Integrating a Distributed and Heterogeneous Organisation Using Constraint Programming

1st September 2000

Carlos Eisenberg

CERN-EST-LEA
1211 Geneva 23
Switzerland
carlos.eisenberg@cern.ch

Abstract

The Internet globally connects a great deal of diverse and heterogeneous information resources, which an agent can dynamically combine in order to synthesise new information. An agent society is used to facilitate a common project within globally distributed, heterogeneous organisations. Besides coordinating a finite number of variables, the agents are used to iron out cultural differences within the organisations to ensure the overall success of the shared project. Constraint programming is used to compute schedules, to translate cultural dependent meaning of terms and to serve as a carrier for unambiguous information exchange for the negotiation based co-ordination process.

1. Introduction

One of the reasons why the Internet has become so successfully popular is that it globally connects people and a vast amount of diverse information resources across borders, be it geographical, cultural or organisational. In principle, within a very short time, any piece of electronic information can travel over great distances and can be present in any Internet connected computer. The possibility to boundlessly communicate through the Internet has motivated the development of a great deal of diverse applications. These applications range from simple communication tools like electronic mail, news groups, chat-rooms over different commercial applications such as online auctioning, e-banking and e-commerce to diverse business applications such as simple information web pages, information, product and customer support and even complex company integration systems.

With the ever-growing wealth and diversity of available information resources and the ease of bringing these together, new ways of using the Internet are opening up. Different information resources can be combined allowing the synthesis of new information. For example with the information of my online travel agency, the online agendas of my working colleagues and my online hotel booking system, I can use the Internet to solve a complex problem such as synthesising a travel plan for my business trip [1] to meet my colleagues overseas. The demand for such applications, which could then even automate this process, is very high.

Intelligent Agents [2] is seen as a new computing paradigm, which abstractly speaking is characterized by innovative features like rationality, autonomy, pro- and re-activeness, distributedness and interactivity. As it will be shown in the following paragraphs, these features prove to be especially useful when it comes to integrate distributed information resources.

Here is an example in the context of our travel-planning problem, an agent would be a kind of mediation software between the user and the Internet. The agent would act on behalf of the user with the task of synthesising a travel plan. The user would interact with the agent and specify his travel-planning problem by defining his travel variables and constraints. The agent would autonomously search for relevant travel information and decide in a rational way about the means of transport, the hotel and meeting dates. Finally the agent would propose a consistent and optimised travel plan to the user.

Such a single agent model can be extended. The information resources themselves can be agents. For example, the travel agency can be represented by an agent who provides travel information, reserves and sells tickets and negotiates ticket prices. Other agents again can represent the personal agendas of my working colleagues and the online hotel booking system. At the end a dynamically and temporarily bound agent society has solved the complex travel-planning problem.

Dealing with different information resources is problematic. A single information representation standard in the Internet does not exist. The information providers are free to choose amongst numerous languages, protocols and formats. One travel agency might use plain html pages whereas another agency uses a Java program. Moreover, the semantics of information in the form of symbols, terms and in general all kinds of signifiers can differ significantly, especially when it originates from different sources. As it is with the natural languages, an isolated term or signifier can be rather ambiguous.

The introduction of agents offers a number of advantages. In our example the agents translate the heterogeneous and contextually loaded information into an unambiguous, and standardised format, which can be unambiguously shared. Another advantage of introducing agents originates from their autonomous, rational, social, proactive and reactive features. These features are desirable in dynamic, uncertain and unpredictable environments where goals can fail and alternatives are required. In order to cope with such an environment an agent typically integrates reasoning capabilities, generates plans and pursues one or more goals. Such "intelligent" features are vital when a temporarily compiled agent society collaborates in order to solve a problem, which was never an explicit part of the individual agent design.

The notion of using an entire agent society for solving complex and distributed problems can be projected and applied in many other cases. This particular research work deals with the problem of integrating a distributed and heterogeneous organisation through an agent society. This research is based on a real world example and concerns the integration of the globally distributed project collaboration ALICE. From the organisational point of view ALICE is a massive project co-ordination problem.

The research concentrates on three core integration problems. Firstly, it deals with the problem of translating heterogeneous, contextual and especially culturally loaded project information into a neutral format to become an integral part of the agent communication language. The translation is achieved by expressing project information by a constraint programming language.
Secondly, the research deals with building an ALICE specific project ontology. The project ontology provides a dictionary of terms that the agents use for modelling their specific co-ordination problems. The project ontology has a unifying function as it also forms the basis of the agent communication language.

Thirdly, the research focuses on methods to achieve co-ordination within a distributed problem context. The co-ordination is negotiation based and ruled by a protocol, which is modelled according to the ALICE context.

The scope of this paper however is limited it does not cover all three integration issues in full depth. The objective of this paper is to address the translation process. In particular, a simple project planning ontology is developed where the co-ordination specific, natural language terms are formulated as constraints. Based on this ontology the complete ALICE project co-ordination problem is centrally modelled and solved. With the help of three example cases it is demonstrated that the terms of the project ontology are culturally loaded and can be translated through constraints. The last sections of this paper are concerned with the co-ordination itself. Features of the negotiation protocol are discussed by reflecting the ALICE specific co-ordination context. A simple example shows how the negotiation based co-ordination process can be modelled in terms of CSP formalisms. Future research work will deal the other integration problems in greater depth.

2. ALICE - A Real World Example of Integrating a Distributed and Heterogeneous Project Collaboration

ALICE is a large collaboration for particle physics and brings together more than 80 institutes from 40 countries. The goal of the collaboration is to develop and construct a large particle detector for the LHC project at the European Laboratory for Particle Physics in Geneva. The detector is a massive and complex device, weighing several thousand tons and integrating thousands of mechanical and electrical components. The size of the project is comparable to that of an Apollo space mission and is project management challenge [3,4].

From the definition and organisational point of view, ALICE is a collaboration of equal institute partners. A central organisation, authority or power structure is not present. The participating institutes are loosely coupled through joint activities, common resources and precedence constraints. Within CERN, a small co-ordination team tries to integrate and co-ordinate the institutes. When conflicts cannot be solved on the institute level, the co-ordination team acts as a mediator. Relevant conflict information is collected and analysed and then a decision is taken according to priorities.

"The Memorandum of Understanding" document defines the legal basis of the collaboration. It includes the list of the participating institutes and funding agencies. It defines what each institute and funding agency is expected to contribute in terms of detector components and financial means. However, the MoU does not go into further detail concerning rules and regulations of the collaboration. For example a protocol or statement on how to proceed in conflict situations or what the priorities of ALICE are is not included. In fact, nowhere is it written. Most conflicts are resolved in small groups at a personal level and the solution is usually based either on tacit rules or on rational reasons.

The overall detector design is modular, which allows for a clear task division amongst the institutes and this means a reduction of dependencies. Nevertheless, the integration of all sub-projects and groups into a single project and the management and distribution of common resources requires intensive co-ordination work.

Especially the ALICE installation phase is very complex and requires intensive pre-planning and co-ordination. During the installation phase the institutes will almost concurrently deliver all their modules in thousands of components to CERN to become assembled in the underground hall. At the same time the installation of another three detectors of the same complexity interferes with ALICE. It is evident that without a detailed installation schedule the installation phase is likely to end in chaos.

As the institutes themselves are directly responsible for the design of their modules, they also deal directly with the management and financial planning of their project. They receive finances from independent national funding agencies. The contribution of all institutes is then listed in terms of products and finances in the Memorandum of Understanding document. CERN, which has the role of centrally co-ordinating the project, contributes manpower, common resources and a small money fund. The institutes act to a large extent independently they can be conceived as smaller organisational cells with an inherent culture and with organisational and administrative structures. This fact adds a great degree of heterogeneity to the project, which makes it difficult to co-ordinate and coherently relate the sub-projects as a whole.

Also the institutes prefer to stay private and reluctance to share their knowledge with others makes the coordination work difficult. The institutes fear that once information is made public it becomes official and must be guaranteed. It is a great risk to promise the completion of a system for a certain date when too many uncertainties are present. Somebody else could rely on published data and use them for his own project plan. An official project plan also means ammunition to lead discussions, criticize performance, reveal problems, find weak points and can indicate a lack of management.

Nevertheless, if ALICE wants to be coherently managed, some degree of knowledge sharing is absolutely necessary. Knowledge sharing in general leads to a better understanding of each other's needs and tasks. It improves mutual trust. Knowing what other groups are doing also allows for creating new knowledge, especially in those areas where conflicts, ambiguities and redundancies exist.

The individual goals of the institutes can differ significantly and in the worst case they can be in opposition. For example, institutes fight over detector space. The way that measurement devices are physically built into the detector defines how well the particles can be measured but it also determines how much other devices are disturbed. Conflicting goals also exist in relation to the use of common resources such as the common money fund, CERN manpower, tools and equipment. In order to keep costs low and to complete tasks on time, the institutes try to get as much manpower as possible from CERN. But CERN manpower is limited and cannot cover everybody's needs. When a conflict is detected the institutes will first try to solve the conflict themselves. A consensus is usually achieved through direct negotiation. If no consensus can be achieved, the co-ordination team is activated. The coordination team will then discuss the problem with the involved parties and take a decision based on ALICE's priorities. A decision concerning the detector design for example is most likely to be taken by the ALICE co-ordination team. A conflict involving common resources or joint activities can be solved through direct negotiation among the institutes.

Problem solving is made more complex as ALICE is a multi-cultural melting pot. People in the collaboration are coming from many different nationalities, ethnic and religious groups. A great variety of professions is required and the collaboration employs physicists, engineers, computer scientists, technicians, managers and many other specialists. Based on these cultural differences people can be distinguished and grouped by certain factors; in particular by knowledge, values, ideals, habits, world-views, concepts and interests. These cultural factors have a great influence on how problems are analysed, processed and solved. For example, a group of physicists would design a detector differently from a group of engineers. The engineer's design would typically emphasize the technical aspects of the detector, for example the construction, safety and use of materials. Physicists in contrary would design the detector towards the best physics performance and easily neglect safety and realisation questions. There are other areas where cultural differences do matter a lot. For example, the working hours in the south of Europe are different from those in the north of Europe. People in Spain are open to the idea of a siesta during a working day, but not the British. Some countries have a 35 hour week while others work for more than 40 hours a week.

In the context of ALICE the cultures of the institutes play an important role. Each institute has its own organisational methods, structures, procedures, concepts and views, which greatly influences the way that project plans are developed. Some institutes are team oriented, while others are resource aligned. Some institutes break down their project into a hierarchical structure of products, systems and components. Others however describe their project by breaking it down through tasks and activities.

3. ALICE Project Planning as a Constraint Satisfaction Problem

Essentially, the integration of the ALICE project with regards to the coordination of activities and resources, is a classic scheduling problem [5]. The delivered project plans are based on activities, which are sequentially linked through precedence constraints and temporally assigned to resources. Additional global constraints determine project boundaries, and different objective functions define how the project can be optimised.

Solving the scheduling problem essentially means finding a project plan, where all constraints can be satisfied concurrently. Secondly, it means choosing amongst all feasible plans the one plan, which is optimal with regards to the stated objective functions.

Most scheduling problems, as it is with this one, are complex, require combinatorial optimisation and belong to the class of problems, which is called NP-complete [6]. Different techniques can be used to tackle NP hard problems and one of the preferred ones with high efficiency is that of constraint programming [7,8,9].

Constraint programming is used to model the ALICE scheduling problem. A central CSP is established in order to prove two things. First of all, that the problem can be formulated and modelled as a CSP and secondly, that a solution actually exists. Stating and modelling a CSP involves three steps. The first step is to identify a set of domain variables, which in our model represents activities and resources. Thereafter possible values and value ranges are assigned to the domain variables. In our model for example, the start and end date ranges for activities. The second step is then to state the constraints between the domain variables like precedence constraints or no-overlap constraints. The last step is then to solve the problem and to find a set of variable values, where all constraints can be concurrently satisfied. For the case that objective functions are declared, the search also includes a search for the optimal solution amongst all consistent solutions.

The actual CSP model was programmed using OPL Studio from ILOG [10].

The CSP model

Domain Variables

The main ALICE scheduling objects are activities and resources. Formulating the CSP under OPL Studio, the activities are represented by three domain variables: the start and finish time, and the duration. The initial domain for the start and end times is usually the entire range from project start to project end. The duration domain is usually instantiated to a fixed value.

The resource objects fall into three categories; unary, discrete and reservoir resources. Unary resources are resources, which when occupied by an activity, cannot be occupied concurrently by another activity. Discrete resources have a capacity and can be used by many activities concurrently as long as the total capacity sum never exceeds 100%. Reservoirs are resources where the capacity can change over time. For example, budget accounts are typical reservoir resources; funding money is paid in and the money is spent for paying activities. When declaring a resource, only a name and if available a capacity is necessary. The resource domain variable is not explicitly modelled under ILOG-OPL studio but handled internally. By stating, "*Activity_A requires unary Resource_A*", OPL creates a link between the activity and resource and takes care that another activity using the same resource cannot overlap.

Constraints

The constraints of the CSP model fall into two categories.
Implicit Constraints handled internally by OPL:

1. *Relation between start, finish and duration*: Start+Duration=Finish
2. *No Overlap*: Two or more activities may not overlap in time, if they use a unary resource or occupy more than 100% of a discrete or reservoir resource.
3. *Project Start / Finish:* Each activity of a project can only start after the project starts and must finish before the project ends.

Explicit Constraints formulated in OPL.

1. *Precedence Constraints*:
   FS - Activity B cannot start before activity A finishes.
   SF - Activity B cannot finish before activity A starts.
   FS - Activity B cannot finish before activity A finishes.
   SS - Activity B cannot start before activity A starts.
2. *Container Constraint*: If activity A contains the activities B and C, then activity A starts before B and C start and finishes after B and C finish.
3. *Milestone:* A milestone is an activity with duration=0.

4. *No Overlap*: Activity A cannot overlap with activity B
5. *Start,Finish:* Activity starts or finishes at a certain date.
6. *Start/Finish before/after:* An activity starts/finishes before or after a certain date

Solution Search

The solution search is under the control of OPL studio and can be done in two different ways. The search can be about finding only a consistent solution or the search can be about finding amongst all consistent solutions the solution, which is optimal with regards to an objective function. The objective function can envisage different goals and in this case is about minimizing the total project length.

Simple ALICE Project Planning Ontology

OPL-studio offers a set of scheduling objects such as activities and resources, and a set of constraints, for example; precedence and capacity limits. However, the offered objects are too basic and too limited to cover the entire modelling demand. It is necessary to extend the available OPL scheduling objects and construct new ones. Due to the versatility and flexibility OPL offers, this can be easily achieved.

Through a central analysis of the scheduling problem and formulating it in terms of objects, their relations, global and local constraints and scheduling objectives, a list of terms and rules are obtained. These terms and rules, if not already part of OPL-studio, are then formulated with the basic OPL objects. This list is a kind of lexicon or terming it more sophisticatedly, a project planning ontology, which translates concepts, terms, relations, etc. into domain variables and constraints.

The following table contains a simple Project Planning Ontology for ALICE. The list is not complete but contains the essential model elements. The first column of the table contains the set of terms, which signify the scheduling objects. The second column contains the scheduling object attributes. The third column lists the basic OPL objects, which are used to model the customised scheduling objects. Column four lists the domain variables of the OPL variables. Column five contains the implicit constraints, which are handled internally by OPL.

Term	**Attribute**	**OPL Component**	**OPL-Domain Variable**	**Implicit Constraints**
Activity	Object_Type	Activity		
	Name		-	
	Start		Start = [date1..date2]	Activity.Start + Activity.Duration
	Finish		Finish= [date3..date4]	=Activity.Finish
	Duration		Duration= [int1..int2]	
	Manpower		Manpower=[0..int5]	
	People		People=[0..N]	
Container	Object_Type	Activity		
	Name			Start+Duration=Finish
	Start		Start = [date1..date2]	
	Finish		Finish= [date3..date4]	
	Duration		Duration= [int1..int2]	
	Manpower		Manpower=[0..int5]	
Milestone	Object_Type	Activity	Start=[date1..date2]	Start+Duration=Finish
	Name		Finish=[date1..date2]	
	Date		Duration=0	
Team	Object_Type	Unary Resource		activity1 requires (Team1)
	Name			$\wedge$ activity2 requires (Team1)
				$\Rightarrow$Activity1.finsih<=activity2.start
				$\vee$ Activity1.start>=activity2.finish
Manpower	Object_Type	Discrete Resource		$\sum$ Activity..Manpower (t)
	Name			<=Manpower.Capacity
	Capacity			$\forall$ t $\in$[project start...project end]

Tab.1: Simple ALICE project ontology.

Term	Variables	Constraint
Activity_Manpower	Activity.People Activity.Duration Activity.Manpower	Activity.People*Activity.Duration=Manpower
Container_Manpower	Contained(Activity.Manpower) Contained(Container.Manpower)	Container.Manpower=∑ Activity.Manpower + ∑ Container.Manpower
Container_Start	Contained(Activity.Start)	Container.Start<= Min(contained(Activity.Start))
Container_Finish	Contained(Activity.Finish)	Container.Finish>=Max(contained(Activity.Finish))
Finish2Start	Activity1 Activity2	Activity1.Finish <= Activity2.Start
Activity_Requires_Unary_Resource	Activity Resource	
Activity_Requires_Discrete_Resource	Activity Resource; Quantity	

Tab.2: Additional constraints modelled with OPL.

Results

The ALICE scheduling problem is modelled and coded as a central CSP in ILOG OPL studio. The execution and tests are successful. Solutions for the stated ALICE scheduling problem exists and if the number of variables are not too great, the optimal solution can be obtained within an acceptable amount of time.

4. Translating Cultural Differences with Constraints

Before dealing with the cultural differences in the context of ALICE's specific problems, the term "culture" needs a general explanation. The term "culture" is commonly used but its meaning is diverse; a uniform and generally accepted definition does not exist. Nevertheless, the modern culture-anthropology [11] has a definition which is broad and generally accepted: "*Culture is considered as a system of concepts, convictions, attitudes and values, which become apparent through the behaviour and actions of people as well as their intellectual, spiritual and material products*."

With this definition the existence of many different types of cultures become apparent. One can speak about the culture of a nation, the culture of an ethnic group or the culture of a religion. Of course culture can also be found on much smaller scales. Companies, organisations, clubs, interest groups, families or even people with the same education or profession can represent a culture. In this way every individual is ultimately part of many cultures and the influences are overlapping. Which culture finally dominates certain behaviour is not obvious and depends very much on the context of the situation.

Looking at ALICE's culture, the professional, national and culture of the institutes are the most significant. Since this research addresses the integration of ALICE through agents representing the institutes, the culture of the institutes must be foregrounded. All other cultures, which are significant, become subsumed into the institute's culture, thus creating the institutes overall culture.

The next question to ask is how can culture be modelled? As it was indicated in previous sections culture influences many things. Culture influences the meaning of signifiers like speech, language, words, terms, symbols etc. Culture influences our way of solving problems, it influences our way of thinking, behaviour and value system. Culture is present in almost everything that cannot be seen to be an individual factor. Culture also means that a choice or an alternative is present. For example, if the entire world agrees to a 40 hour week, then the *working times* would not be a cultural factor. However, the *working times* do differ from country to country and is therefore a function of culture. So instead of assuming "*working time=40h*" it would be better to state the working times to be a function of the culture and assign it to a domain range: "*working time (culture) = [30h..50h]*".
Many more of these examples exist in the context of ALICE and they really do matter. In the following

paragraphs, the cultural differences are demonstrated and the effects discussed. It is explained that instead of introducing a central planning concept it is better to accept the heterogeneities and to deal with the distributed planning. The following three examples will show that culturally loaded project information can be translated into an unambiguous and exchangeable format by using domain variables and constraints.

The first step in identifying cultural differences amongst the institutes is to analyse their individual project schedules. Differences exist in the applied data structures, planning constructs and the semantics of commonly used terms. It is clear that the project schedules cannot be merged without translation and standardisation. The introduction of a central project co-ordination would be extremely difficult since the institutes would need to find a common determiner as to how to do project planning. They would also need to agree on a single meaning of used planning terms. Such a central concept is artificial and stands in strong opposition to the reality that ALICE is a multi-cultural collaboration without a central authority. The institutes themselves have evolved organically and have their own unique organizational structure. Such a central planning concept would definitely lead to hard discussions and poor acceptance.

Instead it is better to leave the responsibility and development of the project plans at the level of the institutes and accept any heterogeneities. The institutes remain as closed and private organizational cells and are represented by an agent. The project coordination is achieved through the agents, which exchange co-ordination details. A central agent represents the role of the CERN co-ordination team and solves difficult co-ordination problems at a higher level.

The basis of the co-ordination and communication process is a dictionary of terms and rules. As each agent has his own understanding of what the used ontological planning terms actually signify, the meaning needs to be translated. The translation can be achieved by formulating a term through neutral domain variables and constraints. By using domain variables and constraints as "meaning carrier", co-ordination information can be unambiguously exchanged and directly related to the internal CSP models of the agents. In this way, the platform of the communication and co-ordination processes among the agents is a combination of shared project planning ontology and constraint programming. The following three examples demonstrate this notion.

A *Milestone* is a central entity for project planning and co-ordination. It is first of all used for signalling the state transition of the project and means an achievement. Secondly it is used for triggering new project phases and for checking on the project performance. At the ontological level we define a milestone object to have a name, a date and a domain. The date specifies when the milestone is supposed to be achieved. The domain specifies when the milestone is accepted and not viewed as late. This leaves room for shifting milestones forwards and backwards in time.

The two institutes, Orsario and Dubioso have to test their equipment together. They both keep a milestone in their plans, which indicate that their equipment is ready for joint testing. As soon as both have agreed on the date, they exchange their milestones. Their understanding of what the term milestone signifies differs.

Institute Orsario defines the milestone to be a strict deadline and a commitment to increase efforts and even resources if the milestone cannot be achieved on time. It arrives at the following milestone definition:

Milestone (Orsario):

Name: "Marm ready for testing with institute Dubioso"
Date: d1= 17th January 2003
Start: [Project Start..d1]

Institute Dubioso defines the milestone differently. A milestone is not a deadline, but a simple co-ordination point. The constraint definition for the milestone is:

Milestone (Dubioso):

Name:"Dipmag ready for testing with institute Orsario"
Date: d1= 17th January 2003
Start: [Project Start..d1..Project End]

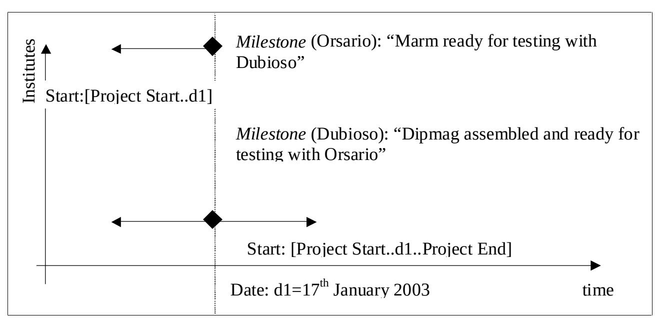

Fig. 1: The term "Milestone" and differences in meaning.

When institute Dubioso receives the strongly constrained milestone from institute Orsario, it believes with high certainty that the milestone will be achieved in time.

Hence, Dubioso can plan for testing the modules at the agreed date.

When institute Orsario receives the weakly constrained milestone from institute Dubioso, it is uncertain whether the milestone will be on time. The commitment is too weak. Planning for the testing activity requires further information from Dubioso. Orsario can backtrack and ask Dubioso about the progress of all those tasks, which are the condition for achieving the milestone.

The co-ordinator agent, who is receiving both milestones, could integrate the information in the central co-ordination plan. He could then reason about the consequences and determine the latest date by when the testing has to be done. If this date is earlier than the right domain boundary given by Dubiosos, the co-ordinator agent would constrain the milestone of institute Dubioso, for example, by changing the right domain boundary or by linking it to an activity.

Translation can also be applied for quality features like *late*. Let us suppose to deal with an activity with 1 month duration. An institute coming from country X would define this activity as late if the real finishing date was 10% beyond the planned finish date. Another institute coming from country Y would say an activity is late if it was 30% beyond the planned finish date.

The definition late does not necessarily have to be relative to the activity duration. One could also define *late* to be an absolute value. For example, everything, which is more than 1 month delayed, is considered to be late.

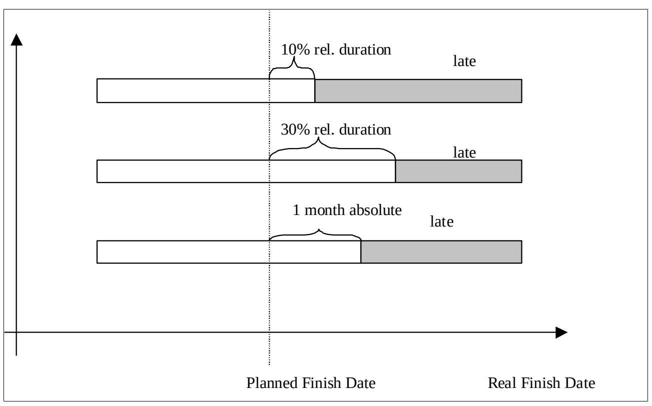

Fig.2: Translation of the quality term" late".

What can we gain from translating the quality term *late*? The term *late* basically defines the tolerance of an institute towards delays. It can be translated by adding the 10%, 30% or 1 month to the domain range. For example, if the institute was specifying:

late: related: relative to duration
amount: 30%

activity: Name: "Testing"
Duration: 20
Finish: [20..100]

Then what they would really mean in terms of the constraint formulation is:

activity: Name: "Testing"
Duration: [20..26]
Finish: [20..106]

What are the consequences of this translation process? With the definition of the term *late,* we can specify the uncertainty of an activity. Instead of a fixed value, the duration becomes a domain. The finish domain is also changed and becomes extended to 106. Therefore, an institute, which has a high tolerance towards delays, will introduce a greater duration domain range than an institute having a smaller tolerance. When calculating schedules, the best case and worst case scenarios can be computed by either using the lower duration bound of 20 or the upper duration bound of 26.

The third example concerns differences in the applied planning constructs and optimisation goals. Institute Hempido wants to present its project plan as a flat list. The management decides to form 3 task teams, red, green and blue and manages the project in a team-oriented way. One of the restrictions is that a team can only work on a single activity at a time. Each activity has got a constant duration and needs one or more teams for completion. The activities are connected through precedence constraints. The schedule is optimised to finish every task as early as possible. Hempidos project schedule looks like the following:

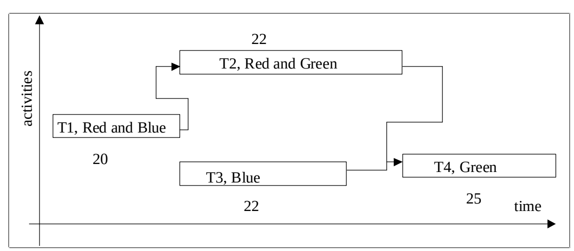

Fig. 3: Project schedule of institute Hempido.

The described problem is a standard resource-scheduling problem, which can be easily modelled with our previous project planning ontology and coded with OPL.

Object instantiation

Activity instantiation:
Activity ("T1", d=20), Activity ("T2", d=22), Activity ("T3", d=22), Activity ("T4", d=25)

Resource instantiation:
UnaryResource("Team_Red"),
UnaryResource("Team_Blue"),
UnaryResource("Team_Green")

Object relations

Precedence Constraints:
Finish2Start("T1","T2"),
Finish2Start("T2","T4"),
Finish2Start("T3","T4")

Resource allocation:
RequiresResource("T1","Team_Blue"),
RequiresResource("T1","Team_Red"),
RequiresResource("T2","Team_Red"),
RequiresResource("T2","Team_Green"),
RequiresResource("T3","Team_Blue"),
RequiresResource("T4","Team_Green")

Scheduling Goal:
minimize (max (T1.finish, T2.finish, T3.finish,4.finish))

Institute Orsario has a different approach and decomposes the project with a product and a work breakdown structure. The work is only stated as a fixed manpower value, which means flexible start times and duration. Manpower is calculated by multiplying duration and number of people. As Orsario is hiring manpower from an external company and can call up to 20 people, it wants to keep the hired number of people as constant as possible. The Project schedule can be adjusted by varying the activity duration and start times. The other commitment and goal of Orsario is to finish the project at the given milestone date.

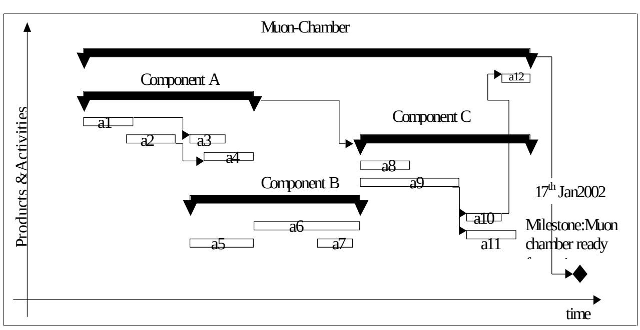

Fig.4: Project schedule of institute Orsario.

Object instantiation

Activity instantiation: Activity ("a1", mp=20), Activity ("a2", mp=12), Activity ("a3", mp=30), Activity ("a4", mp=7), Activity ("a5", mp=10), Activity ("a6", mp=23), Activity ("a7", mp=8), Activity ("a8", mp=11), Activity ("a9", mp=14), Activity ("a10", mp=4), Activity ("a11", mp=3), Activity ("a12", mp=5)

Container instantiation: Container ("Component A"), Container ("Component B"), Container ("Component C"), Container ("Muon Chamber")

Milestone instantiation: Milestone ("Milestone: Muon chamber ready for testing"**,** "17th Jan 2002")

Resource instantiation: DiscreteResource("People",20)

Object Relations/ Constraints

Activity-Container Relation:
Activity_Inside_Container ("a1","Component A")
Activity_Inside_Container ("a2","Component A")
….

Precedence constraints:
Finish2Start (a1, a3), Finish2Start (a9, a10), Finish2Start (a9, a11), Finish2Start (a10, a12), Finish2Start (Component A, Component C), Finish2Start ("Muon Chamber", "Milestone: Muon chamber ready for testing")

Resource allocation:
RequiresResource("a1", "People", integer(mp_{a1}/d_{a1})) ,
RequiresResource("a2", "People", integer(mp_{a2}/d_{a2})) ,
…
RequiresResource("a12", "People", integer(mp_{a12}/d_{a12}))

Scheduling Goals

Scheduling goal: minimize (Σ difference (Σ activity manpower(t) - Σ activity manpower(t+1))

Comparing the two projects in view of their planning constructs they differ significantly. We find a hierarchical versus a flat project breakdown structure, a product versus an activities based project and the use of unary versus discrete resources. Despite those conceptual differences of the models, which makes it rather difficult to integrate them into a single, central CSP model, the models are still CSP's, sharing the same fundamental modelling entities, domain variables and constraints. As we will see later, this fact makes it actually easy to relate and co-ordinate the models with each other.

We have demonstrated three cases as to how cultural differences can be translated. These cases concern the translation of terms, quality, planning constructs and goals. The list of what can be translated is not complete, as many more cases have been found.
Translating the cultural differences with constraints, the global project model becomes more individual, realistic and transparent. False expectations are removed and the agents can implement their own meaning of terms, project model and scheduling goals.

5. Project co-ordination through constraint based negotiation

In relation to a multi agent system, co-ordination essentially means to ensure that a community of agents acts in a coherent manner [15]. In the context of ALICE, coherence means how well the institutes behave as a unit and not conflict with one another.

Conflict potential in ALICE exists due to the decentralized project structure where the individual institutes have no global view of the project, act rather independently of each other, do not share their knowledge and are self-interested. ALICE also has to satisfy a number of global constraints, for example, agreed budget frames cannot be exceeded. This forces the institutes to co-ordinate their expenditures as a whole.

However, the biggest conflict potential exists in those areas, where the institutes have dependencies and interdependencies. These exist in particular for shared resources and linked activities.

Literature shows that co-ordination of distributed problems is complex and can be achieved in many different ways. The work of Yokoo [16] shows that the co-ordination of a DCSP can be achieved through a distributed search algorithm. Jennings and Nwana [15] have studied diverse literature resources on co-ordination techniques and discussed their relative advantages and disadvantages. One of their conclusions is that a universal best co-ordination method is not available and that the choice of the right co-ordination technique is really problem dependent. However, based on their observations and experiences, they point out that a comprehensive co-ordination technique should include at least the following four components.

It should first of all provide structures, which regulate the interaction amongst agents in predictable ways. Secondly, it should be flexible enough for the agents to deal with dynamic environments and to survive with incomplete and imprecise knowledge about other agents. Thirdly, social structures must exist, which regulate how agents behave and relate to each other. Fourthly, the agents must implement sufficient knowledge and reasoning capabilities in order to exploit the social and individual structures and behave flexibly.
After studying the various co-ordination techniques, negotiation based co-ordination is the most suitable method for ALICE. Negotiation is a key co-ordination technique and has become a research topic on its own. Negotiation is characterised by the following advantages:

- Negotiation allows for privacy and does not force the agents to exchange their entire knowledge and preferences.
- Negotiation is not limited to a central or de-central co-ordination process, both can exist concurrently.
- Negotiation reduces the network traffic and avoids bottlenecks. The agents exchange only co-ordination relevant, partial information and not complete plans.
- A negotiation protocol de-couples agents from one another. De-coupling reduces complexity and increases flexibility. De-coupling is also a necessary condition when dealing with heterogeneous and incompatible agents.
- Binding the agents through a negotiation communication process, it gives them the freedom to develop their own private strategies and methods within the protocol limits as to how to negotiate. This fact contributes to real world modelling for a heterogeneous and self-interested agent society.
- Negotiation also allows new agents to join the society easily.

Numerous negotiation techniques were developed and again there isn't a single universal best method. Some negotiation techniques are anchored in the Game Theory [19], others are plan-based [20] and again others are motivated by human negotiation strategies [21,22].

Bussman and Muller [18] have defined negotiation as: *"…negotiation is the communication process of a group of agents in order to reach a mutually accepted agreement on some matter."* Sycara [14] argues that for effective negotiation the agents must reason about beliefs, desires and intentions of other agents. Explicitly it means that in effective negotiation, the agents keep belief models of the other agents and reason about them in order to influence their intentions and beliefs.

In the following, we do not want to adopt and improve a known negotiation technique but rather discuss what features an efficient technique in the given ALICE context should have. Based on those cognition results, a negotiation technique for ALICE will be the subject for future research.

Abstractly speaking, negotiation based co-ordination consists of three key elements, a communication, a reasoning, and a negotiation protocol element. Those elements are separate enough to be considered in isolation.

The objective of the negotiation protocol is to define a common rule platform, which regulates the negotiation process and is designed towards efficiency, effectiveness and fairness. The reasoning element is a process, which is carried out by the agents. They reason about the received negotiation information and apply and relate it to their own beliefs, desires and intentions. The result of the reasoning process is a re-action, which can have an inward and an outward component. The inward component affects the internal state of the agents with regards to his beliefs, desires and intentions. The outward component is the response to the received information, basically the response message.

The communication element of the negotiation process is essentially the pure information and data exchange among the negotiating agents. The communication itself has to be structured and must be based on a common language and communication protocol.

Negotiation Protocol

A suitable negotiation protocol for ALICE should contain and address the following features:

- The protocol needs to define the scope of the negotiation process, in particular what the negotiation objects are and what features can be negotiated.
- The protocol needs to define and regulate the negotiation phases, for example conflict definition, conflict resolution and agreement.
- The protocol should define limits in order to ensure that co-ordination process converges; for example timeout limits if the negotiation process does not progress.
- The protocol needs to motivate the agents to deliver precise, detailed and reliable plans, which ultimately increases the quality of the project plans. Good quality should be rewarded, bad quality should be punished.
- The protocol should discourage the agents to be dishonest, for example by imposing penalties in case an agent lies for getting advantages.
- The protocol must introduce a currency and a value system. “Virtual money” in terms of points is used for paying penalties, rewarding for qualities and favours etc.

Reasoning

A negotiation process between two or more agents will start when mutual conflicts arise; for example two agents concurrently request the same unary resource. In this case, the agents need to coherently co-ordinate their schedules, analyse and reason about the situation and find a consensus. The reasoning process itself is an individual matter and each agent will implement his own strategy within the protocol limits. For example, an agent could first check if his project plan is flexible enough to allow an easy adaptation within the given constraint frame. If this fails and none of the plans are easily adaptable, another negotiation and reasoning round would start, which would then address the relaxation of constraints.

The reasoning must not be limited to only a search for conflict solutions, but could also address how to optimally use the negotiation protocol and achieve advantages. It could address how an agent can influence the other agents so that the interference of their plan is minimized and that availability of common resources maximized. Most likely in this scenario the agents start building a knowledge model of the other agents’ beliefs, goals and preferences as further reasoning material. This kind of behaviour, which goes into the direction of manipulation, reflects the reality in ALICE where the institutes have a tendency to behave in a self-interested way and sometimes even in an egotistical manner.

Taking self-interestedness to an extreme, the agents could even be programmed in such a way that they behave dishonestly and present false facts in order to carry through their schedule and maximize benefits and advantages.

Communication

The basis of the communication process must be a language, which is shared and can unambiguously be understood by everybody. As our agents already formulate their scheduling problem as CSP’s, it is self-suggesting to use directly the constraint language for the information exchange and negotiation. This solution bears the advantage that the exchanged, constraint-based information fragments can be easily weaved into every agent’s CSP model and that it is directly applicable and processable without further treatment. Moreover, the cultural differences would be already an implicit part of the exchanged constraint information and need no further translation.

Co-ordinator Agent

If all agents were benevolent and truthful and the constraints would always allow them to find a solution within acceptable time limits, the agent society could solve the co-ordination problem without external support. However, in a real world situation, problems are likely to appear. The co-ordination problem can be over-constrained by which a solution can only be found if the constraints become relaxed. The co-ordination problem can also be constrained to such a level, where a solution cannot be found within acceptable time limits; again constraint relaxation is due. Moreover, relaxing constraints is a delicate operation. Not every constraint can be relaxed and constraint relaxation is usually limited. The original problem can be easily shifted to become a constraint co-ordination problem. As the agents can also implement self-cantered or egotistical behaviour and in the worst case can be designed to lie, the agent society can easily get out of balance. External arbitration and control mechanisms seem to be necessary.

In order to prevent such situations, a special co-ordination agent is introduced. This co-ordination agent has two main functions. Firstly he helps to solve difficult co-ordination problems. He has the trust from all institute agents and is allowed to read all planning information. From that he compiles a central constraint model, which allows him to efficiently compute schedules or to reason about constraint relaxation. Secondly the co-ordination agent has the task to protect the protocol and prevent the agent society from anarchy. As he has access rights to each agent's schedule he can detect unfairness and dishonest agents. He basically tries to keep the balance in the co-ordination process.

6. The negotiation in terms of the CSP formalisms

This paragraph now illustrates with a simple example how the negotiation-based co-ordination can be implemented using the previously introduced CSP formalism and protocol:

Two institutes agents have a conflict. They both concurrently request a unary resource. Their CSPs are formulated as follows:

<u>Agent A:</u>

Activities: ***A1***:(Duration: [20], Start:[0..30])
A2:(Duration:[30], Start:[0..50])

Precedence Constraints:
Finish2Start(***A1***,***A2***)
Resources: **R1**
Resource Request:
RequiresResource(***A1***,***R1***)

<u>Agent B:</u>

Activities: **B*1***: (Duration: [30], Start: [0..40])
B2: (Duration:[40], Start:[0..70])
Precedence Constraints:
Finish2Start(***B1***,***B2***)
Resource Request: RequiresResource(***B1***,***R1***)

Based on the CSP model the agents compute their schedules and optimise them towards early completion. The instantiated schedules look as follows:

Agent A: A1.start=[0], A1.duration=20
A2.start=[30], A2.duration=30
RequiresResource(***A1***,***R1***)

Agent B: B1.start=[0], B1.duration=30
B2.start=[20], B2.duration=30
RequiresResource(***B1***,***R1***)

Agent A is the owner of resource **R1** and detects the conflict. He sends a request to Agent B to inform him about the conflict:

A -> B: conflict(A1(0,20),B1(0,30),R1(A1),R1(B1))
Adapting one of the schedules means for both agents a delay and therefore is to their disadvantage. Both agents are reluctant to change schedules and prefer the other party to do so. However, the agent who is finally willing to change his schedule, will be rewarded. The other agent must pay him in points. The price for the change is determined by the amount of delay. For example, shifting an activity by 10 time units is equal to 10 points. Now the negotiation process starts as to who is changing the schedule. First, both agents must send the flexibilities of their schedules, which are basically the domain ranges of the conflicting activities.

A->B: dom_range(A1[0..30])
B->A: dom_range(B1[0..40])

After the agents have exchanged their domain ranges they are expected to make an offer.

B->A:
offer_A_1(A1(0,20),B1(20,50),R1(A1),R1(B1),price=20)
A->B:
offer_B_1(A1(30,50),B1(0,30),R1(A1),R1(B1),price=30)

Then, it is up to the agents to accept the offer of the other agent. If one agent accepts, then the negotiation process terminates. If both agents accept, then the cheapest offer is chosen; in our case it would be the offer of agent B. If both agents do not accept their mutual offers, then they can post new offers. If after a number of offers none of the offers is accepted, the cheapest offer under all offers is chosen.

It is possible, that in a conflict situation the given constraints do not allow a solution at all. In this case the constraints itself need to be relaxed. Again this can be done through negotiation. However, relaxing constraints is a delicate operation. Not all constraints can be relaxed and relaxing constraints can threaten a re-computation of the entire schedule.

7. Conclusion

ALICE is a massive project and distributed and heterogeneous due to a decentralized and multi-national organizational structure. The collaboration exists of equal members and a central power structure or authority is missing. A small co-ordination team supports the integration of the project with regards to project management.

Amongst the collaborating institutes exist interdependencies in view of common resources, shared activities and task sequences. Coordinating the ALICE project is basically a scheduling problem and belongs to the class of NP-hard problems. The search for a consistent solution is a complex task as many things are coupled, dependent and interdependent.

After viewing the ALICE scheduling problem and establishing a project planning ontology, the scheduling problem was centrally modelled as CSP. Then the model was coded in OPL Studio, a constraint programming language from ILOG. The development and programming of the central model was successful. This proves that a consistent solution for the given scheduling problem actually exists. If the number of model variables is small enough, an optimal solution according to an objective function is obtainable.

An analysis of the distributed, multi-cultural project environment with regards to project management showed, that the semantics of commonly used terms differs significantly. In order to enable unambiguous communication amongst project members, the meaning of used terms needs translation. Three cases demonstrated that such translation could be achieved by formulating the term semantics by domain variables and constraints, which then can be unambiguously exchanged.

Instead of solving the scheduling problem centrally it is more realistic to leave it distributed and co-ordinate the sub-projects. Such coordinating can be achieved through different techniques such as organizational structuring, a contract net, multi agent planning or coordination through negotiation. Negotiation is the most suitable technique because it maximally de-couples the agents, allows for privacy, is not limited to central or de-central processes and provides maximum flexibility.

Coordinating the problem through negotiation basically involves three elements, a negotiation protocol, reasoning capabilities and communication. The negotiation protocol determines negotiation rules and limits, it motivates and rewards quality and it discourages the delivery of poor quality and dishonesty. The reasoning element is context dependent and different for each agent. Based upon the reasoning process the agent compiles his own negotiation strategy. The reasoning can be on various aspects, for example about the schedule, how to optimally use the protocol and maximize advantages, or how to influence other agents in their beliefs and goals. Since the reasoning process is a private matter it leaves space for dishonesty.

The basis of an efficient communication process needs to be a common language. It is proposed to base the communication process on domain variables and constraints. Both are already used for formulating the project schedules and are therefore interchangeable, and carry already the implicit culturally loaded meaning.
A co-ordinator agent is required when the co-ordination task amongst the agent fails and a central solution must be computed. This agent also ensures that the agents' behaviour conforms to the protocol and that the progress of the agent society stays balanced.

A small example shows how the negotiation-based co-ordination can be implemented with the CSP formalisms of section 5. When conflicts appear the agents' exchange and negotiate about their domains in order to resolve the conflict.

8. References

[1] B.Faltings, *"Problem-solving agents in the Internet using Constraint Programming"*, 1999.
[2] M.Woolridge, N.R. Jennings, "Intelligent Agents: Theory and Practice", Knowledge Engeneering Eeview,10 (2), June 1995.

[3] H. Kerzner. "*Project Management A Systems Aproach to Planning, Scheduling, and Controlling.*", 6th Edition, 1997.
[4] G.Bachy,A.P.Hameri."*What to be implemented at the early stage of a large-scale project*", Int.Journal of Project Management Vol15, No4, pp.211-218, 1997.
[5] N.M.Sadeh ,S F Smith, "Knowledge based production management", 11th European Conference on Artificial Intelligence (ECAI), August 1994.
[6] Garey M, Johnson D: "Computers and Intractability: A Guide to the Theory of NP- Completeness", W.H. Freeman, New York (1979).
[7] Barták, R., "Constraint Programming: In Pursuit of *the Holy Grail*", Proceedings of WDS99, Prague, June 1999.
[8] LePape C: "Constraint-Based Programming for Scheduling: An Historical Perspective", Working Paper, Operations Research Society Seminar on Constraint Handling Techinques, London, UK.
[9] Zweben M, Fox M S, "*Intelligent Scheduling*", Morgan Kaufman, San Mateo (1994).
[10] P. Van Hentenryck. The OPL Optimization Programming Language. MIT (1999).
[11] G.Maletzke,"*InterkulturelleKommunikation*", West - deutscher Verlag, 1996.
[12] N.R. Jennings, "Commitments and Conventions: The Foundation of Coordination in Multi-Agent *Systems*.". The Knowledge Engoineering Review, 2(3): 223-250,(1993).
[13] Jennings N R: "Coordination Techniques for Distributed Artificial Intelligence", G.M.P. O'Hare and N.R. Jennings, editors, Foundations of Distributed Artificial Intelligence, pp187-210. John Wiley & Sons, Inc,, New York, (1996).
[14] Sycara K:"Multi-agent Compromise via negotiation" in Gasser L and Huhns M (Eds):"Distributed Artificial Intelligence 2", Morgan Kaufmann(1989).
[15] Nwana HS, Lee L, Jennings N R: "Co-ordination in software agent systems", BT Technol J Vol 14 No 4, (Oct 1996).
[16] M. Yokoo, "*Asynchronous weak-commitment search for solving distributed constraint satisfaction problems.*", In Proceedings of the First International Conference on Principles and Practice of Constraint Programming , pages 88-102, Springer-Verlag,(1995).
[17] Werkman K J:"Knowledge-based model of negotiation using shareable perspectives", Proc of the 10th Int Workshop on DAI, Texas (1990).
[18] Bussmann S, Muller J: "A Negotiation Framework for Co-operating Agents", in Deen S M (Ed): "Proc CKBS-SIG", Dake Centre, University of Keele, pp 1-17 (1992).
[19] Rosenschein JS, Zlotkin G: "*Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*", MIT Press (1994).
[20] Alder M R, Davis A B, Weihmayer R, Forest R W: "*Conflict resolution strategies for non-hierarchcal distributed agents*", in Gasser L and Huhns M N (Eds): "Distributed Artificial Intelligence 2", Morgan Kaufmann(1989).
[21] Raiffa H: "The Art And Science Of Negotiation: how to resolve conflicts and get the best out of bargaining", Harvard University Press (1982).
[22] Kennedy G, "Pocket Negotiator: How to negotiate successfully", The Economist Books (1993).