CHAPTER 4

A conciliator-coordinator for concurrent engineering

Irène DEGIRMENCYAN[1,2], Arnaud FOUSSIER[1], Patrice CHOLLET[2]
[1]Département Intelligence artificielle, DASSAULT AVIATION
[2]Laboratoire de Mécatronique du L.I.I.S.I

The design of any complex system requires the cooperation of many specialists. In order to optimize his own part, each specialist ignores the overall problem. So the work is done iteratively, conception duration is lengthy and solutions are often difficult to transfer to production.
We have developed a software called **DIAMANT** (Dialogue InterActif pour la Mécanique, l'ANalyse et la Technologie) or Interactive Dialogue for Analysis, Mechanics and Engineering (IDAME) dedicated for managing interactions between the designers working on an unitary projet. These interactions allows information transfer and above all dialogue when conception conflict occurs : **DIAMANT** has a coordinating function. **DIAMANT** helps in resolvong those conflicts : it has a conciliatory function.
Using the results of our previous research we can propose a methodology for solving technological problem by means of constraints resolution : **JADE** (Jeu d'outils d'Aide à la **DE**cision technologique) or Decision Helping Tools for Engineering (DHTE) . This software helps the designer in engineering problems formulation and is a guidance tool during solutions search.
This chapter first presents the use of **JADE** during formalization and resolution of problems by the designer acting alone . Secondly the structure and the resolution process of **DIAMANT** is developed, based on the cooperation of designers helped by **JADE** during the conception.

4.1. INTRODUCTION

Technology, in the french mechanical conception acceptance, is the formalizing of engineering tools.As systems to be created are complex , many specialists have to

R. Soenen et al. (eds.), *Advanced CAD/CAM Systems*

participate in synchronization. Everyone wants to optimize his own part, so the global point of view about the system is forgotten by individuals. It is the project manager's job to coordinate the tasks and to avoid conflicts.

The subjet of our research is the study of this problematic and we have developed a software tool called **DIAMANT**.The purpose of this tool is to manage interactions between designers working on an unitary project and to give help for the global design.

DIAMANT is based on the formulation constraints of individual engineering needs.This is done by defining engineering features. Helping the designer is done by assisting him in solutions search. A resolution methodology of engineering problems is put forward with **JADE** developments. It is a tool for constraints resolution developed in the scope of an army contract DRET # 89 472 [BOU 90] **JADE** ensures that the know-how put into the design of a component is not lost. **JADE** also helps the designer guidance during the solutions search phase by the restriction of the solution domain. A **JADE** prototype is in operation at Dassault Aviation Artificial Intelligence Department.

The dialogue formalization between different designers during the simultaneous conception using **DIAMANT** can be defined as the coordination of designers' actions and conciliation in case of conflict. **DIAMANT** features belong to simultaneous and concurrent engineering :

1 **a shared architecture** : each conception problem is identified as a functionnal feature and modelized in JADE. This one can be linked to a geometrical pattern maker as CATIA for an immediate visualization of changes in the drawings.

2 **Communication and coordination between designers:** DIAMANT system manages communications between the different JADE stations allowing designers a dialogue and information propagation resulting from the influence of designers' decisions and of a designers choice.

3 **Coherency and conciliation between designers:** relations analysis between variables and taking into account constraints linking the problems gives a global coherency of choices or proposals for conflict management.

DIAMANT is an army contract **DRET # 91 34 589**. It is developed at **Dassault Aviation** Artificial Intelligence Department in collaboration with the **L.I.I.S.I** of the **ISMCM.**

4.2. ENGINEERING FEATURES

4.2.1. Aims

JADE's purpose is to give a decision-making help to research department designers facing up to , for example, dimensioning of mechanical parts or assembly problems.

For that, it is necessary to implement :

1 a framework model of conception knowledge which enables information capture on various engineering problems [GRO 87] [GRO 88].

2 an adaptable and powerful process for the handling of this know-how giving a real help to designers.

We have developed an easy to use system which every engineer can use an adapted formalism.

Conception know-how is called *engineering features* which group together :

1 entities modelization handled as objets or attributes (see Figure 1)

2 conception constraints description as equations, inequations or rules.

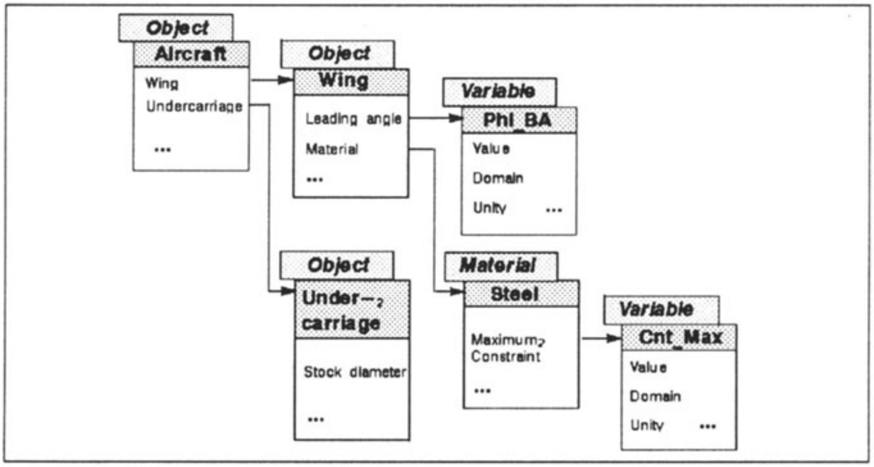


Figure 1. aircraft structure features

4.2.2. Methodology

General references [KRA 87][MUR 87][KIM 86][OHS 89]

Methods used during the last few years for constraints solutions follow four main axis:

1 Data generation and constraints validation : it is the classical way used in research departments where designers view solutions and modify them intuitively still converging to a realistic solution of the problem. This method is generally lengthy.

2 Multi-criteria optimization (Pareto Domain) : constraints are treated with an optimization algorithm searching optima of an objective function in which the criteria are multiplied by coefficients varying between [0.0....1.0] [RAD 85]. The amount of results and the number of criteria (problem size) make the solution choice very difficult.

3 Constraints fulfillment : constraints are organized as a graph and every choice is propagated through it for deducing other variable values [HOW 86] [LEL 88] . With this method we cannot predict any solutions.

4 Data domain calculation : using constraints programming languages as CHIP,PROLOG,PECOS or CHARME we can calculate, for each variable, data domain .This was done taking constraints in account a priori [HEN 89]. These programming languages are limited to linear constraints and do not solve problems with real variables.

Conception is also a very unstable process : constraints evolve by addition or removal. So the constraints system coherency as to be managed in a way of detecting redundant or opposite constraints [SER 87].

Our approach is like the constraints resolution techniques and consist of presenting to the designer solutions domain for every variable belonging to the problem.We work on linear or non-linear constraints with real variables.

We have developed a tool called **JADE** for engineering features formalization and resolution help.

We have also created a linkage between the algebraic model , which group together the engineering features and the geometrical model representing the geometric design problem. This geometrical model is defined using **CATIA** , a CAD/CAM software.

4.3. JADE

4.3.1. Description

JADE is a design-help tool, whose functions are:

1 description of conception problems expressed in terms of constraints :

- 11 linear and non linear equations or inequations without differential forms;
- 12 boolean rules **IF** (conditions) or **THEN** (conclusions) allowing to select active constraints, in accordance with system state.

2 taking into account the designer's **intentions**:

- 21 constraints included in the schedule of conditions
- 22 conception choices

3 exploring of the whole domain of solutions related to a design problem

JADE proposes a process based on a symbolic resolution (adaptability and general nature of processing) and numerically (quickness and precision of the data processing). All solutions appears to the designer as domain shapes upon variables. He can chose values in these domains , being informed at every moment about consequences on other variable domains. The designer works in restricting possible solution domain.

We have developed a **resolution kernel** for calculation of domains for the variables. With it we can optimize variables values of the problem under constraints and obtain, for each variable, a range of values where we can find problem solutions. This resolution kernel is equiped of a data deduction and inference on rules modulus.

JADE's strong points :

1 works as an algorithm generator as it replaces a collection of specific algorithms used in design

2 variable domains are constantly updated. The user can choose the evaluation order of his variables, always being sure of a solution (contrary to some engineering calculation systems where the designer has to respect an evaluation order and learns at the end of the job that there is no solution according to previous evaluations);

3 the user can go back on his choices and evaluate many times the same variable for adjusting, for example, its value according to the effect on the other variables.

The solutions domain is presented to the user as a body of domains for the problem variables : we called it the *space solutions descriptor*. JADE was validated through several examples: Spring or jack dimensioning, aircraft pre-dimensioning, et frame positioning in an aircraft structure.

4.3.2. Use of JADE

Two main steps (see Figure 2)

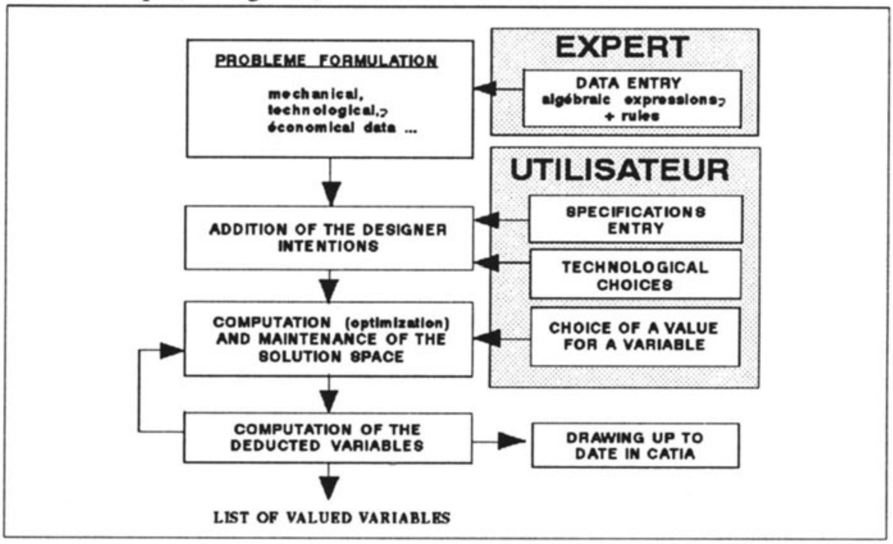


Figure 2. the different steps of resolution in JADE

1 First the definition of the problem by an **expert** :
Data capture by an expert using pop-up menus (loading or resolution of a problem, solving an equation,...) . Through a capture questionnaire, the expert informs the software about the value attributes of the algebraic feature (variable form, type of equation,...).This expert knowledge is expressed and manipulated with a data base oriented object , these objects representing the different entities used in the problem (equations, inequations, rules, variables,....). That are grouped together in families(classes) having common characteristics (attributes) and functionalities(methods)
So the expert enriches the **JADE** problems library as well .

2 Secondly the resolution of the problem by the user :
21 First, he selections the problem in the **JADE** library , then he defines the specifications for the product to be elaborated, and finally defines the value a first set of variables.
22 Then a formal sequence is initiated (calculation of partial derivatives used later in the resolution kernel). The resolution kernel then determines the solutions space as domains of the variables.
23 The descriptor of all the solutions is displayed. The user choose what variable he wants to fix and capture the value, according with the solution domain. The resolution kernel is initiated for calculation of variables deducted from the previous choice and from the still free domains.
24 The user evaluate all the variables of the problem, one after one. But he can go back on his choices, releasing some variables and re-evaluate them. All the deductions induced by the released variables are re-calculated.

4.3.3. Links between the geometrical model and the engineering model

Engineering technical features include, by nature, geometrical features. **JADE** can be showed round a geometrical pattern maker (**CATIA**) for problems with a geometrical nature (see Figure 3). At every variable of the engineering model we can associate one parameter of the geometrical model (see Figure 4). During the process, **JADE** brings up to date the geometrical model at every evaluation of the variables. The modification of the drawing shows the user the effect of his decisions.

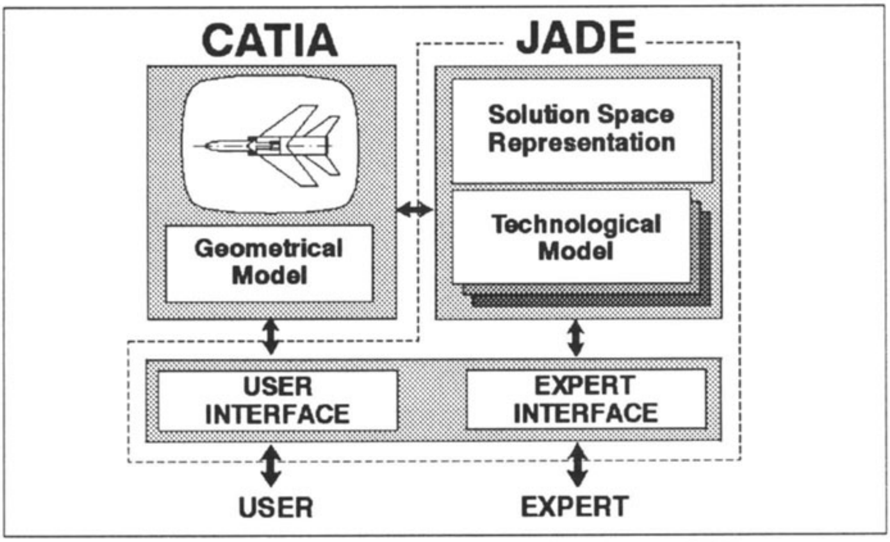


Figure 3. The interface between CATIA and JADE

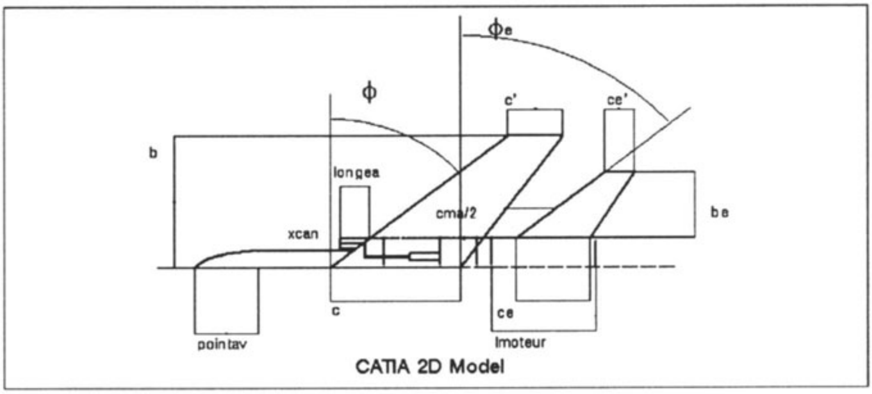


Figure 4. An example of the predimensioning in JADE

4.3.4. Example: pre-proportionning of an aircraft

This example is a simplified design pilot study where the geometrical constraints of the wing area and the engineering rules are taken into account (motorization, radar type,engine type,...)

Variable	Designation	Unity	Constraints
c	aircraft axis rope	m	$E = \frac{c'}{c}$ $\frac{2b_e^2}{s_e} = l_e$
c'	external wing rope	m	
s_{ref}	surface of reference	m^2	$E_e = \frac{c_e'}{c_e}$ $\frac{2s_e}{s_{ref}} = 0.22$
b	wing spread	m	$\frac{4b^2}{s_{ref}} = \lambda$ $\frac{l_e}{l} = 0.85$
l	wing extension	m	
ϕ	wing leading edge arrow	degré	$l_{ong\_alim} - l_{ong\_a} < 0.0$
E	wing tapering		$2.5 < c < 4.5$
c_e	socket rope	m	$1.0 < c' < 2.0$
c_e'	external horizontal feathering rope	m	$25 < s_{ref} < 60$
s_e	horizontal feathering surface	m^2	$3.0 < b < 5.0$

Figure 5. Variables and constraints description

b_e	horizontal feathering spread	m	$2.5 < l < 4.0$
l_e	horizontal feathering extension	m	$40.0 < \phi < 50.0$
ϕ_e	feathering leading edge arrow	degré	$1.5 < c_e < 2.5$
E_e	horizontal feathering tapering		$0.4 < c_e' < 1.0$
p_{ointav}	length of the aircraft head	m	$1.5 < s_e < 6.0$
$l_{ong\ a}$	distance between the plane of the air entries and the wing leading edge	m	$1.5 < b_e < 2.5$ $2.0 < l_e < 3.0$
$l_{ong\ alim}$	limit value of $l_{ong\ a}$	m	$0.4 < l_{ong\ a} < 1.5$

Figure 5. Variables and constraints description

If motorization of aircraft is **single−engined,** Then $\phi_e = \phi$, $E_e = 0.3$	If **radar** of aircraft is type_A , Then $p_{oint\_av} = 1800$
If **motorization** of aircraft is twin−engined, Then $\phi_e = 50°$, $E_e = 0.5$	If **radar** of aircraft is type_B, Then $p_{oint\_av} = 1900$
Si wing of aircraft is low, Alors $l_{ong\_alim} = 0.8$ m	If engine of aircraft is type_1, Then $l_{moteur} = 3.2$ m
If wing of aircraft is hight, Then $l_{ong\_alim} = 0.4$ m	If engine of aircraft is type_2, Then $l_{moteur} = 4.0$ m

Figure 6. Technological choices rules

JADE resolves this problem in approximately 15 minutes with back-tracking.

JADE is a tool dedicated to one designer's use and needs. However, complex systems are used by more than one designer so we have studied a multi-agent system for design help.

4.4. COORDINATION AND CONCILIATION

4.4.1. Aims of the problem

To define a resolution process for complex engineering problems where several expertises and designers act simultaneously. Each expertise is treated independantly by **JADE** for solving a part of the complex problem. But since many conception tasks occur , linking constraints exist between the tasks [MIT 91][DEL 90].

We define as **specific constraints** those related to a design task and as **linkage constraints** those related to task linking.

All these constraints, of specific or linkage type, are not taken into account at the same time by every user. So, in order to guide the designers to a global solution of the problem it is necessary to :

1 coherency between the different solution domains
2 coordination of design choices
3 management of a conciliation between designers.

4.4.2. Retained architecture

We envisage a distributed multi-designer environment [CHI 91] made up of several **JADEs**, each one being autonomous and interactive [LFF 91], and of a system managing the cooperation between designers [KIM 91][CAM 88][SMI 88].

In the **DIAMANT** architecture planned here, all the **JADEs** are equivalent entities and their interactions are managed by an upper-entity called **Coordinator/Conciliator**. This entity manages information transfer between all the **JADEs**, and the coherence between the different JADEs and conflicts between designers [KLE 89][LAA 91][TEN 88](see Figure 7).

4.4.3. Communication protocols

The general architecture of **DIAMANT** and the resolution scenario necessitates communication needs between the differents **JADE** and **DIAMANT** .

The bi-directional character of communications between **DIAMANT** and **JADEs** are the direct consequences of the cooperative approach of the **DIAMANT** project. **DIAMANT** is based on a cooperation between the different engineering resolvers and a conciliator. A cooperative structure between applicators is characterized by a dynamic "giver of orders" framework and an important exchange of asynchronous messages between the differents contributors: every applicator can send or receive messages at every moment.

Communication needs for **DIAMANT** are modelized by an asynchronous bi-directional exchange protocol of characters strings.

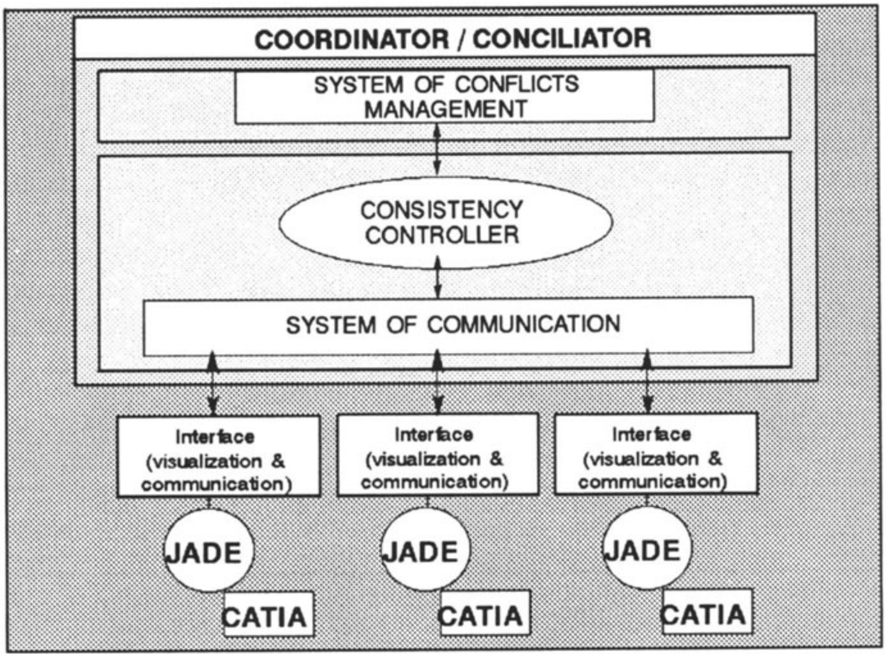


Figure 7. Global architecture of DIAMANT system

4.5. DIAMANT

4.5.1. Linkage constraints

4.5.1.1. Formalization of linkage constraints

The linkage constraints create links between the different **JADEs**. They are expressed as equations, inequations or rules where variables appear. Those variables are called linkage variables, coming from the different **JADEs**. They have an influence on variable domains and also on specific variables, which are those not included in linkage constraints.

As in **JADE** we want to propose solution domains on variables to the designers. But in **DIAMANT** we introduce the global coherency problem: domains have to be a solution to the whole design problem, so they have to be computed from all the constraints. Precise domains of conception variables are linked to those obtained if all the constraints are putted in an unique **JADE**. A distributed architecture permits all designers to be autonomous and not turn their habits upside down [CUT 89].

When each **JADE** computes its solution domains classicaly, it does not take in account linkage constraints. These domains are larger than the exact ones which represent the solution space of the whole problem. To reduce these domains, we have to used the limitations induced by the linkage constraints. So we have directed our research in determining exact domains with a distributed structure for the **JADE** and using the linkage constraints.

4.5.1.2. Constraints localization

Two methods have been explored for linkage constraints localization:

1 **DIAMANT** computes domains as the others **JADE** do. The linkage constraints are located in **DIAMANT**.

2 Add the linkage constraints to the different **JADEs**.

We have retained the second solution because there are less agents and therefore computation precision is better.

4.5.2. Domain calculation methods

4.5.2.1. Field coherency

Each JADE includes its owns linkage constraints, and links them to the other JADEs. It manipulates in each case supplementary variables without the designer knowing. Linkage variables are present in several **JADEs**. So we obtain several domains for the same variable. Domains intersection restricts for a first ti the space of the solution. But it is necessary to iterate with all the **JADEs** because the initial spaces have evolved. We stopped the iterations when each domain cannot be restricted anymore.

At the end of the computation, there are two occurences:

1 one domain at least is void, there is no solution which verifies all the constraints

2 no domain is void : they include all the solutions, but a zone of the domain may not be a solution. Only supplementary iterative calculations can eliminate the zones of the domains which are not solutions.

Variables domains calculated by a **JADE** are the edges of a hyper-parralelepide ex-inscribed to the volume space.This is an appproximation of the solutions space.When we intersect the different hyper-parallelepipeds, we do not obtain a good approximation of the solution space. The hyper-parallelepiped obtained by intersecting hyper-parallelepipeds is larger than the hyper-parallelepiped ex-inscribed resulting from the intersection of the solution volume (see Figure 8).

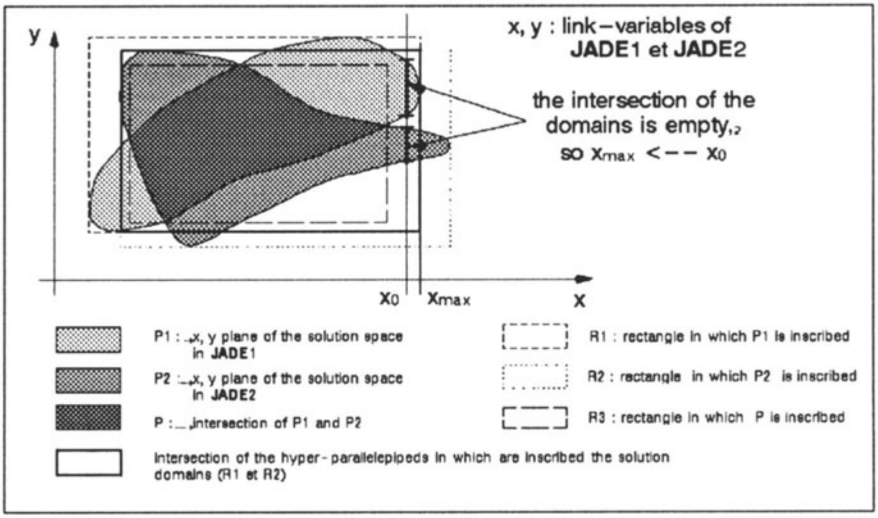


Figure 8. An example of the intersection of the domains

We have studied two methods for computing exact domains : by faceting, by scanning.

4.5.2.2. By faceting

For improving the precison of the hyper-parallelepiped intersection, we have to improve the approximation of the solution space. Every solution space given by a **JADE** is faceted by a hyper-polyedron with parallel faces in pairs. Each normal being a linear combination of variables (ex : ax+by+cz = 0 a, b and c reals, see Figure 9).

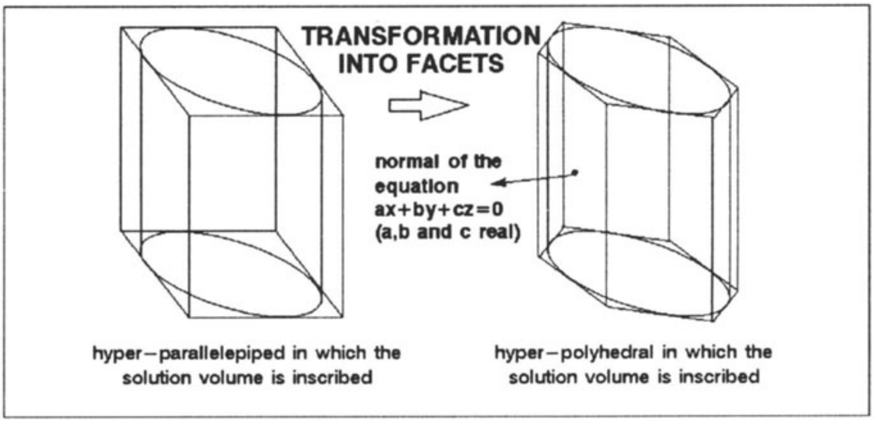


Figure 9. An example of a three-dimensional facet transformation

We obtain, for each **JADE**, a set of inequations of type: m < ax+by+cz < M (a,b,c,m,M reals) representing an approximation of the solution volume. Making the intersection of all the hyper-polyedrons joins all the inequations obtained with the JADEs. Then we have only to determinate the optima of each variable under the constraints. We obtain a more precise intersection of the solution domains.

4.5.2.3. By scanning

We try to reduce the hyper-parallelepiped intersection in order to obtain the hyper-parallelepiped taken from the real intersection of solution domains.

Scanning is done variable by variable: we reduce the hyper-parallelepiped intersection following the chosen direction, fixing step by step the variable corresponding to the chosen direction. Then we optimize all the other variables with all the constaints of the different **JADEs**.

We obtain as many hyper-parallelepipeds as **JADEs** :

i if their intersection is void, we can conclude immediatly that there is no common solution. For the value of this variable,we dont find values for the other variables which respect all the constraints of all the **JADEs**.We can increment the fixed variable and iterate the process (see Figure 10).

ii if their intersection is not void, we cannot immediatly make a conclusion. Even if the intersection of the hyper-parallelepipeds computated by the **JADE** is not void, this does not necessarily mean that the solution domain intersection is void.We cannot say that there is common solutions at the **JADE** set for the value of the fixed variable.We continue by fixing one more variable and we are now at the beginning of the calculation with one dimension less.

The process ends at the dimension 1, where we can conclude the following two factors because intersections of spaces or domains are identical :

1 either domain intersection on the last variable is void and we go back to the previous variable (§i.)

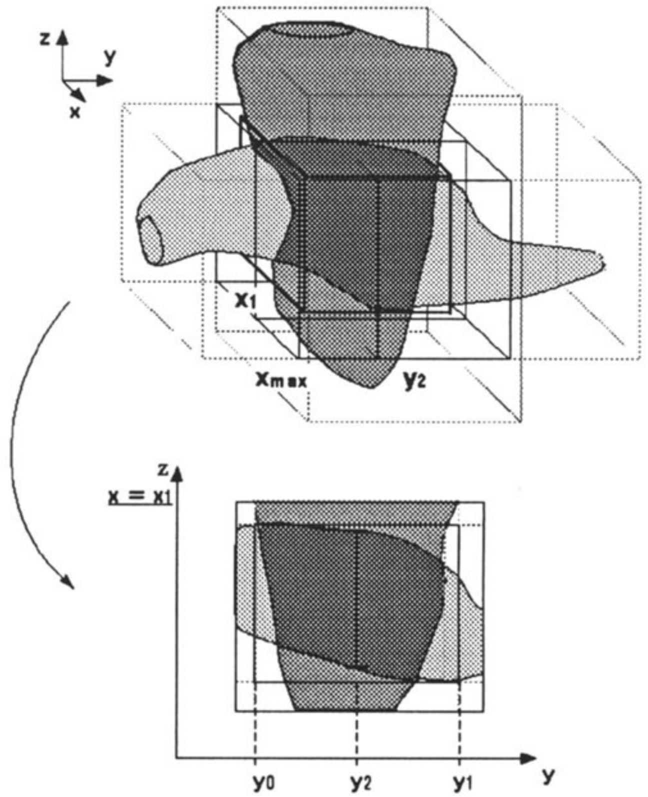
2 or we have at least one solution for the instanciation set and we obtain the minimum or the maximum limits of the first instanciated variable.

We made a scan every time we want to determinate the miniùum or maxmum limits of a variable.

The disadvantage of these methods leading to the determination of precise domains is the computation time. With the computers we use , it is better to aim at less precision, so computing time is acceptable. A larger estimate of domains is not a handicap if we choose to take into account this problem in the resolution process.

4.5.3. Resolution scenario

For user-friendliness, users can decide o the instanciation state in which their problem will be taken in account by **DIAMANT** for coherency processing. So they are not constrained in their solution search and are not disturbed during an instanciation period or during tests.



The plane corresponding to the valuation of x to x1 is not enough to end, so we valuate y to y2 (dimension 1).

The domain finded for z is not empty, so we can end that for $x = x_1$ and $y = y_2$ the problem has solutions and then that the real maximum value of x is between x_1 and x_{max}.

Figure 10. An example of the pass of a three-dimensional scan

4.5.3.1. Expertise formulation

The formulation of the expertise is carried out in each **JADE**. In describing the technological and functional constraints and the functional and geometrical constraints.

4.5.3.2. Local coherency processing

Each **JADE** begins a computation sequence for reducing the domains on its variables until they only contain values corresponding to solutions of the problem formulated by this **JADE**. This said processing is local because it takes into account, for every **JADE**, only its own constraints.

4.5.3.3. Linkage constraints formulation

When all the **JADEs** have a formulated and coherent problem in their memory, linkage constraints are generated :

1 either from geometrical models : the project manager or each designer points out on the whole geometrical model the geometrical linkage constraints (facing functional surfaces, minimum thicknesses, dimensioning to be respected ...),

2 or from technological models, adding constraints of other types.

4.5.3.4. Domain reduction

We realized an intersection of all the domains determined by the different **JADEs** for each linkage variable (variable occuring in linkage constraints). The new domains are reintroduced in every **JADE**. Every **JADE** compute again the domains on the set of variables. We reiterate the calculations of the intersections until the stability of calculated domains.

If any intervals are void, this indicates a formulation problem. In this case, resolution is impossible and we have to look again for a new formulation. Nevertheless, the system may detect what constraint(s) induces the problem. For example, in studying the constraints one after another, or searching coherent sub-problems with a maximum of constraints, that are important sub-systems with solutions.

4.5.3.5. Resolution period

All the **JADEs** now have reduced and coherent intervals. The designers can begin the resolution period, that is to say the evaluation of the variables in the proposed intervals. Each user decides when he communicates to others the modifications he creates (interval reductions, choices or deduction of values) : a proposition is sent to **DIAMANT** constituing of a group of information on the linkage variables. The information of the group are the values of the instanciated variables and the domains for the other variables. We note this group in the following way :

$$E = \{...(var_i, val_j)... + ... (var_j,dom_j)...\}$$

DIAMANT then transmit the sub-sections of this group to the other designers. **DIAMANT** only shows the specific variables of each designer. Each **JADE**

consequently recieves a group of information which it must compare with the current state of its resolution. The designes work in an asynchronous fashion so they are immediately informed of the arrival of new data. But they only start to study the information when they wish to, by example when they cuurent state is stable (end of domain calculation,end of hypothesis study...)

The new data are taken into account one after the other and beginning by the instanciations. Figure 11 shows a flow-chart of treatment of information recieved by **JADE**.

The first step before adding new data in the designer environment consisits of asking him is agreement. The designer can refuse the data if this does satisfy him. We define this conflict as a type **C1**.

If the designer gives his agreement, the information is added at the current state of the resolution in order to determine its influence. A full calculation of the domains is launchedto verify that the information from DIAMANT is contratidictory with the information contained within JADE. The calculation of the domain intersection has taken place. This calculation replaces in part the calculation of "the real domain" which we have previously defined the limits of.

If the domains do not intersect (there exists at least one void intersection) it is the second conflict called **C2**. In the contrary case the designer must validate the deuctions which appear when we take into account the equality constraints. The conflict called **C3** can appear.

The process reiterates until the data sent by the coordinator is treated.

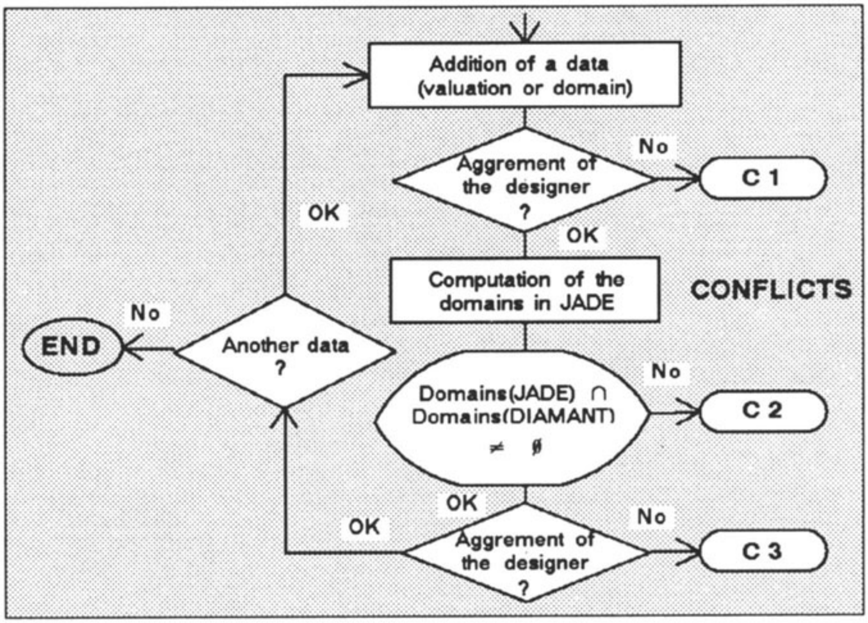


Figure 11. The flowchart of the addition of the coordinator data

4.5.4. CONFLICT MANAGEMENT

4.5.4.1. Introduction

Resolution conflicts appear in all tha cooperative systems [KLE 89] [LAA 91]. Inevitably the hierarchy problem occurs. In the course of the resolution, following the moment certain **JADEs** can have a more important weight than others. Rather than imposing a hierarchy between designers, we find it preferable to define the resolution strategies of the conflicts enabling the guidance of designers when the resolution seems "blocked" (that is to say when the designers do not arrive at converging to a global solution.)

The coordinator can only hope to resolve a conflict if the designers **propose at least one alternative** of solution research for the variable which poses a problem.

The designer must give indications on the values (val_i) or the domains [min_j, max_j] refused. These indications are presented under the form of limited or non-limited intervals :

1 For the instanciated variables :

The refusal of the instanciation var_i <- val_i implies the necessity of the counter-proposition of an interval [val_min_i, val_max_i] does not contain val_i (one or the other of the limits not being able to be known) (see Figure 12)

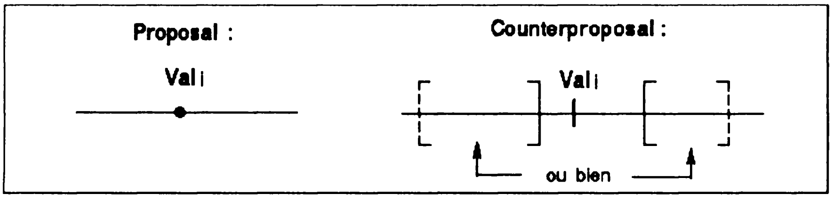


Figure 12. A counterproposal of a designer in the case of an unsatisfactory value

2 For the variables of which only the domains are known:

The refusal of a domain implies the necessity of the proposition of another domain being able to intersect the precedent but not englobaling it entierly (see Figure 13)

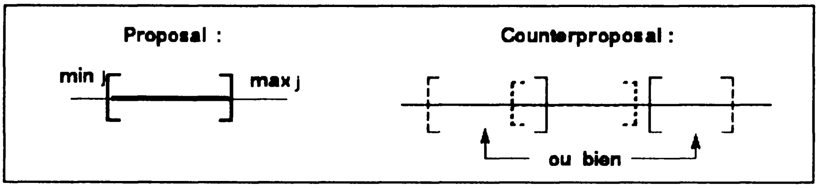


Figure 13. A counterproposal of a designer in the case of a unsatisfactory domain

It should be noted that the confrontation process of the sent information takes place, in parallel, with all of the designers. Consequently, in the case of refusal of one of the designers, the refusal and the counter-proposition are stocked by the the coordinator in the wait for decisions (refusal or acceptance) of other designers. It treats the information group and transmits the synthesis to the designer at the origin of the proposition who can therefore modify his proposition.

4.5.4.2. Conflict case study

Case C1

The designer is unsatisfied by at least one of the data transmitted by the coordinator. He must communicate to the coordinator the reason for his refusal and a qualitative counter-proposal ("increase A" for example) and/or quantitive (new value or new domain).

The knowledge of communal constraints and those of designers enables **DIAMANT** to determine the correlation between the variables. **DIAMANT** can therefore point out the variables which it is possible to modify. So we modify the value of another variable in the wished direction.

When the designers make choices which intervene in the linkage constraints and which are refused, so the reseach of a compromise between two designers is extendable to other designers. On extending the dialogue, it is often easier to find solutions.

Case C2

The intersection of domains delivered by the coordinator and those calculated by **JADE** are void. We consider that the intersection is void when :

1 the two domains, non-reduced to a value, have a void intersection,

2 The instanciation of a variable sent by the coordinator does not belong to its domain calculated by **JADE**,

3 A value deduced by **JADE** or fixed by the designer does not belong to the domain transmitted by the coordinator.

In the last two cases, for resolving the conflict, it is necessary to modify the value and/or the domain :

1 Modification of the domain

It is necessary to determine the constraint(s) which intervene in the modification of a limit. The variables which intervene in this constraint must therefore be modified. If they are not instanciated or deduced, we extend the research to other consraints, until we arrive at an instanciated variable or the initial limit fixed by the expert.

2 Modification of the value

The designer can modify the variable in the neighbouring area of the fixed variable. The visualisation of the sign of partial derivatives of the constraints enables the designer to orientate himself in his resolution. As soon as he has found a compromise which satisfies him, he informs the coordinator.

When at least one of the intersections of the domains calculated by **JADE** and of those transmitted by the coordinator is non void, either the designer returns to case C1, or he informs the coordinator that the domain which has been transmitted to him must be modified. **DIAMANT** therefore determines the **JADE(s)** responsable for the limitation of the domain and their transmittal of the modification demands.

Case C3

The case C3 is analogous to case C1 in the point that the designer indicates a conflict following the decisions of the coordinator; the designer refuses a posteriori a data after the analysis of its effect on the problem that it treats. The detailed actions for case C1 are therefore appliable.

4.5.5. Modelling

4.5.5.1. Application

The application in course of study is in the aeronnautical domain : air conditioning for embarked electronics equipements.

Our interest for this example is created by :

1 a multi-knowledge character adapted for the concurrent engineering of **DIAMANT**. Three types of knowledge occurs:
2 designers for determining the performances of the air circuits,
3 furnishers for dimensioning circuitry components for the required performances,
4 design department for integration of the whole set in the aircraft pod
5 demonstrative and visual character.

DIAMANT is still under study. Model realization for one or several examples gives us the opportunity to enlarge upon and illustrate the capabilities of software help in simultaneous and concurrent design. Firstly, taking into account design constraints created by different knowledges and secondly by an easier dialogue between designers for the research of solutions.

4.6. CONCLUSION

We have described here our research carried out during the last years in design helping. Engineering feature concept introduced in **JADE** gives an homogeneous formalization of design knowledge. The domain solutions descriptor is a powerful tool which discards the need for computing sequences and facilitates solution searching. A **JADE** prototype is used in the Artificial Intelligence Department of **Dassault Aviation.** Its job is to integrate a CAD-CAM software such as **CATIA.**

DIAMANT gives us the opportunity of extending the accumulation of design know-how to complex organizations such as project teams integrating simultaneous and concurrent design. A feasability modelling of **DIAMANT** is planned for June 1995 on workstation (IBM and Sun) in the object-orientated development environment SMECI (ILOG Ltd).

4.7. BIBLIOGRAPHY

[BOU 90] A. Bourne, A.Clément, A.Foussier, J.Saulais, M.Sicard : JADE : un jeu d'outils d'aide à la décision technologique, Outils et application de l'Intelligence Artificielle en CFAO p 148-161 ed Gardan, Hermès 1990.

[CAM 88] S. Cammarata, D. McArthur, R.Steeb : Strategies of Cooperation in distributed problem solving, Reading in distributed artificial intelligence p 102-104

[CHI 91] R.S Chin, S.T. Chanson,: Distributed Object based Programming System, ACM Computing Surveys, Vol 23 # 1, March 1991

[CUT 89] M.R. Cutkosky, J.M. tenenbaum : A methodology and Computational Framework for concurent Product and Process Design, Mechanism and Machine theory, 1989.

[DEL 90] J.P.Delmas, C.Houellebecq, P.Mazas, D.Ouvry, P.thoraval : Systèmes experts et CAO : ARCHIX, un système d'aide à la conception d'un véhicule en phase avant projet, Micad 90, p 458-574, 1990

[GRO 87] M.D. Gross, S.M. Erwin, J.A. Anderson, A.A. Fleisher : Designing with constraints, Computability of Design, eds Yehuda & Keley, John Wiley & Sons, 1987

[GRO 88] M.D. Gross, S.M. Erwin, J.A. Anderson, A.A. Fleisher : Constraints : Knowledge representation in design, Design Studies Vol 9 No 3 july 1988, Butterworth & Co publishers Ltd, 1988

[HEN 89] P.Van. Hentenrick : Constraint Satisfaction in Logic Programming, The MIT Press, Cambridge, MA, 1989

[HOW 86] A.E. Howe, P.R. Cohen, J.R. Dixon, M.K. simmons : dominic : A Domain independant Program for Mechanical Engineering design, Artificial Intelligence Vol 1 p 23-28, 1986

[KIM 86] F. Kimatu, H. Suzuki,: Variational product design by constraint propagation and satisfaction in Product modelling, Annals of the CIRP vol 35/1/1986 p75-78, 1986

[KIM 91] S. Kim, G.Schlageter : Competence Evaluation in Federative Problem solving : How to make expert system Cooperative, Outils de seconde génération, Avignon 91, p 95-108, 1991

[KLE 89] M.Klein : conflict Resolution in Cooperative design, Artificial Intelligence in engineering, Vol 4 #4, p 168-180, 1989

[KRA 87] G.A. Krammer : Incorporation Mathematical Knowledge into design models, expert systems in computer-aided design p 229-247, ed J.S. gero, North Holland, 1987

[LAA 91] B.Laasri, H.Laasri, R.Lesser : An analysis of Negotiation and its role for coordinating cooperative distributed problem solver, Outils de seconde génération Avignon 91 p 81-94, 1991

[LEL 88] Wm. Leler : Constraint programming Languages, Addison Wesley publishing Company, 1988

[LFF 91] C.Lffenecker, J.Ferber, D.Zawadzki : Un modèle multi-agents pour l'aide à la conception de produits Electromécanique, Outils de seconde génération, Avignon 91, p 137-151, 1991

[MIT 91] A. Mittler, P.Martineau : Intégration des contraintes de production au niveau des bureaux d'études, proceeding of MICAD'91

[MUR 87] S.S. Murthy, S. Addanki : Prompt : An innovative design Tool, expert systems in computer-aided design, p 323-341, ed J.S. Gero, elsevier Science publishers B.V, 1987

[OHS 89] S. Ohsuga : Toward intelligent CAD systems, Computer Aided design Vol 21 N° 5p 315-337, 1989

[RAD 85] A.D. Radford, J.S. Gero, M.A. Rosenmann, M.Balachandran : Pareto optimization as a computer Aided design Tool, optimisation in Computer Aided design p 47-77 ed J.S. gero, Elsevier Science Publishers BV 1985

[SER 87] D. Serrano, D.Gossard : Constraint managment in Conceptual design, Knowledge Base systems in Engineering : Planning and Design, eds D.Siram, R.A. Adey, Comptational Mechanics Publications, 1987

[SMI 88] R.G. Smith, R.Davis : framework for cooperation in distributed problem solvong. reading in distributed Artificial Intelligence p 61-70

[TEN 88] R.R. Tenney, N.R. Sandells : strategies for Distributed decision Making, reading in distributed Artificial Intelligence p 236-247