

Sílvio do Carmo-Silva^a, A.C. Alves^a, P. Novais^b, M. Costa^c, C. Carvalho^b,
J. Costa^b, M. Marques^b

^aCentre for Production Systems Engineering (CESP), University of Minho, Campus de
Gualtar, 4700-057 Braga, PORTUGAL (scarmo@dps.uminho.pt; anabela@dps.uminho.pt)

^bDepartamento de Informática-CCTC, University of Minho, Campus de Gualtar, 4700-057
Braga, PORTUGAL

(a35288@alunos.uminho.pt; jorgemaiocosta@hotmail.com; marquesmps@gmail.com;
pjon@di.uminho.pt)

^c(costamiguel@bragatel.pt)

Manufacturing leanness and agility are requirements of today's manufacturing systems. Leanness call for a best fit of the manufacturing systems to products, therefore requiring product oriented manufacturing systems (POMS). Manufacturing agility can be achieved through easy systems reconfiguration to fit changing manufacturing requirements, which may mean dynamically configuring POMS. For this a suitable design system is required. Due to complexity of this design, and to the need for using suitable design methods, which may not be available locally, distributed sources of design services can be used. This paper presents and describes a prototype of a Distributed Design system for POMS based on a POMS design methodology and distributed suppliers of design services.

1 INTRODUCTION

Industrial companies, nowadays, live in a paradigm of high competition in a market environment of frequently changing product demands. Therefore, manufacturing leanness and agility are requirements of today's manufacturing systems as a means of effectively answer varying market requirements and maintain competitiveness. Leanness call for a best fit of the manufacturing systems to manufacturing requirements of products. Manufacturing agility can be dealt with by frequently adjusting or reconfiguring manufacturing systems to fit changing manufacturing requirements. Thus apparently, both leanness and agility can be achieved through Product Oriented Manufacturing Systems (POMS). These may be defined as systems interconnecting manufacturing workstations or cells, usually involving people, which simultaneously and in a coordinated manner address the manufacture of a single product or a family of similar products, subject to frequent

reconfiguration to be adapted to changing manufacturing requirements of products or product families (Carmo-Silva et al., 2005).

The design of POMS tends to rely mostly on human expertise and ability for arriving to acceptable system configuration solutions to fit continuously changing product demand. This design approach is slow and ineffective and tends to introduce inefficiency on manufacturing operations activity. Such human based design may be justified when no suitable design system for POMS design, is available. Since, in addition to several methodologies (Suh, 1990, Burbidge, 1989, Black, 1991, Cochran et al., 2002), there are available a large variety of methods and procedures to aid the manufacturing system design function (Suresh and Kay, 1998, Irani, 1999) apparently there is no reason to base POMS design almost exclusively on human expertise. The use of design systems based on computer applications, which implement valid methods under a suitable design methodology, can provide the missing tool required for achieving fast and good reconfiguration solutions of POMS, to fit changing manufacturing requirements. However, not always the methods required are locally available. Due to this, many design functions may have to rely on distributed sources of design services. This paper presents and describes a prototype of a distributed design system for POMS based on a manufacturing system design methodology and distributed suppliers of design services.

The paper presents in section 2 a description of the computer aided design system framework organized around a POMS design methodology. In section 2 the architecture components and prototype of a POMS design system based on a community of servers providing design services is described. In section 4 a conclusion is presented.

2 COMPUTER AIDED DESIGN SYSTEM FRAMEWORK FOR POMS DESIGN

2.1 CADS_POMS framework

POMS design can be based on a design methodology. One such methodology, named GCD - Generic-Conceptual-Detailed - was proposed by Silva and Alves (2002). It essentially puts forward a hierarchical multilevel and iterative design process for POMS. Important data and restrictions are considered and a range of methods can be used in the POMS design process. Under this methodology the design process is organized in three main phases and includes several design stages and activities.

However important the GCD methodology may be, it can be of little use if not supported by a computer aided design system (CADS). This must address POMS design activities from strategic planning to the POMS organization and production control mechanisms definition. A CADS_POMS framework based on the GCD methodology for POMS design is shown in Figure 1.

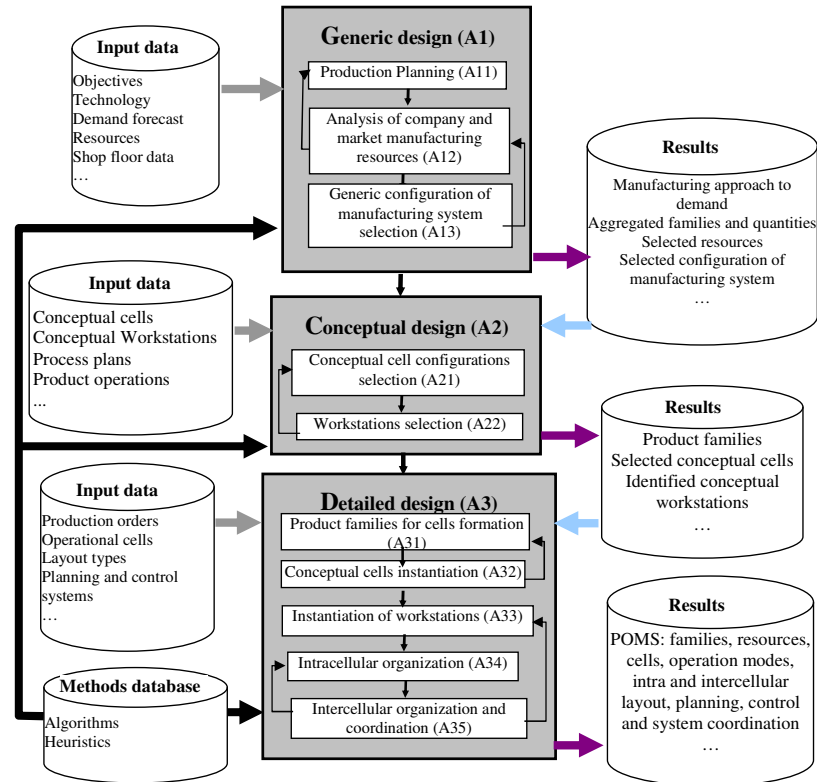


Figure 1. A CADs_POMS design framework based on the GCD methodology

3 A DISTRIBUTED DESIGN SYSTEM

3.1 System Architecture and functionalities

Figure 2 illustrates the main components and gives a simplified view of the architecture of the POMS distributed design system that may be seen as an instantiation structure of the CADs_POMS design framework presented in figure 1.

The system is organized around a central POMS database, a POMS design methods database, a distributed set of design methods servers, for aiding POMS design, and a interactive human-computer POMS design application module, with graphical user interfaces, named POMS designer.

Although some design methods and procedures for POMS design may be available locally, many more can be available through several servers distributed globally. These are identified as POMS design methods servers. These servers supply POMS design services requested by the POMS designer to perform design functions or tasks under the CADs-POMS design framework show in figure 1.

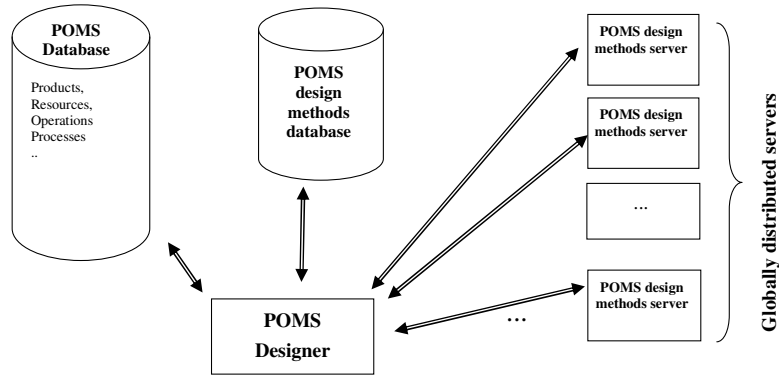


Figure 2 - POMS design system components and architecture

The POMS designer interacts with all components of the POMS design system. It interacts with the POMS system Database in many ways. Initially products, manufacturing resources and a set of generic operations, which can be instantiated by process planning for defining the process plan for each product to be manufactured, are specified or copied into the database. Then, the user can plan the process for manufacturing each product by using the set of standard generic operations which are then parameterized according product processing requirements (Carmo-Silva et al., 2005). Resources, mostly machines that are central to workstations, capable of carrying out product operations, are then chosen according to resource attributes and processing capabilities. Estimates of operation processing times for each resource are also produced. This process is totally under control of user, although some process planning systems may be used to ease this task. These described functions are essential for generating the main data and information required for POMS design. A computer application addressing this process of data generation for POMS design has been already developed (Carmo-Silva et al., 2005). This can be seen as an important part of the prototype of the POMS design system here reported.

Since it is sought frequent system configuration for adapting production systems to changes in product demand, we may refer to this POMS design as POMS reconfiguration.

3.2 Distributed Design Services

POMS reconfiguration relies not only on methods locally available but also on design methods implemented and run at distributed servers. The interaction of POMS designer with the methods database is the first step in the POMS reconfiguration process. This database provides information about the location of servers and the list of methods that are available, and where, for solving specific design tasks or problems, at request of the POMS designer. The methods database also provides information about the data input formats which are used for each method.

The following assumptions were considered in the modeling of the methods database: each method solves a specific problem or design task; each problem relates to a particular design activity in a design phase of the GCD methodology summary illustrated in figure 1; one particular design method may have implementations in more than one server.

The input data for running a method is prepared by the POMS designer and sent to a suitable methods server, which then runs the method and sends back results to the POMS designer in an already known format. The communication process is carried out over the network through the TCP/IP protocols.

3.3 Running Methods on Servers

In an implementation example of the distributed POMS reconfiguration process a methods server application was developed. This, once operating in a methods server that is active, allows the server to accept requests from the POMS designer for design tasks. Figure 3 identifies the graphical interface of the methods server application.

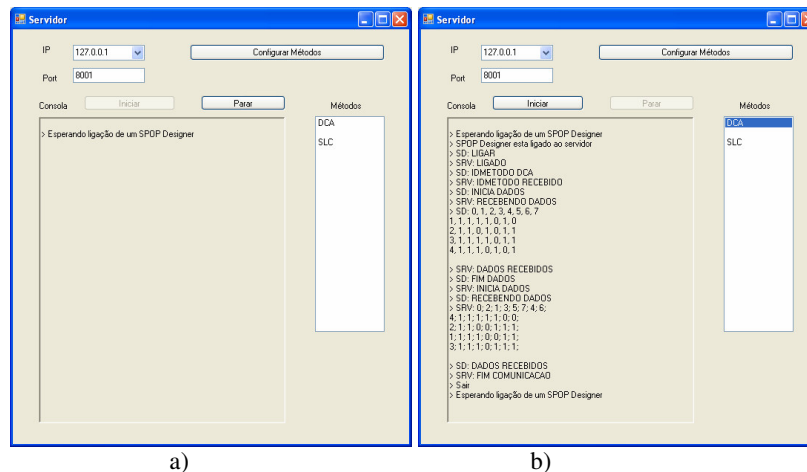


Figure 3 - Methods server application interface: a) waiting for the a request from a client; b) input data from the request and output data sent to the client

It shows the server capability to offer two methods for POMS design tasks, namely the DCA – Direct Clustering Algorithm (Chan and Milner, 1982) and the SLC - Single Linkage Clustering (McAuley, 1972) methods. These are useful for forming manufacturing cells by identifying product families, based on process plan similarities of products. This is part of activity A31 of the detailed design phase A3 (Carmo-Silva and Alves, 2006), as shown in figure 1. Input data sent by the SPOP designer, prepared with basis on a SQL stored procedure for the DCA implementation on the server, is illustrated in the figure 3.

Once there is a need for a design task to be carried out the SPOP designer must choose a suitable method from a methods list provided in the methods database and, additionally, select the server to supply the service.

A request is then generated and stored in a list of requests, figure 4. This list of design service requests by the SPOP designer is generated as a matter of monitoring the answers of servers, and allowing request repetition when for some reason, related for example with communications or server operation, no answer is received from servers.

id	Método	sServidor	Estado	Input	Output	DataCriacao	DataFinal	Tentativas	Log
2	DCA	LOCALHOST	<input type="checkbox"/>	0, 1, 281, 1, 182, 1, 18		30-12-2006 2:03		12	ⓘ Tentando ligação com o servidor 192.168.1.100 na...
3	DCA	LOCALHOST	<input type="checkbox"/>	0, 1, 2, 381, 1, 0, 182...		30-12-2006 2:09		0	
4	DCA	LOCALHOST	<input type="checkbox"/>	0, 1, 281, 1, 182, 1, 18		30-12-2006 2:15		0	
5	SLC	ANIOZ	<input checked="" type="checkbox"/>	1 0 30 1	DADOS DE RESPON...	30-12-2006 2:20		1	ⓘ - FIM - Ⓣ Tentando ligação com o servidor 192.168...
6	SLC	ANIOZ	<input checked="" type="checkbox"/>	1 1 81 1	DADOS DE RESPON...	30-12-2006 2:24		1	ⓘ Tentando ligação com o servidor 192.168.1.100 na...
7	TESTE	ANIOZ	<input type="checkbox"/>			30-12-2006 2:26		0	
8	Teste	CamoSrv	<input type="checkbox"/>			30-12-2006 2:26		0	
9	TESTE	ANIOZ	<input type="checkbox"/>			30-12-2006 2:28		0	
10	Teste	CamoSrv	<input type="checkbox"/>			30-12-2006 2:28		0	
11	DCA	ANIOZ	<input type="checkbox"/>	0, 1, 281, 1, 182, 1, 18		30-12-2006 2:30		0	
12	DCA	ANIOZ	<input type="checkbox"/>	0, 1, 281, 1, 182, 1, 18		30-12-2006 2:30		0	
13	SLC	ANIOZ	<input type="checkbox"/>	1 0 1 81 0 1 30 1 1		31-12-2006 15:15		0	
14	DCA	LOCALHOST	<input type="checkbox"/>	0, 1, 2, 3, 4, 5, 681, 1, ...		01-03-2007 14:06		0	
15	dca	andre	<input type="checkbox"/>	0, 1, 281, 1, 182, 1, 18		01-03-2007 14:09		0	
16	dca	andre	<input type="checkbox"/>	0, 1, 281, 1, 182, 1, 18		01-03-2007 14:09		2	ⓘ Tentando ligação com o servidor 127.0.0.1 na port...
17	dca	andre	<input type="checkbox"/>	0, 1, 2, 3, 481, 1, 0, 1, ...		01-07-2007 10:53		0	
18	DCA	jorge	<input type="checkbox"/>	0, 1, 2, 3, 481, 1, 0, 1, ...		01-07-2007 10:53		3	ⓘ Tentando ligação com o servidor 192.168.100.1 na...
19	DCA	LOCALHOST	<input type="checkbox"/>	0, 1, 2, 3, 481, 1, 0, 1, ...		01-07-2007 10:53		0	
20	dca	andre	<input type="checkbox"/>	0, 1, 281, 1, 082, 0, 18		01-07-2007 11:27		0	
21	DCA	jorge	<input type="checkbox"/>	0, 1, 281, 1, 082, 0, 18		01-07-2007 11:27		0	
22	DCA	LOCALHOST	<input type="checkbox"/>	0, 1, 281, 1, 082, 0, 18		01-07-2007 11:27		0	
23	SLC	ANIOZ	<input type="checkbox"/>			01-12-2007 10:22		0	
24	dca	andre	<input type="checkbox"/>			01-12-2007 10:23		0	

Figure 4 - Requests list

A summary of the request and service supply process is shown in figure 5. The need for a design task to be performed in a particular activity of design phase leads to the choice of a design method listed in the methods database and a selection of the suitable server. This permits a request for the design task after due preparation of the method input data in the correct format. The answer from the server to the SPOP designer request is provided through the methods server application for a request output, figure 5.

3.4 Managing Distributed Servers Community

The system architecture allows dynamic updating of the community of servers providing design services. Any server that joins the community must be registered. This means supplying its location and specifying the design services that it offers. This is done through the SPOP designer that, in addition to ensuring the POMS design process has also the task of updating the method database. This means adding the new server to the methods' servers list, in the methods database, and specifying the design phase and problem that each new method solves. Moreover, the input data format required by each method, offered by the new server, is also specified. In the actual implementation this is done through SQL stored procedures. The removing of methods and servers from the methods data base can also be carried out.

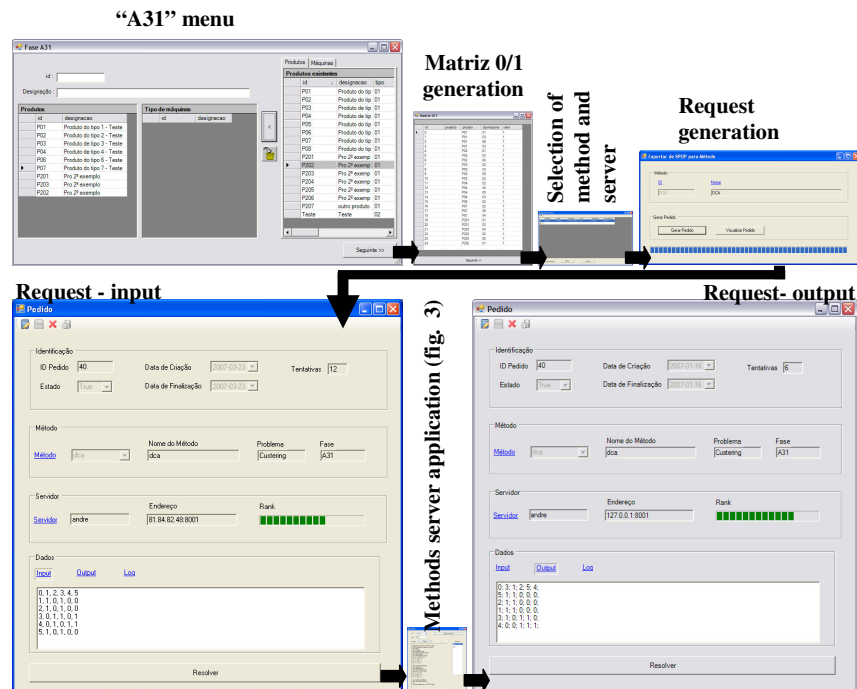


Figure 5 - Forms and menus for the design service request and design results provided

4 CONCLUSION

Frequent reconfiguration for adapting POMS to changing product manufacturing requirements is one way of achieving both manufacturing agility and leanness for gaining competitive advantage in the unpredictable product demand environment of today and tomorrow.

Although in simple manufacturing situations human expertise and ability may be enough to arrive to good POMS reconfiguration solutions, better results are likely to be obtained with a suitable computer aided design systems for POMS. This must be structured around a design framework or methodology which can explore the use of several methods, in different design phases, which have been developed over the years, by many authors, and that can of great value in the POMS reconfiguration process.

The idea of developing an autonomous monolithic POMS design system seems to be unfeasible and not recommended, for many reasons including the difficult of making locally available the necessary POMS design methods which, nevertheless can be accessed through the Internet. Moreover the dynamic up-dating of the system "intelligence" for POMS design, would be a difficult task. Better would be to provide the POMS design system with access to a large community of POMS design service providers. This idea has been explored to develop a computer aided design system, based on a CADs-POMS framework structured under a POMS design

methodology and a community of distributed servers providing POMS design services through the Internet. A simple prototype of such a system is described in this paper.

5 REFERENCES

1. Black JT. *The Design of the Factory with a Future*, McGraw-Hill, 1991.
2. Burbidge JL. *Production Flow Analysis for planning Group Technology*. Clarendon Press, Oxford, 1989.
3. Carmo-Silva S, Alves AC, Costa M. A Computer Aided Design System for Product Oriented Manufacturing Systems Reconfiguration. *Intelligent Production Machines and Systems (Proceedings of the 1st I*PROMS Virtual International Conference)*, D. T. Pham, E. E. Eldukhri, A. J. Soroka Eds., Amsterdam, Elsevier, 2005: 417-422.
4. Carmo-Silva S, Alves AC. Detailed design of product oriented manufacturing systems. *Proceedings of Group Technology / Cellular Manufacturing 3rd International conference – 2006*, J. Riezebos and Ir. J. Slomp Eds., University of Groningem, Holland. 2006: 44, 260-269.
5. Chan HM, Milner DA. Direct Clustering Algorithm for Group Formation in Cellular Manufacture. *Journal of Manufacturing Systems*. 1982: 1: 1.
6. Cochran DS., Arinez, JF., Duda, JW., Linck J. A decomposition approach for manufacturing system design. *Journal of Manufacturing Systems*, 2001/2002: 20: 6;: 371.
7. Irani, SA. *Handbook of Cellular Manufacturing Systems*. John Wiley & Sons, 1999.
8. McAuley, J. Machine grouping for efficient production. *Production Engineer* 1972: 51-57.
9. Silva SC, Alves AC. Design of Product Oriented Manufacturing Systems. In *Knowledge and Technology Integration in Production and Services*, V. Marik, L. Camarinha-Matos and H. Afsarmanesh, Eds. Kluwer Academic Publishers, 2002: 359-366.
10. Suh NP. *The principles of Design*. Oxford University Press, 1990.
11. Suresh NC, Kay JM. *Group Technology and Cellular manufacturing*. Eds., Kluwer Academic Publishers, 1998.