Engineering Applications of Artificial Intelligence 22 (2009) 961–970

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A CSP approach for the network of product lifecycle constraints consistency in a collaborative design context

Pierre-Alain Yvars *

LISMMA, Institut Supérieur de Mécanique de Paris, 3 rue Fernand Hainaut, 93407 Saint Ouen Cedex, France

ARTICLE INFO

Article history:
Received 15 May 2008
Received in revised form
6 August 2008
Accepted 27 November 2008
Available online 29 January 2009

Keywords:
Collaborative product design
CSP
Product lifecycle
Constraints
Conflict detections

ABSTRACT

In this paper, we are considering that the design process can be modelled in the form of a constraint satisfaction problem (CSP). CSP modelling or resolution has proved its efficiency within the framework of single-designer design. We propose to extend the functions of CSP to the context of multi-concept design of the same artefact. We define CoCSP as cooperative constraint satisfaction problem including the actors of the design problem. We are presenting the operating principles of an algorithm for the real-time management of design decisions, based on a model described in the form of a CoCSP for the integration of supply-chain constraints. This algorithm enables the number of design decisions rejected at a given moment in design to be kept to a minimum. The algorithm forms the core of a prototype for an unsupervised, generic constraint-based collaborative design system. Our aim is to produce a platform centred on the notion of constraints that will enable a product design problem to be modelled and solved by integrating supply-chain constraints as far upstream as possible.

1. Introduction

The partitioning of the various professional sectors in companies does not foster the taking into consideration of a wide range of research and development factors. This necessary cross-sector approach, adopted as early as the design phase, has started to arouse keen interest, notably through simultaneous engineering.

Life-cycle cost integrates all the costs incurred by the design, production, distribution, use and withdrawal from service, of a product. According to the cost breakdown of life-cycle, it has been observed that 80% to 90% of this cost is fixed during the design phase proper. Any subsequent changes brought to the product to save costs turn out to be either prohibitive or inefficient, if not both. This is all the more noticeable for logistics costs.

As a matter of fact, the way a product is thought out will have a direct bearing on its shape (weight, volume and bulk), its production method and, if fact, on its physical and technological constraints. These features will ultimately determine handling and storage resources, for both the finished product and its components. Moreover, the design will also have a direct bearing on the means of transport required. One can even take this reasoning a step further to consider the organisational and strategic aspects: one can assume, thus, that this upstream industrial phase will have a limiting effect, because of materials incorporated in the product, on those companies likely to produce the various semi-finished components, and will, therefore, determine the transport cost included in manufacture.

Nonetheless, in practice, marketing and commercial criteria are uppermost in a number of cases and prevail in the design phase naturally, once the service expected of the product and its production cost have been considered.

It is, therefore, becoming increasingly necessary to take logistic constraints and criteria sufficiently into account at the design stage.

The scope of this paper will lie on how to integrate the criteria and constraints of a product life cycle, (design, manufacturing and logistics), in a simultaneous engineering environment.

Over the past few years, we have turned to using constraint programming to solve design problems. This has included the development of meta models suitable for product modelling and design reasoning with a view to constraint-based resolution. (Sellini and Yvars, 1999) This work essentially covered mono-designer resolution, and we subsequently sought to extend the knowledge acquired to include the resolution of multi-designer problems.

CPSs have enabled us to integrate, into the frame work of the same model, those constraints and criteria specific to the product's life-cycle, viz. definition of product, manufacture, cost of manufacture and transportation.

In this paper, we will tackle the contribution brought by modelling and CPS resolution techniques that integrate multi-trade constraints for the collaborative instantiation of a product during the design stage.

After setting forth the core question raised by our paper, we will give an up to date review of the work that has been done up to

* Tel.: +33 0149452925; fax: +33 0149452929.
E-mail address: payvars@supmeca.fr

0952-1976/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2008.11.006

now using CPS analysis, as regards modelling, problem solving involving systems or product engineering, as well as allowing for multiple constraints, (e.g. defining product, manufacture, costs, etc.). This general outline will be supplemented with a synthesis of the current papers dealing with collaborative design. We will subsequently lay down a modelling framework known as CoCSP or cooperative CSP as being a multi-actor design problem model for systems or constraints under heterogeneous constraints. Afterwards, we will detail our own proposal of a CoCSP management algorithm. Eventually, we will present you with the prototype of a collaborative design system based on CoCSPs.

2. The core question

We are seeking to develop algorithmic models and I.T. tools liable to come to terms with the following issue.

A design team (in practice a dozen or so participants at the most) collaboratively works on the instantiation of the same artefact, product or system (Fig. 2).

One assumes that this design team works on the model of the product defined with the help of a CSP, of which we will highlight the contribution to product design, in the lines below.

One solution to this model is an instantiation of the product variables meeting the set of constraints integrated into the CSP. As a constraint is defined as any type of mathematical relation between variables, we will show that it is, therefore, possible to integrate the production definition, manufacturing and cost constraints into the same CSP model.

As far as the latter are concerned, as soon as one is able to establish a set of relations between the product variables and the various costs involved i.e. production and transport, one only needs to represent these relations in the CSP to start integrating as many product lifecycle constraints as possible, as early as the preliminary design phase.

As each designer demonstrates a particular range of competence giving him authority over the instantiation of a given subset of the CSP variables, each of them is liable to make a series of decisions aiming to instantiate one or several product variables. Each design decision of the variable = value type can be regarded as a further constraint, dynamically added to the initial CSP. At some stage, the actual CSP may lack consistency and offer no solution. It will thus be necessary to supply an algorithmic means to judiciously ease some decisions made by some designers so as to help the CSP to become consistent again. Ideally, all the decisions eased should be of minimal cardinality.

Moreover, not every designer welds the same clout over decisions, in a design department. Disregarding a constraint laid down by a project manager will not have the same impact as discarding an engineer's decision.

The system must be able to capture designer's decisions on product variables in a monitoring window, to propagate them and manage the incompatibility between decisions so as to converge towards a fully instantiated artefact, minimising the number of decisions called into question. The process complied with has been represented in the simplified recursive algorithm below, in which:

- The variable *DecisionList* represents the list of those decisions considered by the designers for a cycle.
- The variable *CSP* represents the current CSP.
- The variable *MaxListAcceptedDecisions* represents the list of accepted decisions.
- The function CONSISTENT returns a boolean, which indicates the consistency or unconsistency of the considering CSP.
- The function CYCLE starts the main catching designer's decisions (CATCHDECISIONS function), the consistency analysis (function CONSISTENT) and the maximum cardinality set of consistent decisions.

CYCLE(*DecisionList*)
begin
If (*DecisionList*)
Then
  If CONSISTENT ($CSP \bigcup DecisionList$)
  Then
  $CSP \leftarrow CSP \bigcup DecisionList$
  Else
  ListeMaxiDecisionsAcceptees ← CoCSP( )
  $CSP \leftarrow CSP \bigcup MaxListAcceptedDecisions$
  End If
  $l \leftarrow$ CATCHDECISION( )
  Return CYCLE(l)
End If
Return 1
End

It is not within the scope of this study to rewrite the function CONSISTENT, which allows the consistency of a set of constraints or even to guarantee the existence of a solution. These algorithms are now fully developed and understood in the industrial sector and we will use a library available on the market that provides them.

The crux of the matter is to judiciously generate the content of the *MaxListAcceptedDecisions* variable as the decision acquisition cycle effectuated by the designers takes place.

The final objective of this paper is to supply a design team with the tools enabling them to design an industrial product, while integrating, at the earliest, the constraints upstream of the product lifecycle.

In the following paragraph, we take a state-of-the-art look at the use of CSPs in design, as well as assessing current work in the field of collaborative design.

3. CSP and collaborative product design: a state-of-the-art overview

3.1. CSPs

A CSP (Tsang, 1993) is defined by a triplet *(X, D, C)* such that:

- $X = \{x_1, x_2, x_3, \ldots, x_n\}$ is a finite set of variables, which we call constraint variables with n being the integer number of variables in the problem to be solved.
- $D = \{d_1, d_2, d_3, \ldots, d_n\}$ is a finite set of variable value domains of X such that

  $$\forall i \in \{1, \ldots, n\}, \quad x_i \in d_i.$$

- $C = \{c_1, c_2, c_3, \ldots, c_p\}$ is a finite set of constraints, p being any integer number representing the number of constraints of the problem

  $$\forall i \in \{1, \ldots, p\}, \quad \exists X_i \subseteq X / c_i(X_i).$$

Solving a CSP boils down to instantiating each of the variables of X while meeting the set of problem constraints C, and at the same time satisfying the set of problem constraints C.

Here, a constraint is any type of mathematical relation (linear, quadratic, non-linear and Boolean…) covering the values of a set of variables.

Our CoCSP model solving is based on existing CSP algorithms. Then, we think useful to present a survey about those solving methods.

One finds several types of CSP in the relevant literature.