Published in the Proceedings of the 5th. Int'l Conference on Information Systems Analysis and Synthesis, 1999

A MULTI-AGENT SPARE PART GROUPING SYSTEM FOR LOGISTICS OPTIMIZATION

Thomas E. Potok and Nenad Ivezic
Collaborative Technologies Research Center
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, 37831, USA

ABSTRACT

The readiness of any modern military is closely related to the availability of spare parts for equipment needing repair. One approach to solving this problem is through advanced manufacturing processes such as just-in-time or lean manufacturing. Moving to this type of environment typically involves grouping parts based on how they are made. This is a very intensive manual process that we have applied multi-agent technology to address. We have demonstrated that part groups can be effectively constructed based on a wide variety of characteristics of the parts. With an agent representing each part in the system, a minimum spanning tree algorithm is used for part agents to group themselves, be added to a group, or be split from a group. We have further shown that the results from our agent system can provide a different perspective to humans about a seemingly well understood problem.

1. INTRODUCTION

A key challenge to military readiness is the capability to maintain and quickly repair damaged aircraft. This challenge requires the manufacturers and distributors of spare parts to be able to supply the military with the right part, in a short amount of time, for a reasonable price. If this goal is not met, then potentially vital equipment can be idled for long periods of time until the needed part is available. Historically, this challenge has been addressed by developing and maintaining a very large spare parts inventory. The most notable problem of large spare parts inventory is the increasing setup and maintenance cost.

A solution to these type of problems is for part manufacturers to deliver a needed part in a matter of hours. This approach completely bypasses the need for an inventory of spare parts and thus greatly reduces inventory costs.

One key step for just-in-time manufacturing of parts is to group parts based on similarities in the manufacturing process of these parts [Womack and Jones (1996)]. From this grouping, very productive manufacturing cells can be developed. A family of parts can be tailored to the capability of a specific supplier, or group of suppliers. For example, a bracket may require a piece of metal to be cut to a specific size, then to have three holes drilled in the metal, and finally to have the metal bent to a specific angle. Chances are there are a number of other parts that go through a very similar process. A supplier that can build one of these parts could easily build the other. If these parts are grouped based on how they are made, then this group of parts can most likely be made faster and cheaper than if a collection of different manufacturers are used to build the same parts.

This concept is not new, and is often referred to as Group Technology (GT) [2]. GT, however, is suited for developing cells for parts to be produced on a single manufacturing floor. Most of the methods of addressing this problem are rooted in mathematical approaches, there are examples of the use of AI techniques to solve this problem [3,4]. However, the inventory problem described above extends well beyond a single manufacturing floor, and requires innovative approaches to address it.

Many manufacturers are required to work in conjunction to produce spare parts in a just-in-time environment. Agent technology provides a flexible solution to grouping parts from a broad set of suppliers rather than from a single manufacturing floor.

2. APPROACH

We have developed a multi-agent system that is capable of grouping parts that will be produced by a general set of part manufacturers, rather than in a single manufacturing environment. We have defined agents that represent the characteristics of each spare part. These agents interact in three distinct ways. First the agents cluster themselves based on part attributes, i.e., similar materials or demand rates. This provides the ability to filter the part information based on attributes from a collection of parts.

Second the agents to cluster themselves based on their represented manufacturing process, i.e., the processes that are used to produce the parts. An example result of this clustering is shown in Figure 1. This figure shows a Phyllips tree representation of the five-part families generated by the agent for this data. This tree shows five major branches at the center of the tree, and far more groups as you move towards the leaves.

This clustering can be done strictly based on the manufacturing process or on a broader set of characteristics, such as part cost, and processing sequence. If the manufacturing process is not available, which is typically the case, then parts are grouped into families based on patterns of the part attributes. In other words, if an 8-inch aluminum extrusion fits into the "extrusion" family, then there is a good possibility that a 7-inch aluminum extrusion will also fit into the "extrusion" family. The part agents cluster themselves based

---

This material is based upon work supported by the Defense Logistics Agency under Contract No. SP0410-98-D-0005. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Logistics Agency.

on minimizing the dissimilarity of the processes used in making the parts. This dissimilarity can be thought of as two part agents comparing their manufacturing process information and generating a factor representing how far apart this information is. From this factor the part agents with small dissimilarities tend to form clusters. These clusters can then be merged or separated based on the number of part families, or level of dissimilarity that a user desires. The lower the level of dissimilarity, the large the number of clusters, or families produced.

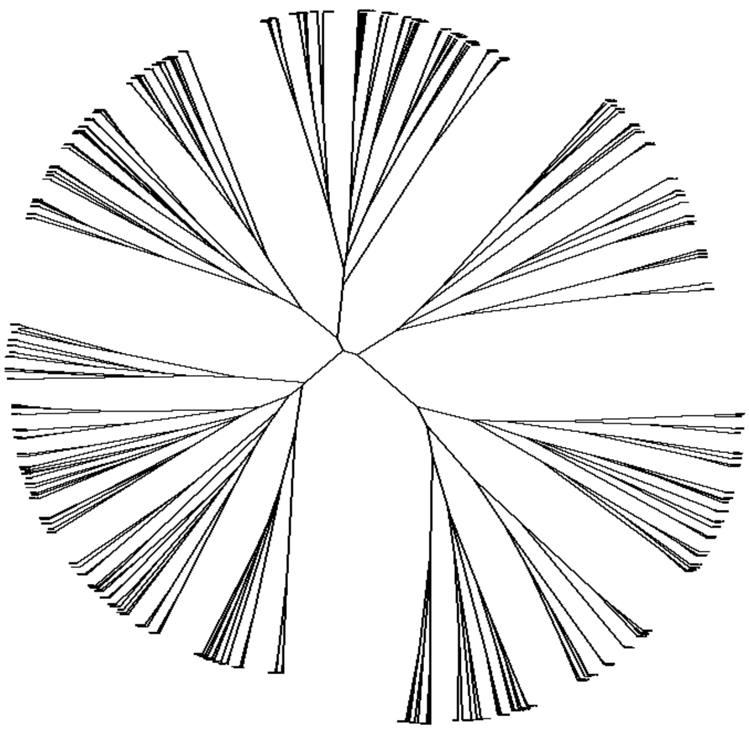

Figure 1 A Phyllips tree representation of the 276 C130 parts. There are five basic groups of roughly equal size. The length of the branches shows how close one group of parts is to another.

The third and final type of interaction is for the agents to cluster themselves based on a specific vendor capability. This provides for vendor specific input into the grouping process. Currently this is a manual interaction with the agent, however, we envision agents at a wide variety of manufacturers negotiating part placements within groups.

3. EXPERIMENT

We have conducted several experiments that exercise the functionality of this multi-agent system. Typically, a new grouping algorithm is developed by testing how the algorithm performs on a known set of part data. This data may have been previously published in the literature, or the data is generated to illustrate the advantages of the new algorithm. Our case is quite different. We have access to a volume of production part information, and access to several key manufacturing experts. Our first experiment was to group a large set of parts that were all very similar in function, and in the way they were manufactured. The main problem we faced was in showing the part grouping to the manufacturing experts. The techniques used in the grouping literature are very illustrative, however, for only a small set of parts and manufacturing process steps. For a large number of parts and process steps, the number of pixels on a reasonable computer screen becomes a critical limitation. We opted for a summary presentation of the data that gave a reasonable presentation of the groups, but was limited on details, (Figure 1).

We next moved to a smaller set of very different parts from the C130, described in detail below. For this case we changed our algorithm from a very simple coefficient-based grouping to a minimum spanning tree-based clustering algorithm. In the simple approach, the agent moderator performs most of the work by conducting a pairwise comparison of each part agent. The minimum spanning tree algorithm relies on the part agents to compare themselves, and then decide whether parts should form a group, be added to a group, or whether a group should be split. For this experiment, we defined a knowledgebase of part information, a formal part agent structure, and a general purpose moderator. This general architecture has remained constant for the remainder of the project, while several functional enhancements have been made. The grouping results are summarized in the following section.

Our next experiment was to group a very large set of general parts from two different aircraft. With this grouping we added the ability for the user to be able to weight the importance of process steps that contribute to the groupings. For example, the user may want to group parts based on a fixed process, such as a furnace, or a large immovable machine, or to avoid grouping based on a simple, or clerical process, such as an inspection, or the updating of production planning information. With this enhancement we did see differences in the part groups that were developed.

Example Results

We have used our grouping agent to analyze a wide variety of parts from two currently active aircraft, the F16, and C130. We were able to obtain data on parts that included the processing steps involved in building these parts. From this data the agent was able to group various sets of parts into part families where all of the parts within a family were made in a similar way. Below we describe one of our experiments, based on 276 active C130 Parts. These part types include extrusions, sheet metal parts, plate metal parts, welded assemblies, and bonded assemblies. For these parts, there are a total of approximately 60 unique process steps. For example, there are process steps for drilling, bending, cutting, milling, and many others. One part may require 12 of these steps, while another part may require 25. From the process steps associated with each part, the agents representing these parts form clusters based on the similarity of the processes needed to build the parts.

Figure 2 shows a more detailed breakdown of how the 276 parts actually grouped. We start by looking at the five major groups shown in Figure 1, labeling them 1 though 5. Each of these five groups is made up of parts from the original part types, e.g., extrusions, or sheet metal parts. Figure 2 shows a comparison of the original part types to the groups determined by our agent. The results are quite interesting. At first glance, it might appear as if the agent misclassified several part types, namely, plate metal parts, sheet metal parts, and extrusions, since these types span two or more groups. However, further analysis from human parts experts show that the agent groupings are quite good, and has opened new lines of thought from the experts on what types of parts can be produced within the same manufacturing cell.

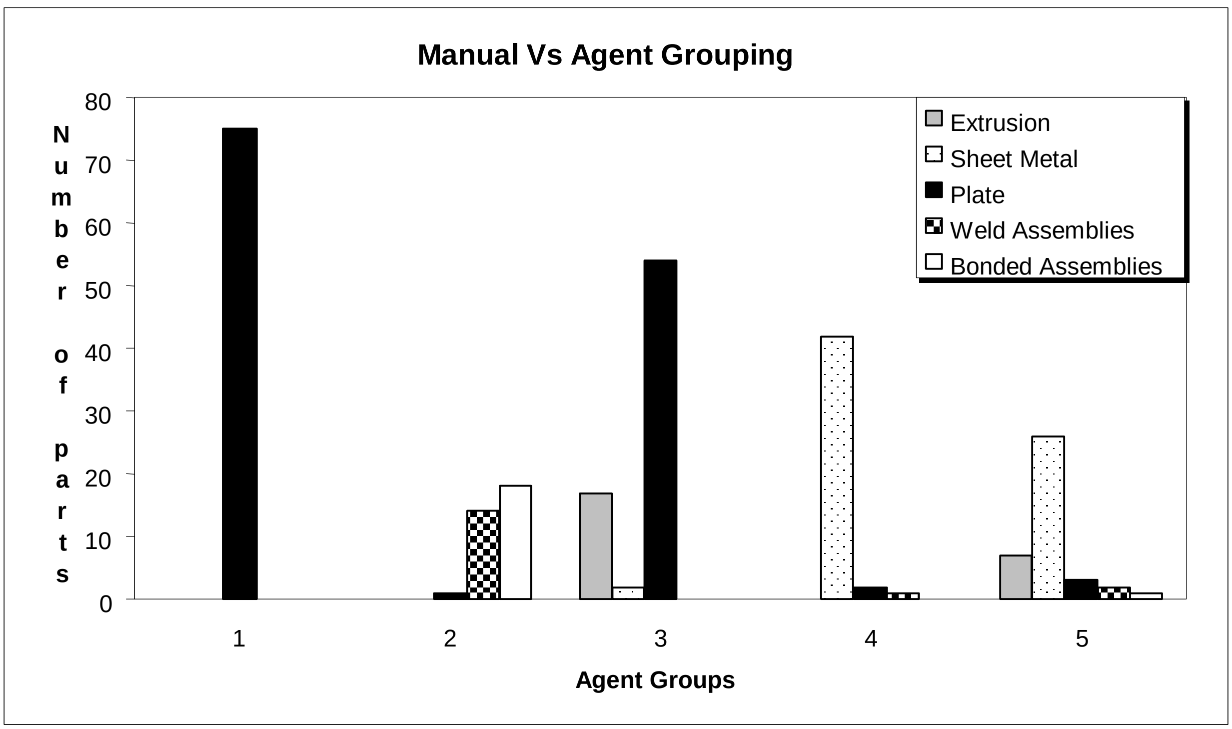

Figure 2 A comparison of the manual part grouping to the agent derived part grouping.

For example, traditionally, manufacturers do not put sheet metal, metal plate, or extrusions within the same manufacturing process due to the different types of material being used. However, many of the actions performed on this material are virtually identical across different cells. It offers a great opportunity to be able to consolidate different types of material within one general purpose cell.

4. DISCUSSION

We are very pleased with the results of the multi-agent grouping system we have developed. We have been able to group a wide variety of parts, often providing new insight into what type of parts can potentially be manufactured within the same cell. The results are very encouraging to our sponsors, and we have made very good use of multi-agent technology to solve a very challenging aspect of the military logistics project.

Currently this system does not support learning. This is mainly due to the lack of a viable mechanism to determine the proper grouping of parts. Manual results are not feasible, except for very small part groups. There are numerous metrics available for categorizing the suitability of a part group. But, these metrics assume that you are grouping the parts for a specific manufacturing environment, which we are not doing. One possible approach is to use traditional metrics and data sets are a means of comparison, and eventually a training set to see if we can incorporate learning into this system.

5. CONCLUSION

Clearly there are major breakthroughs needed before the military logistics problems are solved through just-in-time methods. However, multi-agent technology is quite well suited to address some of these issues. We have demonstrated that parts can be effectively grouped based on a variety of different characteristics of the parts. Agents provide an advantage in this type of problem by specifying a clear goal, but providing several potential ways of achieving it. We have also shown that the results from our agent system can provide a new and interesting perspective to humans about a seemingly well understood problem.

6. ACKNOWLEDGEMENTS

The Oak Ridge National Laboratory is managed by Lockheed Martin Energy Research Corp. for the U. S. Department of Energy under contract number DE-AC05-96OR22424. This work was supported by the Defense Logistics Agency, and the Lockheed Martin Corporation. We would like to thank Dan Carter, Rich Splivalo, and Jim Treadwell for their support and expertise in this project.

7. REFERENCES

[1] J. P. Womack and D. T. Jones, Lean Thinking, Simon&Schuster, 1996.

[2] C. T. Mosier, J. Yelle, and G. Walker, "Survey of similarity coefficient based methods as applied to the group technology configuration problem," Omega, 1997, Vol. 25, pp 65-79.

[3] M. K. Lee, H. S. Luong, and K. Abhary, "A Genetic Algorithm Based Cell Design Considering Alternative Routing," Computer Integrated Manufacturing Systems Vol. 10, No. 2, pp 93-108, 1997

[4] D. Enke, R Kanchitpol, and C. Dagli, "Machine-part Family Formation Utilizing an Art1 Neural Network Implemented on a Parallel neuro-computer," Computers in Industrial Engineering, Vol. 34, No. 1, pp. 189-205, 1998.