Agent-Based Dynamic Engineering Design Process Modeling Framework

Internal Report No: TR-PSI-1-2004
February 2004

Authors:
Vadim Ermolayev[1] (editor)
Eyck Jentzsch[2], Wolf Matzke[2], Juergen Schmidt[2], Guenther Schroeder[2],
Stephan Weber[2], Jens Werner[2]

[1]Department of IT, Faculty of Mathematics, Zaporozhye State University
66 Zhukovskogo st. 69063 Zaporozhye, Ukraine

[2]Cadence Design Systems GmbH
Mozartstr. 2 D.85622 Feldkirchen, Germany

Abstract. This document specifies the generic modeling framework for the simulation of the Dynamic Engineering Design Processes (DEDP) in the domain of Semiconductor and Electronic Systems (SES) Design. The idea is to leave design activities to human designers and to focus on the proper means to achieve the optimal performance in DEDPs. That is why the agents mentioned throughout the report are of two major types: personal assistants providing people with the automated support of their routine functions and utility agents providing specific functionalities for DEDP-MAS as a whole. The framework is grounded on the agent-based Task-Oriented DEDP Model in which an agent models an actor as the locus of welfare accumulation. It is also stated that the more effective and efficient mechanisms for cooperative design may be obtained through rational collaboration among such actors – namely several types of negotiation and coalition formation. The framework assumes that the shared knowledge and terminology representation for collaborative activities among agents is structured and represented in the form of ontologies.

Contents

List of Figures

List of Acronyms

ACL	Agent Communication Language
CfP	Call for Proposals
CNP	Contract Net Protocol
CoA	Coordination Agent
D	Designer
DAML-S	DARPA Agent Mark-up Language – Services. A DAML ontology for specifying Semantic Web Services
DEDP	Dynamic Engineering Design Process
DEDP-MAS	Multi-Agent System for Modeling Dynamic Engineering Design Processes
DRP	Design Re-use Provider
DS	Design Solution
DSP	Design Support Provider
FB	Functional Block
FCEM	Fellow Capabilities' Expectation Matrix
FIPA	Foundation for Intelligent Physical Agents
FP	Functional Point
FSD	Functional Specification Design
I	Initiator of negotiation
IC	Integrated Circuit
KSLOC	Kilo Source Lines of Code
LARKS	Language for Advertisement and Request for Knowledge Sharing
LS	Logics Synthesis
MAS	Multi-Agent System
OA	Ontology Agent
OWL	Web Ontology Language
P	Negotiation participant
PhL	Physical Layout
PLP	Partial Local Plan
PM	Project Manager
PMI	Project Management Institute
SPA	Service Providing Agent
ST	Software Tool
TO	Tape Out
UDDI	Universal Description Discovery and Integration (language)
UoW	Units of Welfare
VHDL	Very high-speed Hardware Description Language

1 Introduction

The purpose of this report is to specify the initial draft of the generic modeling framework for the simulation of the Dynamic Engineering Design Processes (DEDP) in the domain of Semiconductor and Electronic Systems (SES) Design. The report is structured as follows. Section 1 describes the organization of a DEDP, points out to the factors which make a DEDP a highly dynamic process, and introduces the Task-Oriented DEDP Model. Section 3 introduces the idea of DEDP productivity metrics based on the interpretation of an actor within a DEDP as the locus of welfare accumulation. Section 3 also specifies the mechanisms for rational collaboration among such actors – namely several types of negotiation, coalition formation as well as the framework for adjusting actors' behaviors and strategies through monitoring the capabilities and credibility factors of their fellows. Section 4 drafts out the requirements for the coordination and communication among the agents, which actually model actors in DEDPs. Section 5 deals with the shared conceptual knowledge representation and structuring. While the ontologies maintained by the system Ontology Agent specify the core knowledge of the system providing the semantic frame for the consensual conceptualizations, the local knowledge of each member-agent may differ from the core set in the sense that the local knowledge is the instantiation and the more detailed knowledge specification within the core knowledge frame. Section 6 contains the concluding remarks.

2 A Model of a Dynamic Engineering Design Process

A DEDP is the kind of a workflow which is defined in frame of the in-house technological and management routines. Though the specific content of the DEDP steps may vary from company to company, there are still many common features shaping out a (generic) DEDP structure (e.g., Fig. 1a, 1b for Digital IC DEDP). A DEDP starts with the formulation of the conceptual idea for the future design. The idea goes further on through the step of Functional Decomposition which results in the set of concurrent design threads of a similar structure for each functional block extracted. Several consecutive DEDP steps take place for each functional block: Functional Specification, Logics Synthesis, Physical Layout, Prototype Manufacturing, etc.

Each of the steps comprises design verification procedure to check different aspects: logics equivalence, signal timing, physical design rules, electrical rules. Verification results, if unsatisfactory, may require the reiteration of one or even several previous DEDP steps. More details are shown at Fig. 1a and 1b.

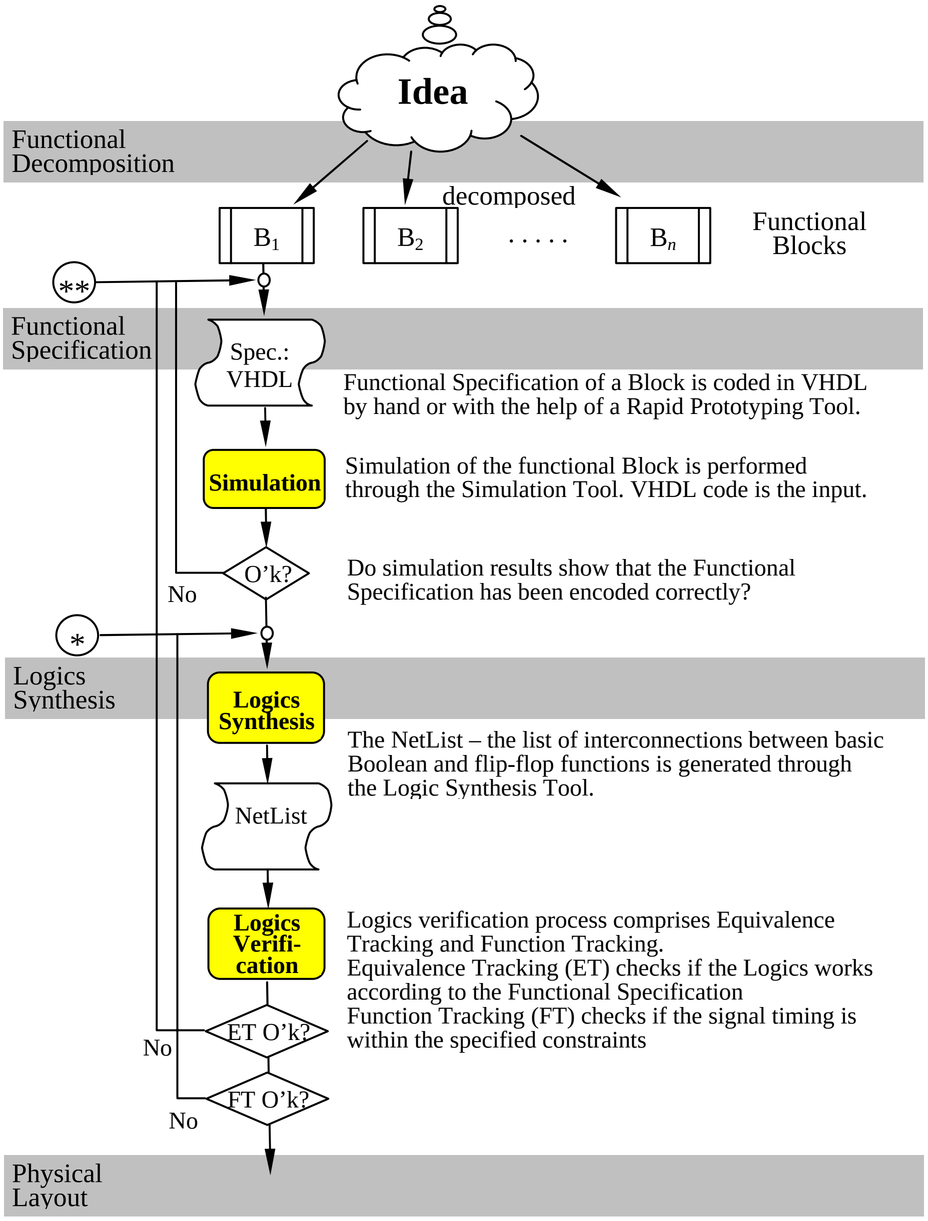


Fig. 1a. The Steps of an Engineering Design Flow for Digital Integrated Circuit (IC).

2.1 What brings dynamics to a DEDP?

Though a DEDP is a well-defined engineering flow, there are still some factors that make it impossible to plan or to define this flow in all details before it actually starts.

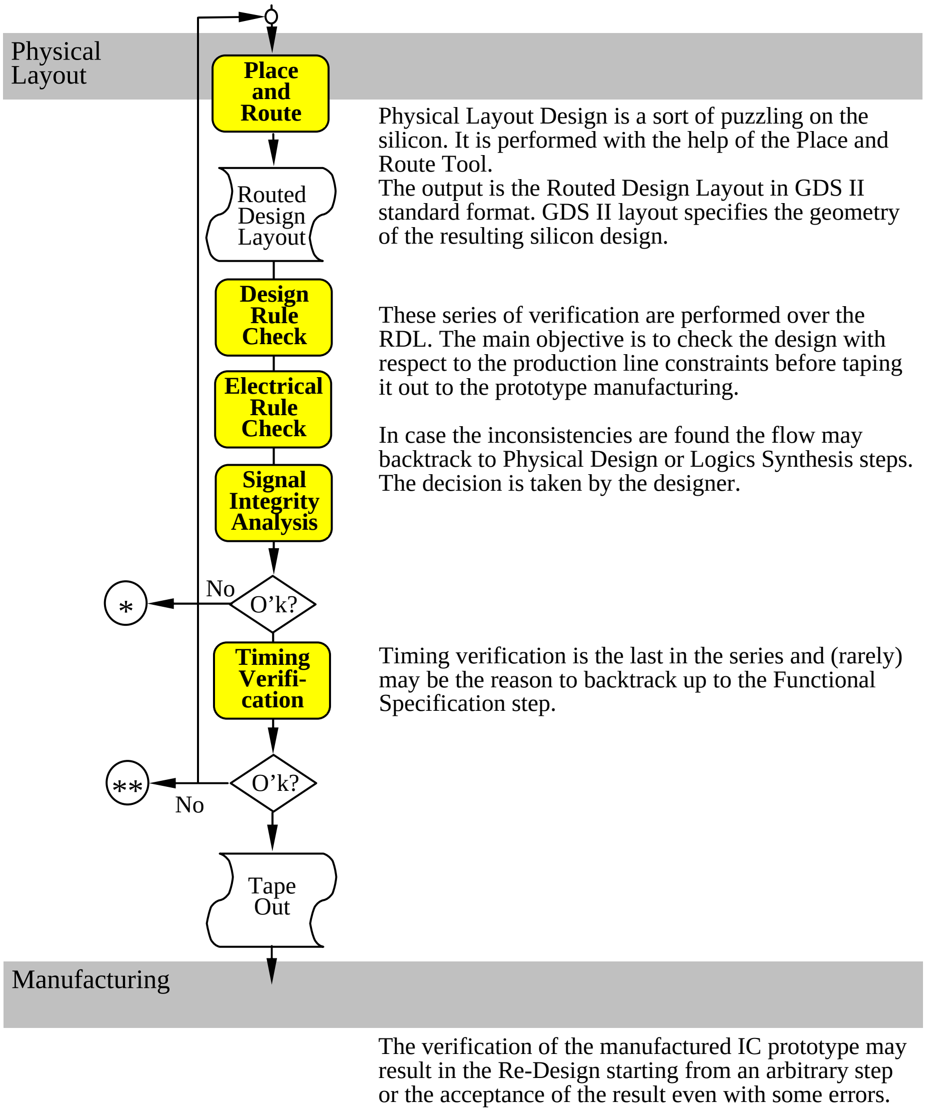


Fig. 1b. The Steps of an Engineering Design Flow for Digital IC.

These make such a flow dynamic. The factors are:

- **Functional decomposition**. As the conceptual idea of a design may be decomposed into the functional blocks in different ways by different designers it is impossible to define the concurrent threads of this DEDP in advance
- **Altering capabilities**. As the capabilities (the workload and the experience) of the designers change in time it is not well clear how to plan the optimal configuration of the flow performers with respect to the accomplishment time, the quality, and the cost of service.

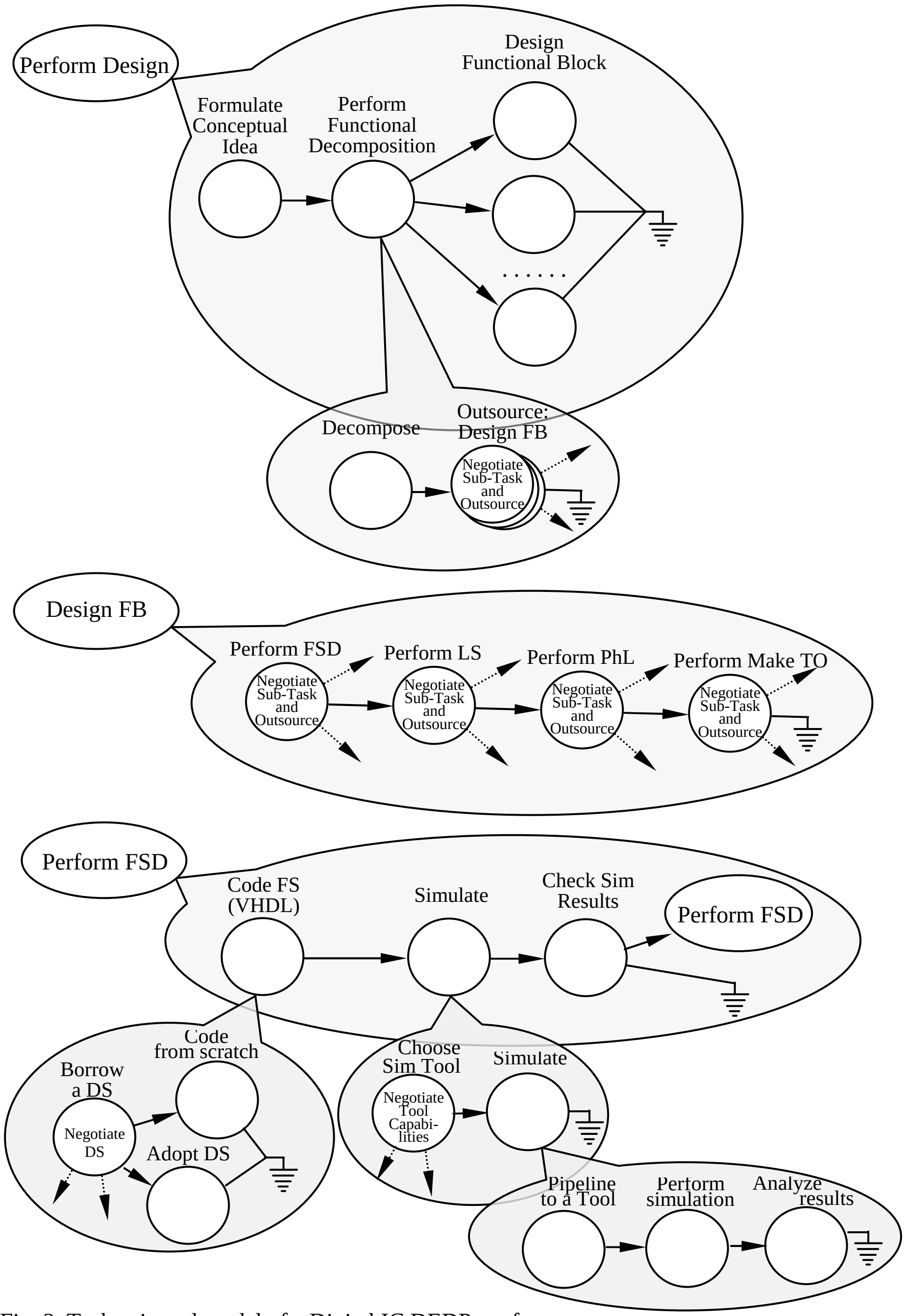


Fig. 2. Task-oriented model of a Digital IC DEDP – a fragment.

- **Design Solution reuse**. As the designs are often re-used or adopted from the other designs the major technological design steps may vary out of this

- **Backtrack loops**. As by the result of the verification at any DEDP step it might be necessary to backtrack to one of the previous steps, it is impossible to plan the number of such loops in advance

- **Tool choice**. As the characteristics of the different tools which may be used at a specific design step vary with respect to both productivity and the working experience a designer has in using these tools, it is hard to predict in advance which tool will be optimal for the step, and which one will be actually chosen by the designer. The choice of a tool may of course influence the resulting number of design iterations at this DEDP step.

These factors point to the necessity to take the decisions on the configuration of a DEDP "on the fly", in line with its actual execution, each time an optimal path should be chosen from the set of possible alternatives. A task-oriented model of a DEDP may be applied for that.

2.2 A task-oriented model of a DEDP

It is worth mentioning that the knowledge of the task structure for a dynamic flow like a DEDP is dispersed among the participants of the flow. For example, if the design of a Functional Block B_1 is allocated to the designer D_1, it is up to his knowledge to decide:

– If it is necessary to further decompose B_1 or to consider it as an atomic design thread

– If he will perform the design steps himself or will better try to outsource one or another step (which is actually the sub-task) to another designer

– Which tools he will use at different steps

– Whether he will borrow an existing design solution for B1 or some of its parts or better try to design it from scratch

– If it is necessary to backtrack one or another step according to the verification results

If D_1 outsources a sub-task T_i to D_j than it becomes the sole responsibility of D_j to proceed with this task by taking appropriate decisions and performing corresponding activities (outsourcing, self-performance, tool choice, backtracking, etc.). Of course, D_j may hire D_1 for some of its sub-tasks (being in turn the sub-tasks of D_1).

It is assumed that a *task* $\mathrm{T} = \{t_1, t_2, ..., t_k\}$ is the set of one or more *activities*. Each activity t_k, being atomic for a given actor, may be recognised as a (sub-) task by another actors according to their local knowledge. For example (refer to Fig. 2), 'Perform Design' may be treated as the atomic activity by a person (or an agent) who has never worked in IC design, but will comprise at least 'Formulate Conceptual Idea', 'Perform Functional Decomposition', 'Design FB' from the experienced designer's point of view.

Actors are capable to:

– Perceive incoming tasks from the environment through their communication interface

– Generate new tasks or activities without any external influence in response to some events in the environment

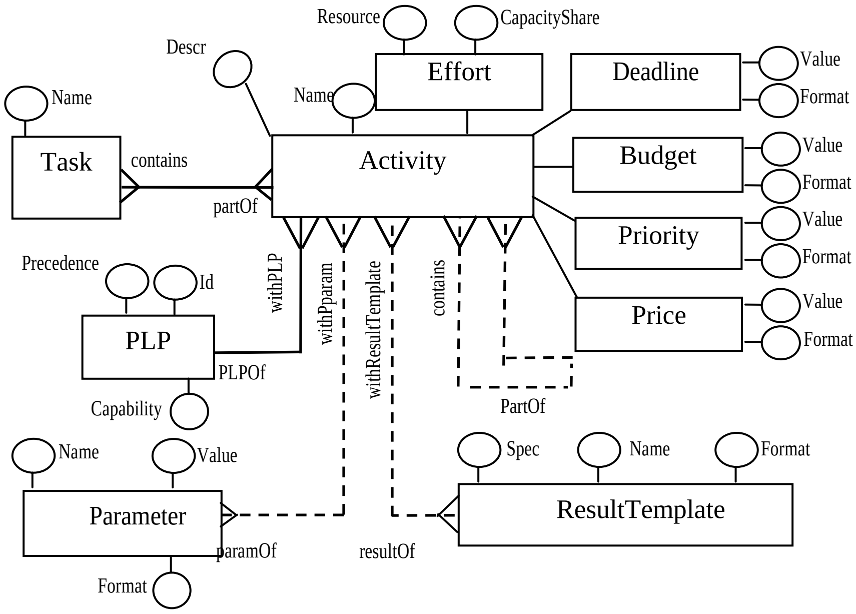


Fig. 3. ER-style diagram of Task Ontology (adopted from [EKT01]).

– Reason weather to perform the atomic activity on their own or to try to outsource it to one of the fellow actors

– Coordinate their joint activities through distributed planning and negotiation

The semantics of a task is formalized in the *Task Ontology* [EKT01] as schematically shown on Fig. 3. Task Ontology provides the conceptualization and the basic vocabulary for different actors to specify the knowledge on the specific tasks and activities in the consensual way. The conceptualization of a task introduces the related notions of the constituent *activity*, activity *atomicity* (for the given actor), the working *capacity* and the required *effort* associated with one or another activity, the *budget* and the *Partial Local Plan* (PLP) for an activity.

The concept of the atomicity is used to reflect the fact that the same activity may be either *atomic* or *non-atomic* for different actors (refer to Fig. 4). For example, a designer working on 'Perform Design' task may consider 'Design FB' an atomic activity to be outsourced to one of the fellow designers.

However, a designer who accepted 'Design FB' may reason that this sub-task is non-atomic and comprises (Fig. 2) at least 'Perform FSD', 'Perform LS', 'Perform PhL', and 'Make TO' activities. To say more generally, actor A according to his knowledge, formalized in the terms of the *Task Ontology*, may consider an activity t_i atomic and having some certain properties shown on Fig. 4. However, actor B, to whom t_i is outsourced by A, has quite a different knowledge of this activity (on the lower level of granularity). For B t_i is a sub-task composed of t_i^1, t_i^2, …, t_i^m which have their own relationships reflected by corresponding PLPs.

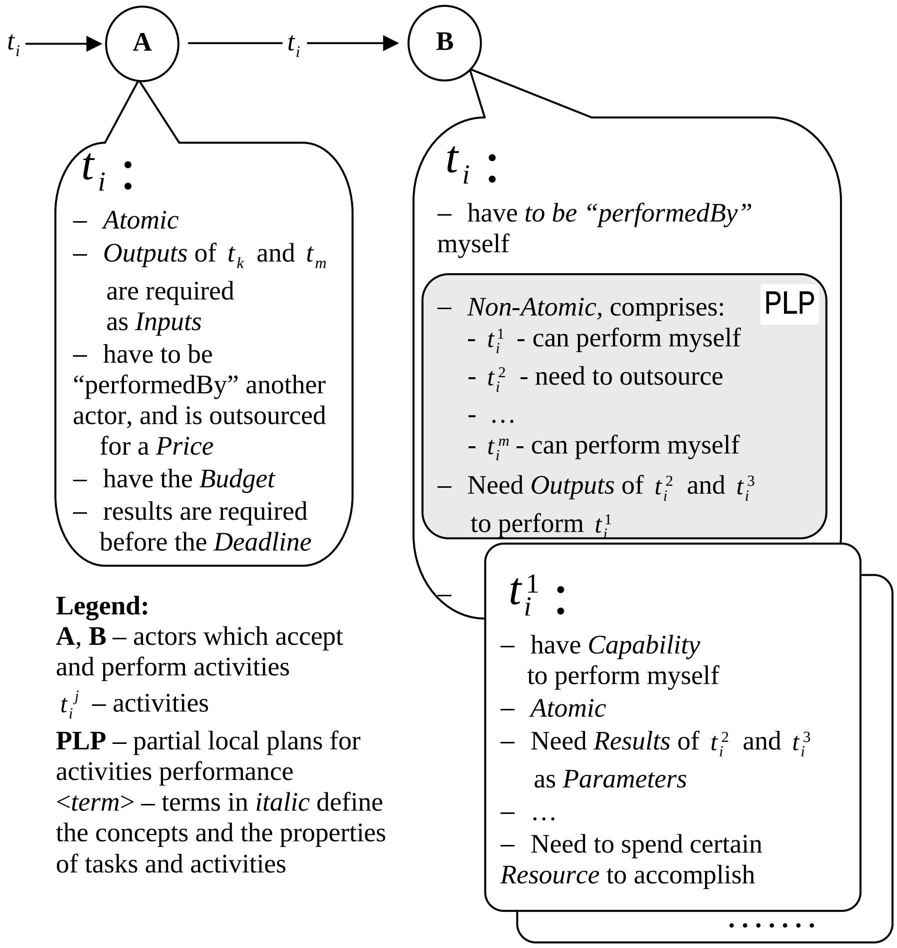


Fig. 4. Activity properties from the point of view of different actors.

2.3 Actors and their roles in DEDP

Let us now pay more attention to the actors who perform tasks and activities. Normally a workflow, like a DEDP, is the kind of an abstraction which is used to model and to control business processes in the real world. A business process is performed within an organization. An organization may be a real company or a virtual entity, open or closed, well structured or flat. More important in our context is the fact that the organization is formed of different actors having their organizational roles. These roles reflect actors' capabilities, authorities, policies, spheres of influence, and spheres of awareness [EP02].

The draft analysis of a DEDP (Fig 1a and 1b) hints at the following actor roles which are likely to be present in a flow:

– A Designer (D). A person who performs designs, interacts with Project Managers, Design Support Provider, Design re-use Provider, and other Designers

– A Project Manager (PM). A person who controls and monitors the performance of a DEDP. A PM may be a super-ordinate for the Ds of his project and a peer to another PMs and the Ds which are not involved in his project

– A Design Support Provider (DSP). An DSP is a program component or a programming environment wrapped by a software agent. An DSP provides services to Ds.

– A Design Re-use Provider (DRP). A DRP is the person who collects, annotates, and provides DSs on Ds' requests

Listed actor roles are filled up both by humans (D, PM, DRP) and by software (DSP). Hence, for the sake of homogeneity it seems to be reasonable to provide wrappers to the actor roles. These wrappers might be:

- Assistant agents for D, PM, DRP
- Software wrapper agents for DSP

The main purpose of these wrappers is to provide the uniform control and simulation software framework for DEDP in the form of a Multi Agent System, referred hereafter as DEDP-MAS.

3 Welfare-Based DEDP Productivity Measure and Rational Collaboration among Actors

One of the important purposes to model DEDPs is to assess the productivity of design processes. Normally, productivity is the integral characteristic of a process and stands for the ratio of the input value to the produced output value. Known productivity measuring methodologies ground themselves on the assessment of design complexity characteristics and pretend to do it fairly by applying heuristic weights to compared parameters (e.g., the normalized transistor count [NUM00] in SES design, FP, KSLOC in software design, etc.). In dynamic environments, where DEDPs are dynamic as well, it is at least very hard and at most impossible to propose a graybox-type heuristics to measure the productivity of a process. Such measures, even if proposed, appear to be not very reliable. It is therefore important to extract more or less stable components from a DEDP and to build the measure which is invariant to the dynamic characteristics of a process. From the other hand, there is the consensus in the Project Management community (PMI recommendations for instance) that a productivity metrics should also be based on the quality metrics of the product. The quality of the design may be only measured by its assessment pointing out to the number of revealed defects per the opportunity of a defect occurrence. Though such an assessment may be performed by reviewing a design, it might also be helpful to have some indirect estimates of the design quality based, say, on how popular the specific design is in re-use. This approach also allows measuring the productivity with respect to the best practices by assessing these best practices as marketable artifacts.

Let's measure the productivity of a DEDP by the assessment of the productivity of the process participants and use a welfare-based model for that. In frame of this model an economically rational actor (modeled by an agent) is the locus of welfare accumulation. An actor receives the units of welfare (UoW) for performing DEDP (sub-)tasks, for providing his Design Solutions (DS), or for lending its Software Tools (ST). Otherwise, an actor may outsource a (sub-)task or require a DS or an ST and pay his UoW for that.

It is obvious that an actor may be considered more productive if he receives more and spends less UoW. In a long run (say, dozens of different DEDPs) the productivity of an actor may be reliably measured by the level of his welfare. The productivity of an organization or a group may therefore be assessed as the sum of the welfare of its members. Important point here is that this productivity measure is invariant to the DEDPs which were actually used to collect the welfare.

An actor may get (or lose) UoW only through collaboration with other actors in DEDP performance. Collaboration occurs when:

- An actor outsources a DEDP (sub-)task to its sub-ordinate by directive or contracts another actor for a (sub-) task
- A DS is re-used in different DEDPs
- An ST is borrowed to perform a DEDP activity

These types of encounters are directive assignments, contracting negotiations, and, possibly, group-buy combinatorial auctions for licensing STs. The mechanisms for these encounters comprise the protocol, the strategy and the social norms. The protocol determines the rules of interaction. The strategy determines the pro-active rational behaviour of a participant. The social norms define the commitments of the participants and the binding conventions.

3.1 Negotiations on Outsourcing a Sub-task or an Activity

Negotiation on outsourcing an activity takes place each time an actor realizes, according to its knowledge of the activity or because of the overload, that the activity should be outsourced to one of the fellow colleges and the actor believes that several appropriate candidates capable to perform this activity are available. Negotiation is performed instead of a directive sub-task assignment in the cases an actor wants to make an optimal choice from the set of the possible contractors. An extension of the **FIPA Iterated Contract Net protocol (CNP)** [FIPA01a] is proposed as the interaction protocol for this kind of negotiation (see Fig. 5). An activity requestor agent is considered an Initiator (**I**) in this encounter. The actors about which **I** believes that they are capable to perform the activity (FCEM, section 3.4) form the party of the invited Participants (**P**).

The first round of the interaction, which is actually the extension of the FIPA protocol, aims to find out if any of the known capable **P**s may agree to perform the activity. Negotiation set for this round contains activity signature only (for example, 'Perform FSD'). **I** may start exploring another opportunities of outsourcing the activity if all **P**s from the sphere of its awareness ([EP02] and section 3.4) refuse in the first round. For example, **I** may require the list of matching freelance service providing agents (SPA) from an Internet public directory (like a web service repository). We shall not go into further details in this direction as far it doesn't add much in the concept.

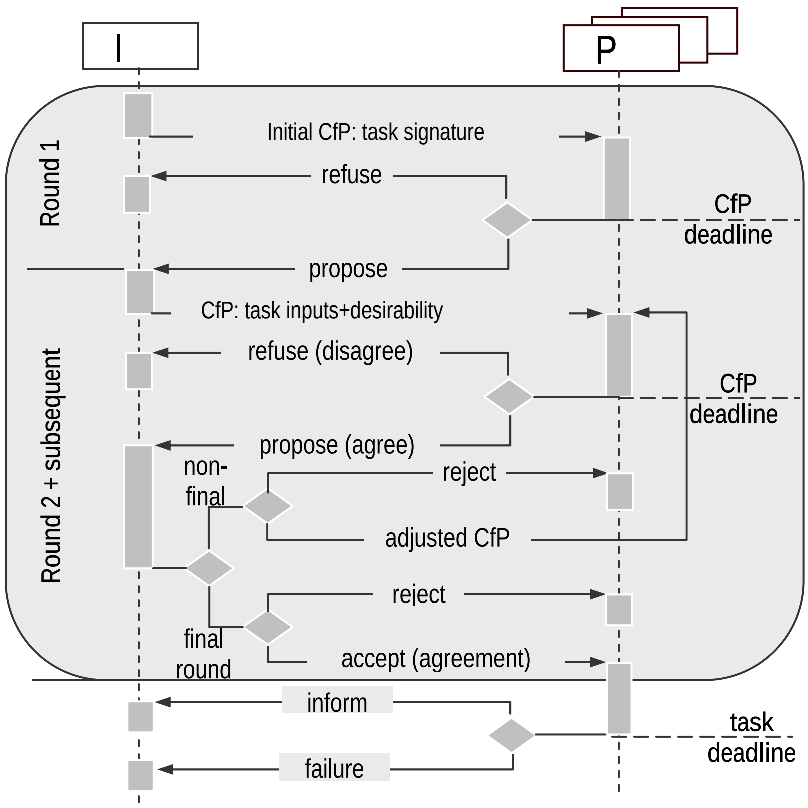


Fig. 5. Extended Iterated FIPA Contract Net protocol for outsourcing a sub-task

Negotiation on the second and the subsequent rounds is about the terms of the possible contract. **I** advertises the activity inputs and the discrete results desirability function as the incentive (in UoW) over time. **I** than chooses the best **P**s proposal weighted by the respective credibility values (Section 3.4) in case several **P**s proposals result in the agreement. Subsequent rounds are used to adjust the activity inputs or the desirability function in the case if no one of the **P**s has agreed on the previous round.

Ps refusals and propositions are shown on Fig. 6. These feedbacks are formulated in a constructive way to allow **I** to adjust its CfP in the subsequent round. A feedback contains two incentive-time points defining the segment on which a possible agreement may be stricken. More details on how the **P**s may compute their feedbacks are given in [EBT02]. Evidently the area of agreement for the current round could be formally defined as the union of all those parts of the feedback segments which are on and below the **I**'s desirability function polyline. All other points of Ps' feedbacks indicate their disagreement with the offer of the current negotiation round.

In case in a round n no agreements were detected by **I** it concedes just enough for not to concede in the next round n+1. The concession in incentive value is computed

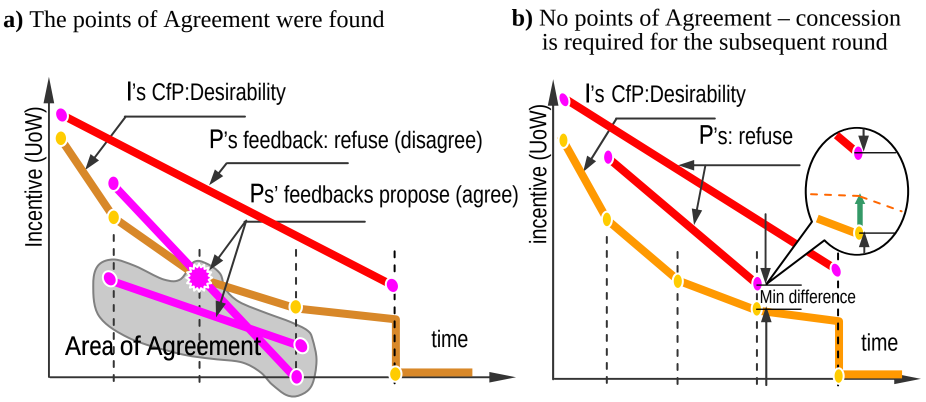


Fig. 6. Negotiation on outsourcing a sub-task: Agreement, Disagreement, and Concession for the next round

as the half of the minimal difference in incentive between the current desirability function and the current feedbacks of **P**s. **I** may continue to concede in the series of rounds if:

1-st, the **P**s concede accordingly in a monotonic way

2-nd, the concession still makes the possible deal individual rational for **I**

I considers the negotiation round as final if it can accept one of the **P**s' agreement and strike the contract deal. The chosen **P** becomes the Contractor and commits itself to the Task Coalition (section 3.5) for the time necessary to perform the outsourced activity.

I may declare the negotiation round as final by repeating the desirability function without concession. Hence, if **P**s do not concede enough to make agreement in the last round, negotiation ends without reaching the agreement.

Negotiation ontology [EKT01] is used as the namespace and the formal semantic frame for the contents of the messages agents communicate with while negotiating on outsourcing a (sub-) task.

In the cases an actor has no choice or prefers to assign the execution of a (sub-)task in a directive manner it uses a directive conversation which is actually a very simplified case of the interaction protocol presented above in this section.

3.2 Negotiations on a DS Re-use

Negotiation on a DS re-use occurs when a human designer realizes due to his experience that a DS for his current design might be available at his colleges or in the corporate DS repository. The designer than instructs his personal assistant agent to perform DS re-use negotiation for him by providing the desired ranges of DS characteristics like, for example: input voltage, V (2.5 – 5.4); output frequency, MHz (1.80 – 1.85); linear dimensions: length, mm (20-30), width, mm (10-15); etc. *Design Solution Ontology* is used as the formal mean for representing the shared conceptualization and the terminology for that. Like for sub-task allocation the extended **FIPA Iterated CNP** is used for this kind of negotiation. (see Fig. 7). A DS

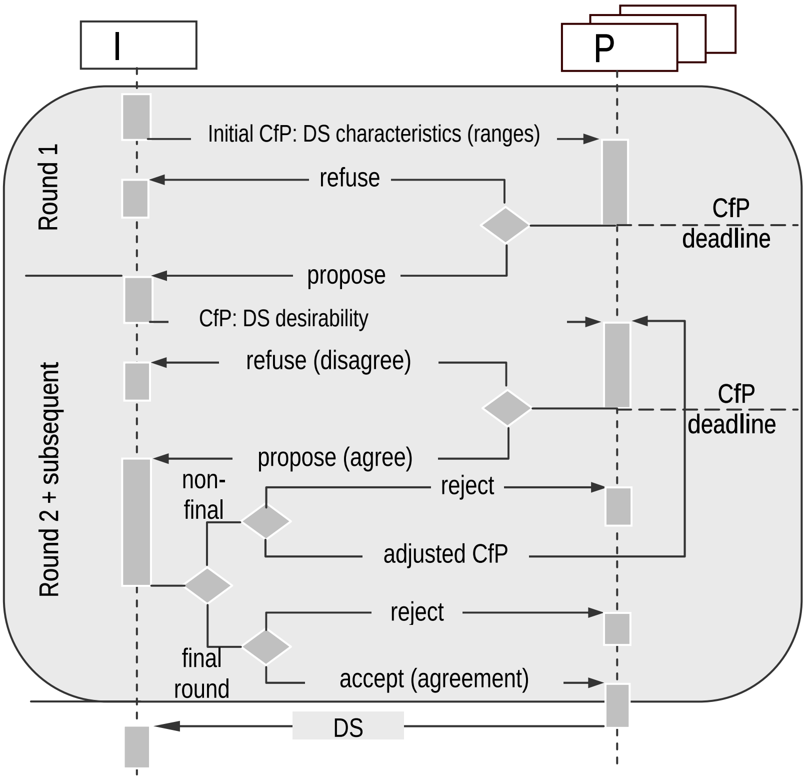


Fig. 7. Extended Iterated FIPA Contract Net protocol for DS re-use negotiation.

requestor agent is the Initiator (**I**) in this encounter. The agents about which **I** believes that they are capable to provide a matching solution are the invited Participants (**P**).

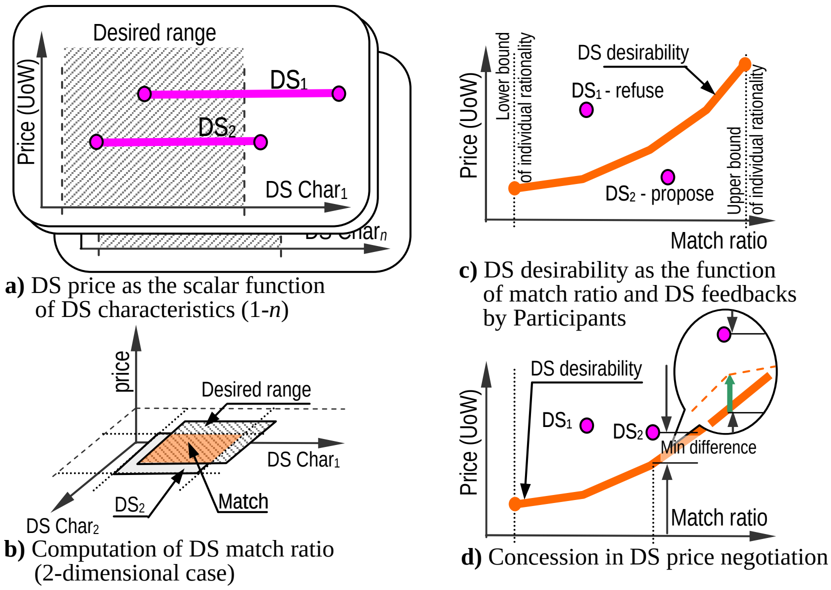
The first round of the interaction, aims to find out if any of the believed capable **P**s may actually have a desired DS in stock. Negotiation set for this round contains DS type only (for example, 'Input Signal Amplifier'). In case no proposals were received back from **P**s at this round **I** will proceed with the design from scratch.

Negotiation on the second and the subsequent rounds is about the terms of the possible contract. **I** advertises the desired space of the ranges of DS characteristics (Fig 8a) and the discrete DS desirability function as the price over the DS match ratio (Fig. 8c). **P**s reply with their DS characteristics ranges and the price of the DS. **I** than computes the match ratio *M* for each received feedback (Fig. 8b):

$$s_i^{\min} = \begin{cases} d_i^{\min} & \text{if } s_i^{\min} < d_i^{\min} \\ s_i^{\min} & \end{cases},$$

$$s_i^{\max} = \begin{cases} d_i^{\max} & \text{if } s_i^{\max} > d_i^{\max} \\ s_i^{\max} & \end{cases}, \qquad (1)$$

$$M = \prod_{i=1,n} \left(1 + \frac{(d_i^{\min} - s_i^{\min}) + (s_i^{\max} - d_i^{\max})}{d_i^{\max} - d_i^{\min}} \right),$$



a) DS price as the scalar function of DS characteristics (1-*n*)

b) Computation of DS match ratio (2-dimensional case)

c) DS desirability as the function of match ratio and DS feedbacks by Participants

d) Concession in DS price negotiation

Fig. 8. Negotiation on a DS re-use: ranges of DS characteristics, DS match ratio, desirability and concession.

where:

s_i^{min} , s_i^{max} - the lower and upper bounds of the characteristic range for the characteristic *i* of the DS by **P**,

d_i^{min} , d_i^{max} - the upper bounds of the desired characteristic range for the characteristic *i* of the DS by **I**.

Geometrical interpretation of (1) is given on Fig 8b. *M* shows which part of the volume of the desired characteristics space is covered by the space of the ranges of characteristics of the proposed DS.

After the match ratios are computed for each **P**s' feedback **I** proceeds with the DS price analysis as shown on Fig 8c by comparing DS propositions, which are now the points on the Match ratio – Price grid, with its DS desirability function. **I** chooses the best **P**s proposal weighted by the respective credibility values (Section 3.4) in case several **P**s proposals result in the agreement. Subsequent rounds are used to attempt to find the agreement on the DS desirability – DS price in the case if no one of the **P**s has agreed on the previous round.

Ps refusals and propositions are shown on Fig. 8c. A feedback indicates the agreement if the corresponding match ratio – price point lies below or on the desirability function polyline. All other points of **P**s' feedbacks indicate their disagreement within the current negotiation round.

In case in a round *n* no agreements were detected by **I** it tries to concede just enough for not to concede at the subsequent round *n*+1. The concession in price value is computed as the half of the minimal difference in DS price between the current desirability function and the current feedbacks of **P**s (Fig. 8d). **I** may continue to concede in the series of rounds if:

1-st, the **P**s concede accordingly in a monotonic way

2-nd, the concession still makes the possible deal individual rational for **I** with respect to the upper bound of the price value.

I considers the negotiation round as final if it can accept one of the **P**s' proposition and strike the contract deal. The chosen **P** becomes the Contractor and commits to provide the DS to **I**.

I may declare the negotiation round as final by repeating the desirability function without concession. Hence, if **P**s do not concede enough to make agreement in the last round, negotiation ends without reaching the agreement.

Negotiation ontology [EKT01] is used as the namespace and the formal semantic frame for the contents of the messages within this kind of negotiation.

3.3 Choosing a Software Tool

There are principally two possible scenarios for selecting and providing an ST to a designer. The simpler one occurs when the relevant tools are already licensed by the company, the department, or the group under simulation. It is therefore under a human designer's responsibility to choose the tool from the available variety according to his requirements (e.g., his familiarity with the tool, his favorable attitude to it because of the previous successful experience, whatever). In this simple case the tool provision parameters are not negotiated and are not subjected to bargain – the usage price (in UoW) is fixed and may be determined according to the internal policy of the organizational structure. The protocol for this scenario is straightforward and comprises the request from a designer, the reply of the STP providing access to the chosen tool, the request from STP charging UoW from D for the usage, and actual UoW transfer by D. The strategy for a D here is very simple and is to choose the cheapest ST from the subset of the equally most attractive among the available ones.

The more complicated scenario occurs when the required tool is not available and should be therefore licensed. In this settings the STP should ground his necessity to purchase the tool by the unsatisfied demand from Ds. From the other hand the Ds may form the so-called buyers' coalitions to carry more conviction and to expect the wholesale discounts. A combinatorial auction framework may be used for such kind of scenarios (refer to Section 3.5.2).

3.4 Adjusting Behaviors and Strategies

An actor is situated in the environment. The environment comprises DEDP tasks and other actors who collaborate with the given one to perform these tasks. An actor may contribute to the execution of some tasks in different ways:

- By performing a task or an activity
- By lending an ST it wraps to another actor, who performs a (sub-)task
- By providing a DS to another actor, who performs a (sub-)task

The tasks/activities the actor contributes to form his Sphere of Influence within the environment (refer to Fig. 9).

Actor's capabilities to influence the environment change over time. At a certain

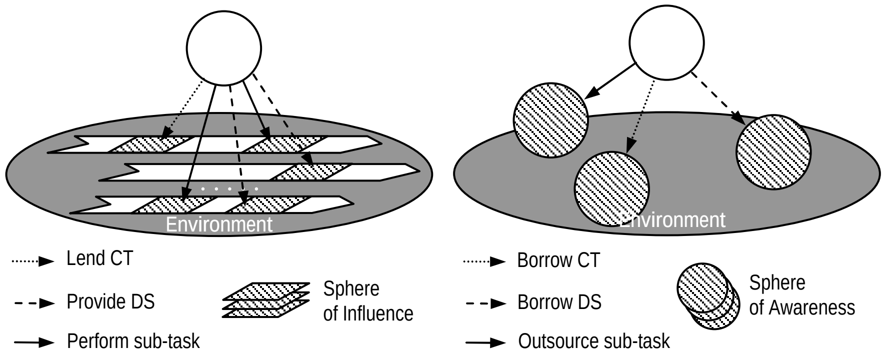


Fig. 9. Actor's Sphere of Influence and Sphere of Awareness

time:

– An actor may have or have no free capacity to perform an activity which he was able to do before because of the bulk of other concurrent activities under execution

– An actor may produce new DSs and therefore gain more capabilities to provide them for re-use

– An actor may acquire new STs and thus provide new ST services to other actors

The more capabilities an actor has – the bigger his Sphere of Influence becomes bringing him more UoW and making him more productive. Hence, an actor needs to produce new solutions, gain more experience and learn to perform his sub-tasks more effectively.

To be effective an actor should be aware of his fellows who may provide him services at optimal conditions and, therefore, should constantly update his beliefs about changing capabilities of his fellows. Fellow actors, whose capabilities are believed to be known at a certain time, form the Sphere of Awareness (Fig. 9.) in the environment of an actor. It is also important for an actor to maintain his beliefs on if he may trust to one or another fellow within the Sphere of his Awareness.

The mechanisms of adjusting actor's beliefs on his fellows' capabilities and credibility are based on the reinforcement learning. The idea is to use the results of each encounter to reinforce the capability and credibility factors. These beliefs are than used as the weight factors in accessing the proposals of the **P**s in contracting negotiations (Sections 3.1-3.3). Hence, properly maintained beliefs on fellows' capabilities and credibility may help in fine-tuning the negotiation strategies and may influence decision making on if to outsource a (sub-) task or to perform it personally, if to ask for a DS or to design from scratch.

3.4.1 Adjusting Beliefs on Capabilities

Possible mechanism to define the perspective contractors is capability matchmaking (e.g., based on LARKS [SWK02]), or service discovery technique based on UDDI, or another service matching facilities (e.g., semantic matching based on DAML-S profiles [PKP02]). However, in case there is some capability beliefs record maintained autonomously by an actor in the course of cooperative task execution, the use of this knowledge may substantially facilitate to lowering computation costs by eliminating unnecessary directory/matching service usage. Evidently, if A believes that B, C and D are capable of performing desired activity because they did it before, he will rather proceed to contracting negotiation with B, C and D directly instead of trying to find some other fellows with matching capabilities[1].

A model and a mechanism of agents' capability assessment is based on actor's beliefs representation in the form of Fellows' Capability Expectations Matrix (FCEM). Actors accumulate and adjust their local beliefs on the capabilities of their collaborators from the experience of cooperative performance. New portions of this knowledge appear each time a (sub-)task is being outsourced to a fellow. Subjective beliefs of the actor on the probabilities of its fellows' capabilities to perform the given (sub-)task are thus updated. FCEM for capability beliefs representation is maintained in the following form:

$$\mathbf{C} = \begin{array}{c} \\ \text{SPA}_1 \\ \\ \ldots \\ \\ \text{SPA}_n \end{array} \begin{array}{c} \begin{array}{ccccc} t_1 & \ldots & t_j & \ldots & t_m \end{array} \\ \begin{bmatrix} c_1^1 & & c_1^j & & c_1^k \\ & & \ldots & & \\ & \ldots & c_i^j = (q_i^j, p_i^j) & \ldots & \\ & & \ldots & & \\ c_n^1 & & c_n^j & & c_n^m \end{bmatrix} \end{array}, \tag{2}$$

where dimensions m and n change reflecting the appearance of new incoming (sub-)tasks and newly discovered or perishing fellows – Service Providing Agents (SPAs).

Capability estimations c_i^j change each time an agent negotiates with its fellows on outsourcing (sub-)task t_j. Element q_i^j in tuple c_i^j stands for the quantity of recorded negotiations with fellow agent SPA_i concerning (sub-)task t_j. Element p_i^j stands for the capability expectation. The rule for c_i^j updates is as follows:

$$\begin{aligned} &1. \quad p_i^j \leftarrow p_i^j + \frac{r}{q_i^j}, \\ &2. \quad q_i^j \leftarrow q_i^j + 1 \end{aligned} \tag{3}$$

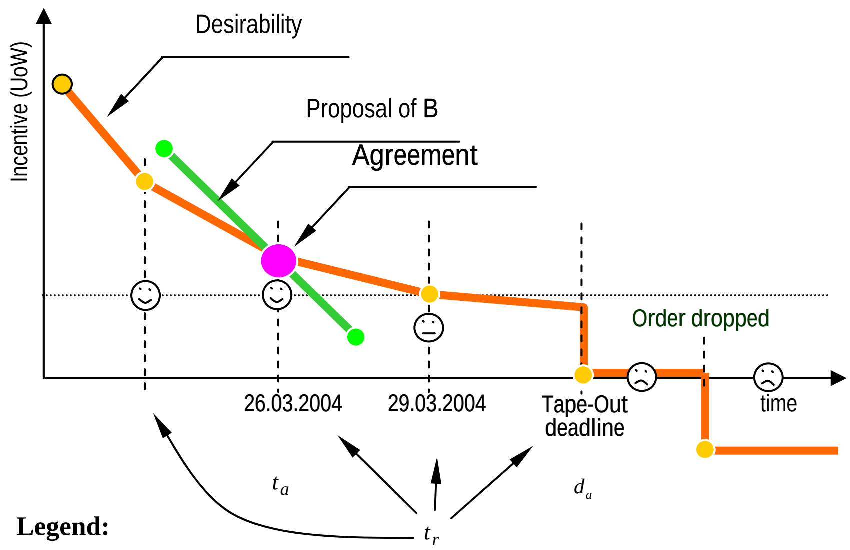
where: r is equal to: 0 – if the fellow rejected t_j, 0.5 – if the fellow replied that it can accept t_j, and 1 – if t_j was finally outsourced to the fellow.

The mechanisms for maintaining the beliefs on fellows' capabilities to provide a DS or to lend a ST are the same to that presented above.

3.4.2 Adjusting Beliefs on Fellows' Credibility

As we have agreed that collaboration among actors is the only way to bring them about the higher level of welfare, they shouldn't behave as if they are in a strictly competitive encounter. That means: the only way an actor may sell his fellow short is to violate the time of the contract results delivery (which is allowed according to the

[1] Applying to a capability registry may still appear to be necessary in case B, C and D fail to provide constructive proposals.



Legend:

- agreed accomplishment time and incentive

- factual accomplishment times and result desirability values

Credibility factors associated with the Service Provider:

- before the agreed time – the fellow remains as credible as it was

- after the agreed time – the fellow looses certain percent of its credibility

- after the deadline – the fellow looses all of its credibility

Fig. 10. Task accomplishment times and corresponding credibility changes.

relative cooperation commitment [EP02]). Hence, it is sufficient to access the credibility of a fellow in (sub-) task outsourcing negotiations only.

A self-interested fellow actor, due to the appearance of the new highly attractive task offers in the competitive environment or due to the peculiarity of its behavior, may lower previously declared capacity it is spending for the bulk of the activities under execution [EBT02]. This will lead to the increase of the performance duration, may therefore seriously decrease the requestor's desirability of these results and, thus, lower the actor's credibility value in the eye of the task requestor.

Let for example A contracts B for 'Make TO' task. The result of the contract is: *the tape-out of the design is at the factory*. The agreed delivery date is 26.03.2004, though the deadline advertised by the task requestor before is 30.03.2004. A will evidently consider B, who delivered the tape-out before or right on the agreed date as credible. However, if B delivers the tape-out on 29.03.2004, A may feel itself aggrieved, though his commitments to the factory are fulfilled. A's beliefs on the credibility of B will therefore be lowered. Further on, if the tape-out appears at the factory after the announced deadline, A may rightfully consider that the contract terms were seriously violated. B's credibility should be therefore drastically lowered. Finally, imagine that A still waits for the tape out when the factory has already dropped his advance order for chip production. In the latter case A may even want to require a penalty (UoW) in addition to lowering B's credibility down to zero. To summarize, it is natural to measure the changes of A's beliefs on B's credibility by the losses of the desirability of the contract results (refer to Fig. 10).

The mechanism of accounting fellows' credibility values is similar to that of adjusting the beliefs on changing fellows' capabilities (2-3). Credibility assessment values change over time as the requestor agent adjusts its subjective beliefs by comparing the desirability values (Fig. 10) derived from:

1-st – task duration the contractor committed to within the task outsourcing negotiation and

2-nd – actual results delivery time

Corresponding credibility matrix elements are than recomputed due to the following:

$$Cr_{i,j} := Cr_{i,j} \times \begin{cases} 1, t_r \le t_a \\ p_a(t_a / t_r), t_a < t_r \le d_a \\ 0, t_r > d_a \end{cases}, \qquad (4)$$

where: t_a is the time the parties have agreed to accomplish task a, t_r is the actual time of a results delivery, d_a is the deadline and p_a is the weight coefficient characterizing the current priority of a for the task requestor agent.

Credibility threshold values associated with respective activities and stored in agents' PLPs are used by task requesting agents to assess possible risks and alter their strategies.

3.5 Coalitions of Actors

The notion of a coalition in collaborative activities stands for a temporary group of actors pursuing their goals coherently in a rational way. A coalition is viable only in the case when it brings extra benefits to its members. For example, in our context a coalition of actors in DEDP environment is viable if each coalition member gets more UoW than if it acts on his own.

Coalitions are shaped out by coalition formation mechanisms which define the binding agreements in the form, e.g., of commitments and conventions and the pay-off distributions. A pay-off distribution here stands for the mechanism of distributing extra welfare received by the coalition as a whole among the individual coalition members.

The reasons to form coalitions of actors in DEDP settings are as follows:

– **A task may be performed only by a group of actors, but not by a sole individual:**

Though a Task Coalition may bring no extra UoW to its participants, but they will do no worse than if they did alone – so the coalition is individual rational. From the other hand, the Task Coalition helps each of the members to get UoW by providing contracts. This possibility may be considered as an extra benefit.

– **A Design Re-use provider** (an DRP) **may propose discounts for certain volumes of a DS**:

A DS Users Coalition may be the mean to group the demands of individual DS consumers. By that they may get their retail volumes of a DS for a discount price and save some UoW – that is the extra benefit.

– **A DSP may propose discounts for lending certain volumes of a ST installations**:

The extra benefit is similar to a DS Users Coalition.

A Task Coalition is formed through contracting negotiations, as was mentioned in Section 3.1, and is mainly a social structure which regulates the proportion of the self-interest and the benevolence of its participants through binding commitments and a convention. UoW in a Task Coalition are distributed according to the contracts on (sub-) task execution. However, the latter two ones are classical buyers' coalitions (ref., e.g., [JS01]) and therefore require the pay-off distribution mechanism.

3.5.1 Social Norms in Coalitions

Social norms are used to frame out actors' behaviors in Task Coalitions. These rules are hardwired into the design of the corresponding agents to provide for more reactivity in resource-bounded conditions. By joining the Task Coalition an actor pledges to follow coalition rules regulating the proportion of its benevolence and self-interest. These rules may be classified, following Jennings Commitment-Convention hypothesis [JEN96] as actor's Individual and Joint Commitments and Coalition Convention:

– **Relative co-operation commitment**. Coalition members are relatively committed to co-operatively achieve the overall goal: to accomplish the task with maximally achievable effectiveness (maximal quality, balanced load, minimal time, ...). The ratio of this commitment depends upon the discrepancy between the actor's autonomous intentions and the overall goal of the task coalition. By this commitment an actor is allowed to a certain extent violate the terms of his contract – to accomplish the task later than it was stated by the contract. This may of course result in the loss of credibility. The reason for an actor to risk his credibility is the acceptance of other very attractive task offers resulting in the redistribution of his capacity.

– **Results delivery commitment.** Since a task is accepted by the actor for the performance the actor pledges to unconditionally accomplish this activity and to deliver the results immediately after the work is done. This commitment is also valid for the DS or ST delivery.

– **Negotiation convention**. The actors engaged in negotiations on outsourcing a task or DS re-use pledge to truthfully advertise desirability functions related to the negotiated task or DS. In response, the perspective contractors (the **P**s) commit to truthfully report about their readiness to perform the task or the availability of the DS.

3.5.2 Coalition Formation and Payoff Distribution in DS and CT Buyers' Coalitions

A stable and efficient coalition formation and pay-off distribution mechanism for a buyer coalition in combinatorial auction settings may be borrowed from [JS01].

4 Coordination and Communication in DEDP-MAS

As it was mentioned before collaboration focuses on getting extra UoW from team activities. Coordination in a broad sense is a complementary addendum to collaboration as it focuses on saving more UoW. Actors may save UoW in the process of their cooperative work by making their activities more coherent to each other, by effectively "managing the interdependencies between activities" (cf.[MC91]).

Activities within a task may be *independent*, *strongly dependent* on other activities and facilitated (or *weakly dependent*) by other activities [NL99]. Task t_1 is said to be independent of task t_2 if the performance of t_2 does not depend from the process of t_1 performance or the results of t_1 and vice versa. An example of independent task within an IC DEDP is several "Design FB" sub-tasks for different functional blocks. The task t_1 is said to be strongly dependent of task t_2 if the results of t_2 are essential to start the execution of t_1. For example, "Perform LS" task strongly depends on "Perform FD" because the result of "Perform FD" (functional specification) is one of the inputs for "Perform LS". Finally, task t_1 is said to be facilitated by task t_2 if the execution of t_2 or the results of t_2 may help to execute t_1 in less time, with less resources consumed, or obtaining better quality with the same resource consumption. For example, getting proper DS from a fellow may facilitate to "Perform FD" task if the price paid for the DS is less than the price of doing the design from scratch on alone.

Evidently, it is irrational to coordinate independent activities – there is no way to get savings from that. Coordination of the weakly dependent tasks is arranged through negotiations on outsourcing a task (Section 3.1), on DS re-use (Section 3.2), or on borrowing an ST (Section 3.3). Coalition formation may also be considered as the way to coordinate the activities of the coalition members. In a more narrow sense, the coordination of strongly dependent tasks is essentially the routine of putting these tasks in the proper sequence and providing the conveyer for feeding the results of predecessors as the parameters to the successors. The sequence of these tasks is defined by the local knowledge of the participating actors (their PLPs of these tasks). In a PLP (refer to Section 2.2) a predecessor task is indicated as the pre-condition for the successors. The conveyer for the task results is maintained by the utility Coordination Agent (CoA) in the form of a tuple space (refer to, e.g., [RCD01] for a survey). More details may be borrowed from [EP02].

Collaboration and coordination is realized through the conversations among the participating actors. Conversations are eventually the kind of logical transactions composed of several communicative acts. The communicative acts are compliant to FIPA Communicative Act Library Specification [FIPA01c]. FIPA ACL is used as the agent communication language. Communication messages are structured according to [FIPA02].

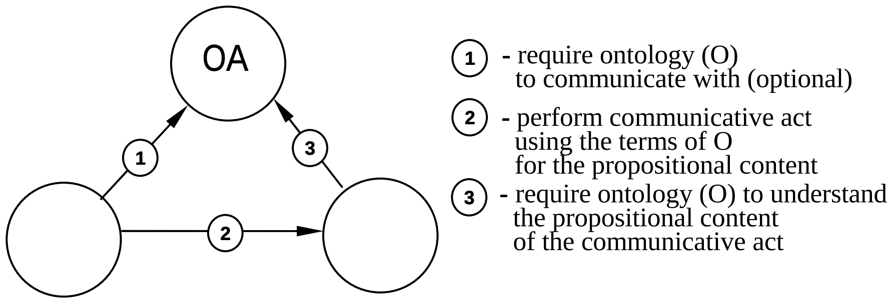


Fig. 11. FIPA scheme of ontology-based communication.

5 Shared Knowledge and Terminology Representation

DEDP-MAS agents need shared formal terminology specifications in the form of ontologies to understand each other in their encounters. These ontologies provide consensual conceptualizations of the phenomena used in DEDP interactions like a task, an activity, an incentive, a desirability function, a feedback, capacity, match ratio, a DS characteristic, etc. DEDP-MAS ontologies are not hard-wired into the agents, but are represented explicitly and coded in OWL [OWL03]. Pending FIPA Ontology Service Specification [FIPA01b] a dedicated Ontology Agent (OA) is the utility agent of DEDP-MAS. The function of the OA is to supply DEDP-MAS agents with ontologies on their requests. DEDP-MAS agents use these ontologies in their communication as it was mentioned in Section 4 and is shown on Fig. 11.

While DEDP-MAS ontologies specify the core knowledge of the system providing the semantic frame for the consensual conceptualizations, the local knowledge of each member-agent may differ from the core set in the sense that the local knowledge is the instantiation and the more detailed knowledge specification within the core knowledge frame. For example, a *Task Ontology* specifies the generic concepts and properties of a generic task, while the specification of a 'Perform FSD' task at one of the D's knowledge base formalizes the local knowledge of the specific D on this specific task, but in the terms of the generic *Task Ontology*. When D_1 intends to outsource a 'Perform FSD' task to D_2 it points to the Task Ontology as to the terminological frame, though the knowledge of D_2 about 'Perform FSD' may certainly differ from that of D_1 (refer to Fig. 4). The local knowledge of DEDP-MAS agents is stored in their local knowledge bases in the form of MASDK OL ontologies which use DEDP-MAS ontologies as namespaces.

The hierarchy of DEDP-MAS ontologies and local ontologies is presented on Fig. 12. DEDP-MAS knowledge base comprise the ontologies of the Task Family, the Ontologies of the Negotiation Family and the Family of Design Artifacts – the ontologies describing the generic concepts and properties of Design Solutions and Software Tools. The top-level ontology of the Task Family is the generic Task Ontology [EKT02]. It provides the namespace for DEDP Ontology and its subclass ontologies of Analog IC DEDP, Digital IC DEDP and Mixed IC DEDP. These ontologies are used as the namespaces for the ontological specifications of design

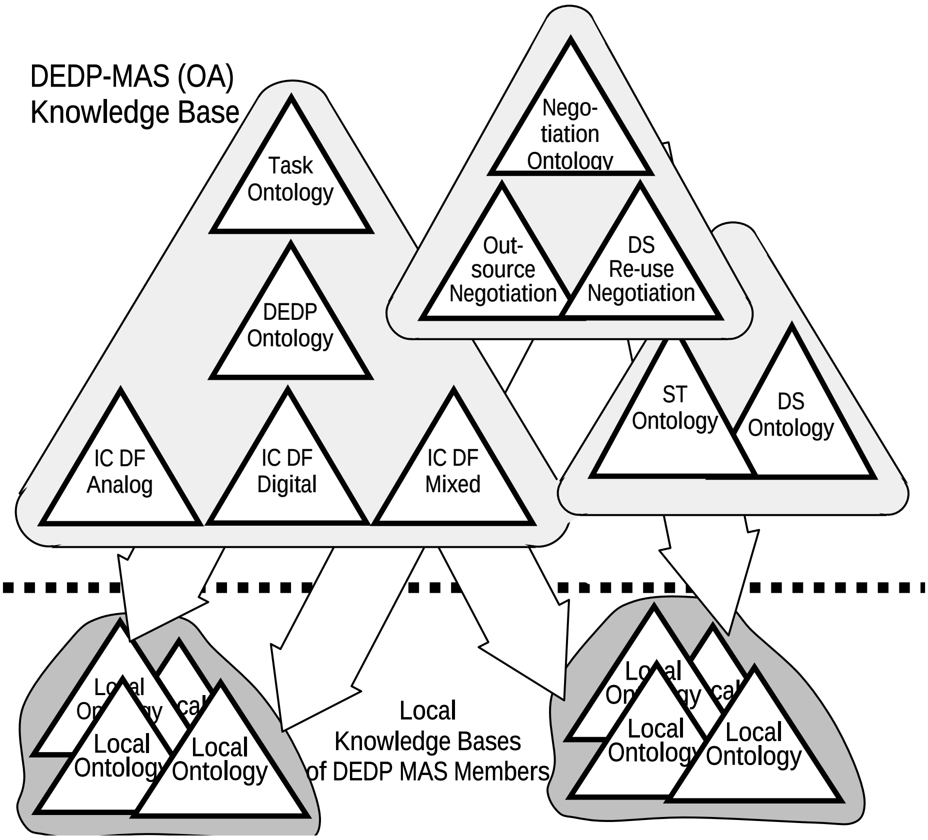


Fig. 12. DEDP-MAS Ontologies Hierarchy

tasks at the local level.

The top-level ontology of the Negotiation Family is the generic *Negotiation Ontology* [EKT02]. The ontologies conceptualizing a negotiation on outsourcing a task and a negotiation on a DS re-use inherit the top-level *Negotiation Ontology* as their semantic frame. These ontologies are further on used to specify the propositional content (advertisements and feedbacks) in peer-to-peer negotiations on the local level. Top-level conceptualizations of the generic properties of DSs and STs are provided by the *DS Ontology* and *ST Ontology* respectively. These ontologies provide terminology for the semantic annotation of the available Design Solutions and Software Tools by different DEDP-MAS members.

6 Concluding Remarks

This report specified the initial draft of the generic modeling framework for the simulation of the Dynamic Engineering Design Flows (DEDP) in the domain of

Integrated Circuit (IC) Design. It is well known that design itself is a very highly creative human activity which is often referred to as an art rather as a routine engineering development. However, it is important to organize design activities in very well structured and optimized processes in order to be at the proper level of organizational maturity. The very high level idea of the framework is to leave design activities to human designers and to focus on the proper means to achieve the optimal performance in DEDPs. That is why the agents mentioned throughout the report are of two major types: personal assistants providing people with the automated support of their routine functions (like Ds, DSP) and utility agents providing specific functionalities for DEDP-MAS as a whole (like CoA, OA, and DRP).

The framework presentation has been structured as follows. First, the factors which make a DEDP a highly dynamic process were presented. These factors motivated the introduction of the Task-Oriented DEDP Model and the idea of DEDP productivity metrics based on the interpretation of an actor within a DEDP as the locus of welfare accumulation. It was also stated that the more effective and efficient mechanisms for UoW accumulation may be obtained through rational collaboration among such actors – namely several types of negotiation and coalition formation. The presented mechanisms for adjusting actors' behaviors and strategies through monitoring the capabilities and credibility factors of their fellows may be used to further enhance the effectiveness and the reliability of the modeling framework. The framework also drafted out the requirements for coordination and communication among the agents, which model actors in DEDPs. It was stated that one more critical factor for proper DEDP modeling in the defined distributed settings is the proper shared knowledge and terminology representation. The framework assumes that this knowledge is structured and represented in the form of ontologies. While the ontologies maintained by the system OA specify the core knowledge of the system providing the semantic frame for the consensual conceptualizations, the local knowledge of each member-agent may differ from the core set in the sense that the local knowledge is the instantiation and the more detailed knowledge specification within the core knowledge frame.

The implementation of the DEDP-MAS based on the presented modeling framework seems to be feasible.

The further activities towards this implementation are seen as follows:

- Evaluation of the framework by manually applying it to the two very simple IC design cases – one for a digital IC and the other for an analog IC.
- Implementation of the demonstration software prototype of DEDP-MAS for the one of the analyzed cases
- Extension of the demonstration prototype to the research prototype by adding the functionalities for the another case if necessary
- Planning the series of evaluation experiments with the research prototype for several real design cases which will form the experimental testbed
- Preparation of the experimental testbed by the design and the deployment of the ontologies and the agents' knowledge bases
- Actual performance of the evaluation experiments and the analyses of the experimental results
- Elaboration of the recommendations for the further DEDP-MAS development and the in-house use

References

[EBT02] **Ermolayev, V. A., Borue, S. U., Tolok, V. A., Keberle, N. G., Plaksin, S. L.** (2002) Arranging Cooperative Business Activities in Dynamic Coalitions of Rational Actors. Tech. Report: Zaporozhye State University, Zhukovskogo, 66, 69063, Zaporozhye, Ukraine, May, 2002. Available from: http://eva.zsu.zaporizhzhe.ua/eva_personal/PS/ITM-SI-WF-EB-02.pdf

[EKT01] **Ermolayev, V. Keberle, N., Tolok, V**. (2002) OIL Ontologies for Collaborative Task Performance in Coalitions of Self-Interested Actors. In: H. Arisawa, Y. Kambayashi, V. Kumar, H.C. Mayr, I. Hunt (Eds.):Conceptual Modeling for New Information Systems Technologies ER 2001 Workshops, HUMACS, DASWIS, ECOMO, and DAMA, Yokohama Japan, November 27-30, 2001. Revised Papers - LNCS vol. 2465, p. 390-402

[EP02] **Ermolayev, V. A., Plaksin, S. L.** (2002) Cooperation Layers in Agent-Enabled Business Process Management. *Problems of Programming* #1-2, 2002, p. 354-368

[FIPA01a] FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. FIPA Contract Net Interaction Protocol Specification. Ref. No XC00029E. 2001. http://www.fipa.org/specs/fipa00029/.

[FIPA01b] FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. FIPA Ontology Service Specification.Version D, Ref. No XC00086D. 2001. http://www.fipa.org/specs/fipa00086/.

[FIPA01c] FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. FIPA Communicative Act Library Specification. Version F, Ref. No PC00037F. 2001. http://www.fipa.org/specs/fipa00037/.

[FIPA02] FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. FIPA ACL Message Structure Specification. Version G, Ref. No SC00061G. 2002. http://www.fipa.org/specs/fipa00061/.

[JEN96] **Jennings, N. R.** (1996): Coordination Techniques for Distributed Artificial Intelligence. In: G. M. P. O'Hare and N. R. Jennings (eds.) Foundations of Distributed Artificial Intelligence, Wiley, 1996, pp. 187-210.

[JS01] **Yamamoto, J., Sycara, K.** (2001) A stable and efficient buyer coalition formation scheme for e-marketplaces. In: Proc of the 5-th Int Conference on Autonomous agents, Montreal, Quebec, Canada

[MC91] **Malone, T. and Crowston, K.** (1991) Toward an interdisciplinary theory of coordination. Center for Coordination Science. Technical Report 120, MIT Sloan School of Management.

[NL99] **Nagendra Prasad, M. V., and Lesser, V. R.** (1999) Learning situation-specific coordination in cooperative multi-agent systems. *Autonomous Agents and Multi-Agent Systems*. 2(2), 1999, p. 173-207

[NUM00] Measuring IC and ASIC Design Productivity. White Paper. Numetrics Management Systems, 5201 Great America Parkway, Suite 320 Santa Clara, CA 95054, 2000

[OWL03] OWL Web Ontology Language. Semantics and Abstract Syntax. W3C Proposed Recommendation, 15 December 2003. http://www.w3.org/TR/2003/PR-owl-semantics-20031215/

[PKP02] **Paolucci, M., Kawamura, T., Payne, T., Sycara, K.** (2002): Semantic Matching of Web Services Capabilities In: Proc. of Int. Semantic Web Conference (ISWC'2002), June 9-12, 2002, Sardinia, Italy

[RCD01] **Rossi, D., Cabri, G., Denti, E.** (2001) Tuple-based Technologies for Coordination. In: A.Omicini, F.Zambonelli, M.Klusch, R.Tolksdorf (eds.): Coordination for Internet Agents - Models, Technologies, and Applications, Springer-Verlag, 2001. XXVII, 523 pp.,

ISBN 3-540-41613-7, 83–109.
[SWK02] **Sycara, K., Widoff, S., Klusch, M., Lu, J.** (2002): LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. Autonomous Agents and Multi-Agent Systems 5 (2002) 173–203