWF as a WSMO implementation**
  - General structure very aligned to WSMO

  => SWF Goal & Service Description Language is a specialization of WSMO

- **Language:**
  - For modeling we use WSML (as in WSMO Use Case deliverable)
  - Will be transformed in to technology-depend format

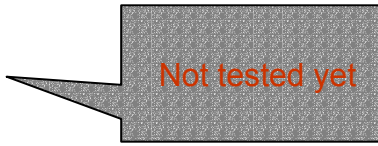
Goals

- Understanding of Goals:
  - express a desire that user delegates to a Fred for automated resolution
  - currently modeled as the "counterpart" for SWF Service Capability
    - expresses a desire
    - is the entity that actively interacts with a SWF Service => holds additional infos
- Cooperative Goals
  - a Goal that can be achieved in cooperation only (e.g. "Purchase")
  - specifies compatible "normal" Goals that Freds may have in a cooperation (e.g. "Buyer" and "Seller" Goals as compatible Goals of "Purchase")
  - change to previous version:
    - Goal in SWF = specialization of WSMO Goal
    - Cooperative Goal = Goals only resolvable in a cooperation
- Goal Schema and Goal Instance
  - Goal Schema = ontological structure of Goals resolvable by the system
  - Goal Instance = concrete expression of desire "thrown" during runtime

Under discussion really the right way?

Goal Description

Goal Schema Description Elements



- **nonFunctionalProperties**: based on WSMO Core Properties. mandatory: *title, creator, description, date, time, identifier, rights, version*. Important is the *title* and *description* (used by users to identify and select Goal Schemas).
- **usedMediators**: OO Mediators (terminology)
- **requirementSpec**: the ontological schema that is handed over to a service as input to that service during invocation. Is "instantiated" by a corresponding Goal Instance.
- **resultSpec**: the ontological schema that is expected to solve the Goal. Is "instantiated" by a corresponding Goal Instance. The resolution of a Goal might require usage of several services / several cooperations.
- **postCondition**: conditions on the resultSpec, i.e. the constraints that define desired state of the information space. If this holds after Goal Resolution, the Goal is solved. The postcondition describes conditions of resultSpecs in relation to requirementSpecs.
- **effects**: arbitrary states of the world that are expected to hold after the Goal Resolution.

Goal Description

Goal Instance Description Elements

Not tested yet

- **instanceOf:** link to the Goal Schema that the Goal Instance is derive from
- **nonFunctionalProperties**: based on WSMO Core Properties. mandatory: *creator, date, time, identifier, rights,* derived from Goal Schema: *title, description, version*. Additionally:
  - timeConstraints: timeframe in which the Goal shall be resolved
  - resourceConstraints: constraints on possible partners or services to be used
  - goalResolutionContraints: "user preferences" on the whole Goal Resolution process
- **owner**: the Fred that has created this Goal Instance
- **requirements**: instantiated ontological structure that is handed over to a service as input during service invocation.
- **results**: instantiated ontological structure that is expected as a result of service(s) usage to solve the Goal.
- **status**: information on the Goal Resolution process, handled by the Goal Solver; possible values: open, solved, not solved, processing, cancelled

Goal Description

Cooperative Goal Description Elements

```
cooperativeGoal:[
 nonFunctionalProperties => nonFunctionalProperties
 compatibleGoals[
   compatibleGoal =>> goalSchema
   compatibleGoalConstraints =>> axiomDefinition
 ]
cooperativeGoalConstraints =>> axiomDefinition
]
```

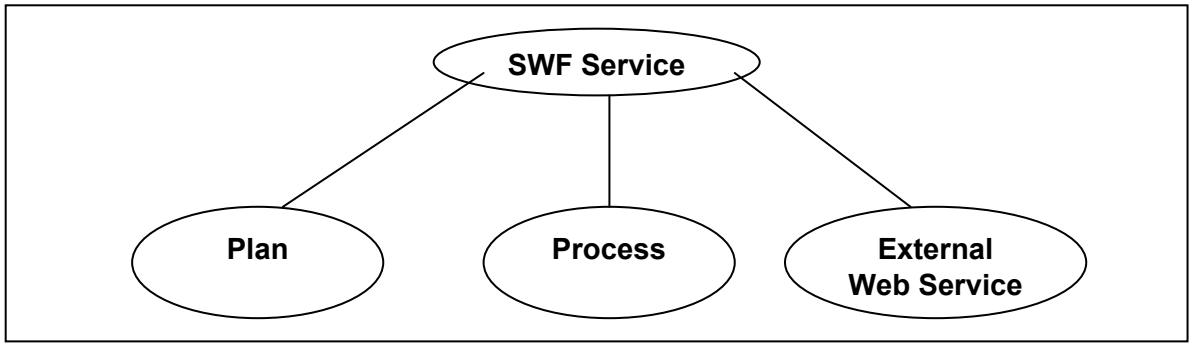
- **compatibleGoals**: the set of Goal Schemas which in interaction support the solution of the Cooperative Goal.
- **compatibleGoalConstraints**: constraints defined for the Goal Schema,

  e.g.: cardinality: 0-n means that there can interact 0, 1, or up to n partners with this Goal. Defining the Cooperative Goal "Playing Football" might have the sub-Goal "Providing a Team" with cardinality 2, while "Providing a team" might have "Being a team member" with cardinality 11.
- **cooperativeGoalConstraints**: constraints defined across Goal Schemas.

this is a Redcution => maybe a GG Mediator here

Services

SWF Service Model – 3 Services types:



1. Plan = internal Java program (need: JVM)
2. Process = multi-step service, each activity can be resolved arbitrarily (needs: Process Engine)
3. external Web Services (execution via WSDLExecutorPlan)

=> Aim: **1 common Description Language for all SWF Services**

– for discovery we only need the descriptive information

– the Service Type is only of interest for Execution (different grounding)

Service Description

SWF Service Description Elements

Not tested yet

- **nonFunctionalProperties**: based on WSMO Core Properties. mandatory: *title, creator, description, date, time, identifier, rights, version*. WSMO Web Service specific nFPs are needed for discovery heuristics, but not incorporated in current version
- **usedMediators**: OO Mediators (terminology)
- **capability**: functional description, i.e. WHAT the service does
- **interface**: behavioral description, i.e. HOW to USE the service
- **grounding**: access (point to physical location) and Service Type

Service Description

SWF Capability Description Elements

- **precondition**: input required by the service & conditions over this
- **assumption**: arbitrary constraint of the state of the world that has to hold before the service can be executed

- **postcondition**: (computational) result of the service & conditions over this, in relation to the input
- **effect**: arbitrary constraint of the state of the world that results as a change after execution of the service

=> currently: 100 % WSMO compliant

different proposal under discussion:

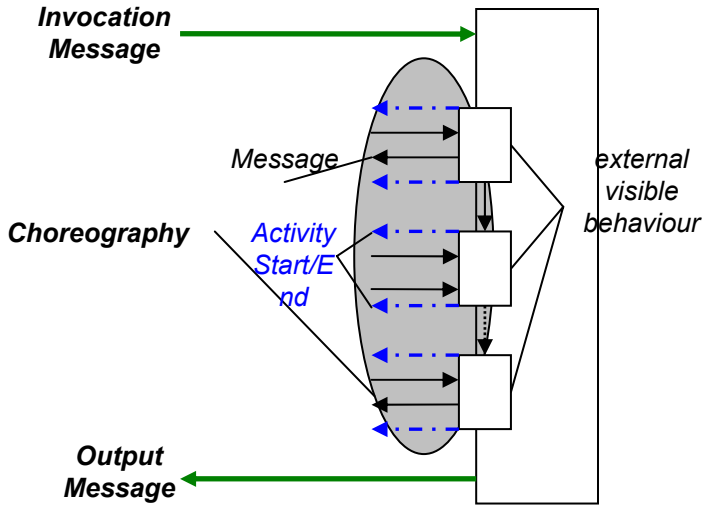
1) **Input** = ontological structure required
2) **precondition** = conditions over 1), i.e. integrity constraint
3) **assumption** = arbitrary constraints on state of the world

4) **Result** = ontological structure returned
5) **postcondition** = conditions over 4), i.e. integrity constraint
6) **effects**= arbitrary constraints on state of the world

Service Description

SWF Service Interface Description Elements

```
swfServiceInterface:[
  invocationMessage => invocationMessage
  outputMessage => outputMessage
  choreography => choreographyDescription
]
```



- **invocation**: Goal send input
  - must be compliant to Goal requirements
  - must satisfy the Capability precondition
- **outputMessage**: always the last message of a service behavior is a outgoing message (content: result or failure)
- **choreography**: WSMO Choreography description

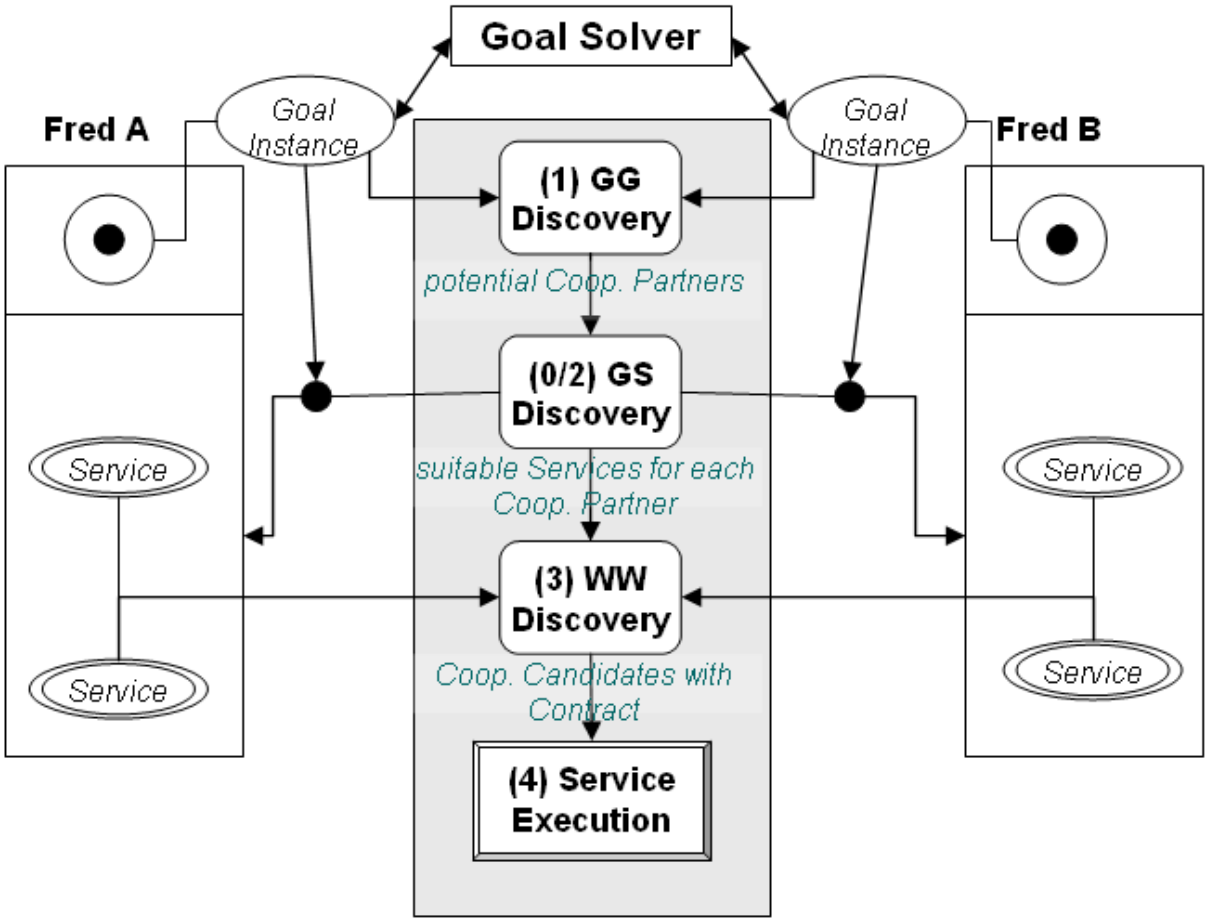
not specified in WSMO yet

(we only need the individual behavior description here – the "global interaction model" is located in WW Discovery)

Orchestration: handled by FRED Processes, not needed / regarded in SWF

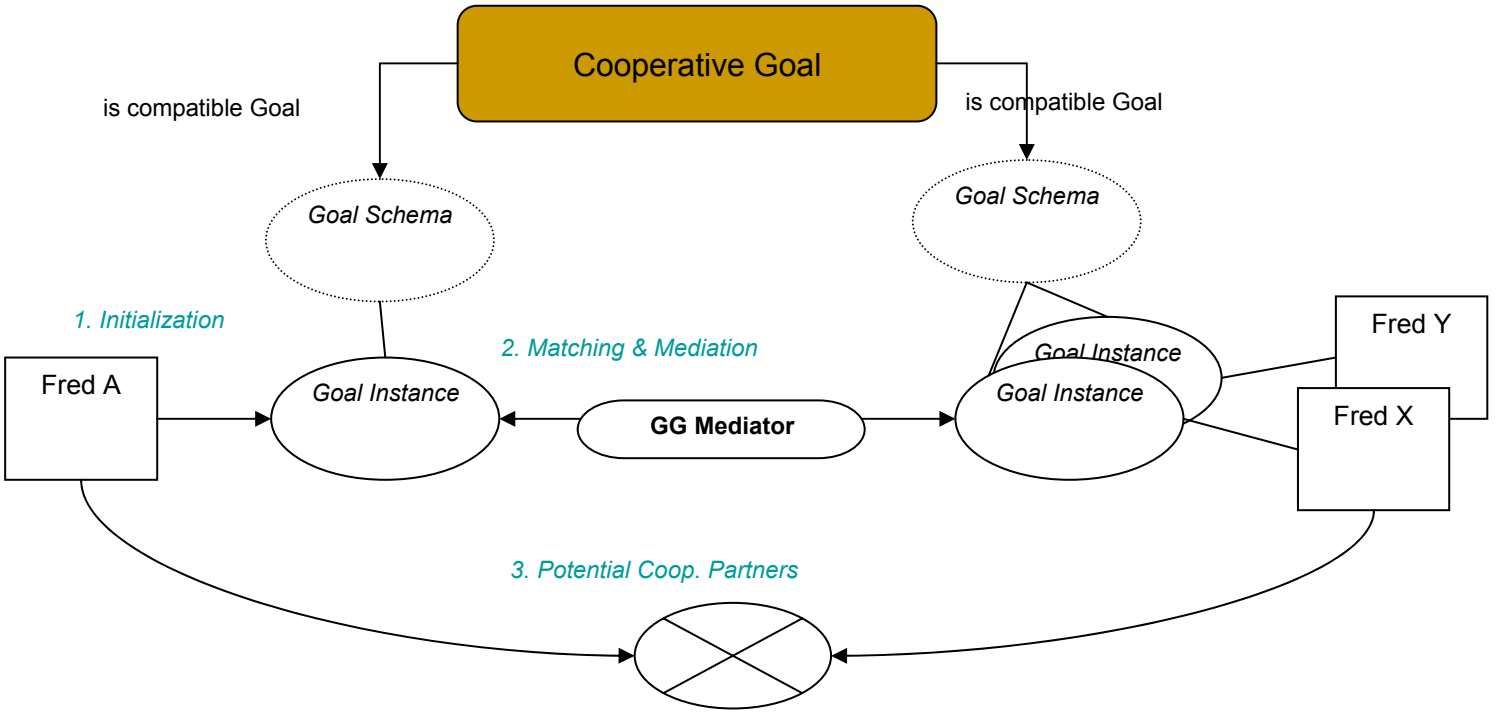
(orchestration & service composition is objective in FRESCO)

Automated Cooperation Workflow



GG Discovery

Detection of potential cooperation partners



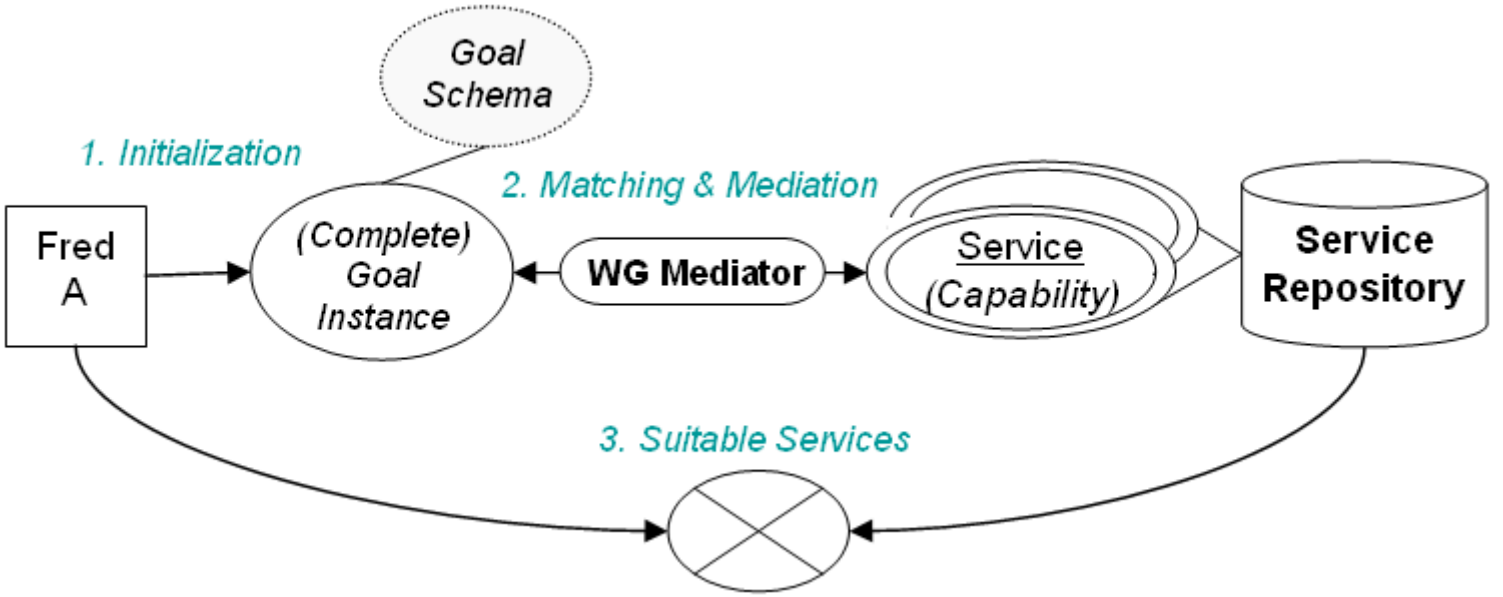
GG Discovery Workflow

- What to match
  - **Object of Interest** (to be the same)
  - **Cooperation Role** (to be compatible)
- Cooperative Goal specifies this compatibility completely
  - matching of **Object of Interests** by Cooperative Goal designer
  - **Cooperation Roles Compatibility**:
    - specifying compatible Goal Schemas
    - Owners of corresponding Goal Instances will have compatible Roles
- Functional aspects:
  - initiates Cooperation
  - can be performed at design time
  - different engines that initiate GG Discovery
    - permanent
    - event-driven
    - during runtime

GS Discovery

Detection of suitable SWF Services for each partner

(equivalent to Service Discovery in WSMO)



GS Discovery Workflow

- What to match:

  SWF Service Capability postcondition & effect have to "match" **Goal Instance result** and well as the corresponding **Goal Schema postcondition and effects**

- Algorithm = WSMO Discovery

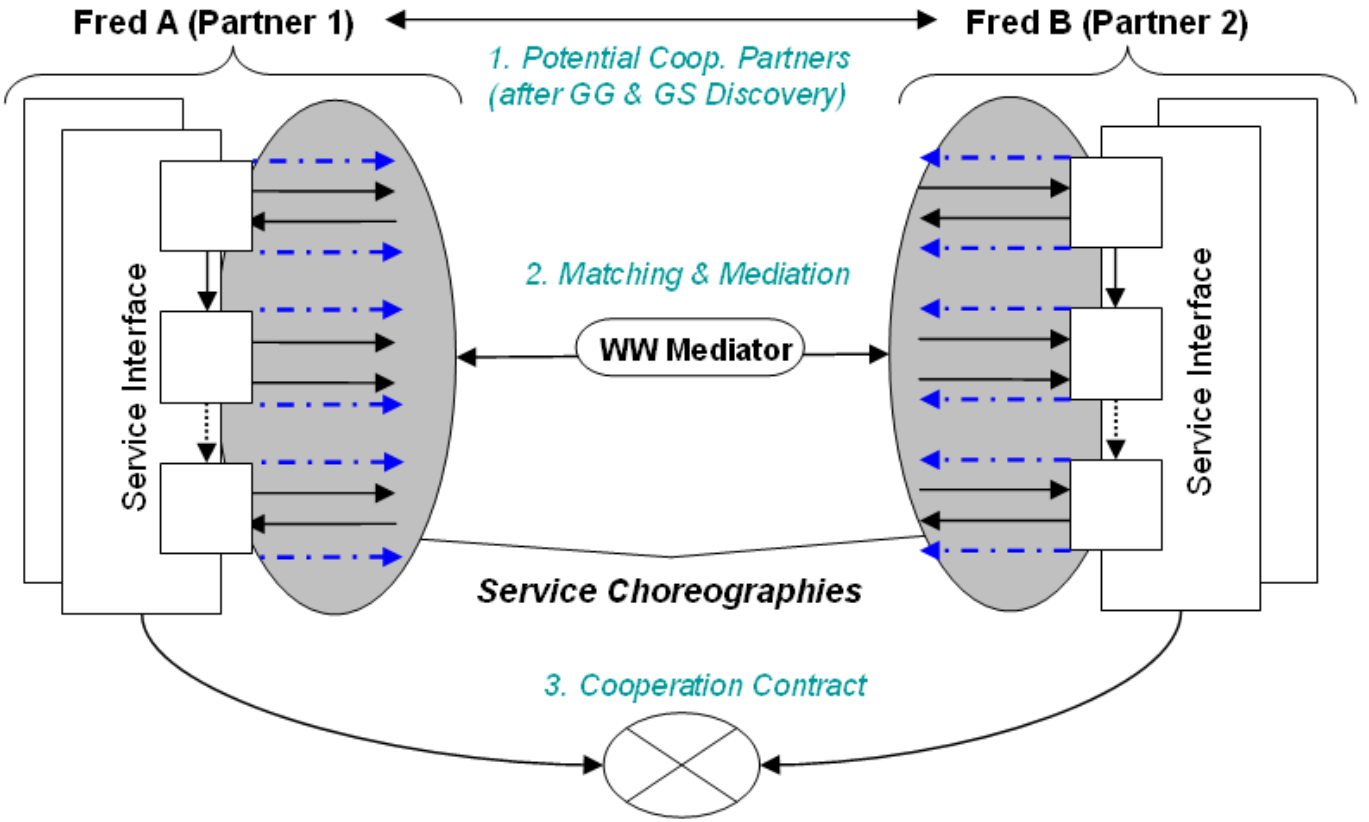
- Functional aspects:
  - pre-GS-Discovery at design time possible (indexing)
  - result: a set of possible usable SWF Service for each partner

Remark: there has to be at least 1 SWF Service that matches the Goals of each partner

(Service Composition tackled in FRESCO, currently manual as processes)

WW Discovery

Determine compatibility of detected Services for the Cooperation Partners
(pre-requisite for automated Service Execution)



– Outlook: SWF Discovery Mechanisms –

WW Discovery Workflow

- What to match:

  determine / establish **behavioral compatibility**,

  i.e. determine the **global service interaction model**

- Algorithm – ideas:
  - individual behaviors must be *inverse* (simplest case)
  - determine compatibility of MEPs
  - also: messaging technology has to be the same

- Functional aspects:

  determines the Cooperation Contract

  - specifies all information on Freds, Goals, Services, Service Interaction
  - is the "Service Execution Agenda"

– Summary –

Open Points

- Conceptual Decisions on Goal & Service Description Language
  - Goal "requirements"
  - Service Capability & Interface Description

  => *Test & Refinement in Use Case modelling*

- Basis Technology for SWF Mechanisms:
  - **reasoner** –vs- **conventional technologies**:
    - reasoner not stable & performant
    - conventional technologies need transformations everywhere, loss of flexibility (still preferred at this point in time)
  - important decision ...

- Matchmaker Specification – how shall they really work?

- Usage of WSMO Mediators – translation of connection facility

Future Work

- detailed system design
- Use Case Implementation
- supported environments:
    1) FRED environment (Meeting rooms, FIPA, Cooperation Contract)
    2) Web environment (virtual meeting rooms, SOAP, virtual cooperation contract)

    but: the core will be the same

and: Dissemination …

</ Semantic Web Fred Goal & Service Description Language>

Net Dynamics