C CAPP IMPLEMENTATION TECHNIQUES

CAPP implementation techniques covered broadly current techniques; therefore, this section approaches the following topics matter:

- Artificial intelligence (AI),
- Group Technology,
- Software Programming and Internet, and
- Knowledge Acquisition and Representations

C.1 Artificial Intelligence (AI)

CAPP used AI as a major technique because process planning activities was believed highly knowledge intensive, complex, and dynamic in nature (Ming et al., 1999), and because process-planning was predominantly an open-ended problem that accepted multiple solutions based on the preferences, experience and domain knowledge of the planner (Maropoulos, 1995).

Therefore, the various AI techniques applied in CAPP included the knowledge-based systems (KBS) (Wong and Siu 1995, Pande and Desai 1995, Kryssanov at al. 1998), case-based reasoning (CBR) system (Ben-Arieh and Chopra 1997, Champati et al. 1996, Kim 1999), neural networks (NNs) (Lin et al. 1995, Marri et al. 1998, Dimla 1999, Chang and Ho 1999, Devireddy and Chosh, 1999, Joo et al. 2001, Ahmad et al. 2001), Petri nets (Wu et al. 2002), fussy logic (Dweiri and Meier 1996, Ip 1998, Wang 1999), Genetic Algorithms (GA) (Gupta et al. 1996, Dutta and Yip 1996, Rajasehharan 1997, Hamamoto et al. 1999, Morad 1999, Bhaskara et al. 1999, Meziane et al. 2000), or hybrid systems (Rozenfeld and Kerry 1999, Suresh et al. 1999, Lee et al. 1999, Ming et al. 1999, Chang and Chang 2000).

There was no doubt that the introduction of AI techniques has boosted both the interest in the problem and the capability of the CAPP systems, but its benefits were considered disputable. One considered that CAPP greatly benefited from artificial intelligence methods (Cay and Chassapis, 1997), and others considered that AI has had great success in manufacturing only where it has been deployed as "Intelligence Amplification" and not when it has tried to tackle problems with complete automation in the area of CAD/CAM systems (AAAI, 1997). Also, research carried out in the last twenty years showed that the classical techniques of developing expert systems did not created a complete CAPP expert system (Kryssanov et al., 1998), few expert systems had significant impact on manufacturing industries (Ming et al., 1999), Genetic Algorithms (GA) were far from having an impact in practice (Aytug et al., 2003), and it was unusual to see reports about successful general CAPP solutions based on AI (Rozenfeld and Kerry, 1999), therefore AI results were far from desirable (Jain et al., 1998).

With all these, it must be highlighted that little work was reported on the use of NNs and fuzzy logic in process planning (Meziane, 2000), and the fact that the knowledge-based expert systems, considered among the most practical and successful applications of AI techniques (Wong and Siu, 1995), provided well-adapted solutions in the process planning field (Kiritsis, 1995).

Furthermore, from an AI standpoint, manufacturing was seen as a rich, research driver, application area, and too often work has proceeded without good understanding of the actual manufacturing problems and so with only marginal practical gain. From the manufacturing side, there was general recognition that AI has had important contributions to make, but there was limited understanding of AI technologies and their relevance (AAAI, 1997).

C.2 Group Technology

Group technology, has been used in CAPP systems as a code system to classify the geometric characteristics of the part and its raw material information (Sheu, 1998), but the disadvantages of the CAPP systems based on group technology was believed to consist in the abundance of the modifications that needed to be applied to the new process plans (Kuric et al., 1999).

C.3 Software Programming

Although software programming, and in particular the object-oriented programming, offered a new way of thinking about solving problems (Luo et al., 1997), and a promising alternative to other techniques (Lo and Lin, 1999), it received little attention in CAPP (Liang and O'Grady, 1998),

Therefore, a new CAPP software development should require not only a deep knowledge of manufacturing processes, the concerns, and difficulties that could arise for each problem (AAAI, 1997), but also consider software surrounding characteristics, relationships and context, and focus on the problem before the solution, and on what the system will do before how it will do it (Jackson, 2001).

In addition, a large and complex problem should be transformed into a number of smaller and simpler sub-problems, which could solve not only the problem, but could also help to capture, describe, understand, and analyze the problem better (Jackson, 2001). Consequently, manageably-sized flexible and generic software modules could be specified, coded, tested, and debugged separately (Watt and Brown 2001, Molina and Bell 2002).

Also, as new designs and knowledge have been developed (Luo et al., 1997), one should consider the software agents as a solution for business integration (Aldakhilallah and Ramesh 1999, Morad and Zalzala 1999, Meziane et al. 2000, Ünver and Anlagan, 2002, Lee et al. 2003), and Web technology as a solution for expanding the cooperation among enterprises (Huang and Mak 1999, Cheng et al. 2001, Peng 2002, Koutsogiannakis and Chang 2002, Yang and Xue, 2003).

In addition, similar to many problems in mathematics and computer science considered unsolvable by algorithms (Watt and Brown 2001, Alan Turing 1912-1954), algorithms also were not appropriate for process planning because a very large search space for many alternative solutions will result (Luo et al., 1997).

Therefore, to mitigate the software development risks (Jacobson et al. 1999, van Vliet 2000), one should: analyse the ways in which the companies are run (Wang et al., 2002) and their important social, organizational, and managerial factors (Kazman and Bass, 2002); get requirements right; consider the relationship between the problem and the computation techniques (Reed, 2002); avoid the use of a single tool to solve all problems (Pavnaskar et al., 2003); develop tools to support formal knowledge (Zwikael and Globerson, 2004); and align the architecture for problem solving (AAAI, 1997) of the new software system with the organization's business goals (Sharma et al. 2001, Kazman and Bass 2002), cultural issues (AAAI, 1997), and people's need of information as manufacturing tasks progress (HMS, 2005).

C.4 Knowledge Acquisition and Representation

The process of acquiring knowledge and translating it into computer format has been known as knowledge engineering (Kiritsis, 1995). Knowledge representation was defined as a combination of data incorporating facts, events, and their relations, also known as declarative knowledge, and the rules on how to use them (Grabowik and Knosala, 2003), also known as procedural knowledge.

Knowledge representation was considered a major problem of automated process planning systems (Kiritsis, 1995), because none of the various methods used alone could be sufficient to grasp all the multifarious aspects of the large and complex domain like the machining domain (Ahmad et al., 2001) (Table C.1). Consequently, the process planner required a great experience (Ciurana et al., 2003) and an understanding of all other knowledge sources together (Kryssanov et al., 1998).

Therefore, when developing new application, one should: observe the natural relationship between understanding, knowledge, and experience (Zwikael and Globerson, 2004); consider how to match the available knowledge, enabling technology, and existing resources, with the application requirements (Zwikael and Globerson, 2004); adopt a practical approach (Luo et al., 1997) based on recognition that workers are a source of creativity (Turner, 1998); and remember that the tolerances on the working piece have been almost always dependent on the detailed knowledge of the machine-tool operator (Whybrew and Britton, 1997).

Table C.1 Examples of CAPP knowledge representation

Authors	Knowledge representation techniques
Gary (Hugh, 1994 and Kiritis, 1995)	Production rules that use “if-then” and preferences weighted according to their importance.
Khoshnevis et al., 1999	A hierarchical tree of frames for features, machining processes, machines, and tools, and their relationships.
Ming et al., 1999	Facts and basic rules for explicit knowledge and neural network weights for implicit knowledge.
SIPP (the University of Maryland)	Hierarchical knowledge divided in static knowledge, stored by objects, and problem-solving knowledge.
Wu et al., 2002	Petri nets for operation planning knowledge and algebraic equations for operation selection.