A Hierarchical Entity in a Heterarchical Grid of Manufacturing Systems

Daniël Telgen[1,2*], Leo van Moergestel[1], Erik Puik[1], Alexander Streng[1], Roy Scheefhals[1], Tommas Bakker[1], Alexander Hustinx[1], Laurens van den Brink[1,2], and John-Jules Meyer[2]

[1] Research Centre Technology & Innovation
HU University of Applied Sciences Utrecht,
Nijenoord 1, 3552 AS Utrecht, The Netherlands

[2] Department of Information and Computing Sciences
Utrecht University
Princetonplein 5, 3584 CC Utrecht, The Netherlands

Abstract

Automated manufacturing systems are classically hierarchically controlled by a centralized control system, which makes the system deterministic and therefore easier to optimize for efficient manufacturing of a large number of one single product. However, in modern manufacturing the demand for customization and high-mix, low-volume production is growing. This move is strengthened by the acceptation of 3d printers for industrial products and new technologies that make it easier to reconfigure manufacturing systems. Hence, new paradigms like agile manufacturing, which focuses on a shorter time to market, and flexibility are becoming more important to industry. One of these paradigms is grid manufacturing, which uses a group (grid) of autonomous manufacturing systems that can be controlled as a heterarchy (where every system is autonomous and equal to each other). In this paper the goal is to determine, by simulation, if it is useful to develop a hierarchical entity to reserve some of these systems to partly break the heterarchy. This way it would be easier to optimize performance of manufacturing batch products. To fully utilize a grid it would be of interest to be able to use both hierarchical control, where a hierarchical entity reserves specific manufacturing systems, and heterarchical control, where a product can negotiate with any manufacturing system to complete the next step. Since both hierarchical and heterarchical control have advantages it would be of interest to dynamically choose one of both strategies, depending on the current demand.

1. Introduction

Reconfigurable Manufacturing Systems (RMS) are of high interest to industry. The possibility to quickly change a system provides the ability to have a shorter time to market, and scale systems quickly according to current market needs. This flexibility can be approached from different levels, e.g. from modularization of changeable hardware to the agility of the company. An important aspect in this field is how flexible hardware can be controlled. Traditionally, manufacturing systems are controlled from a centralized system in a hierarchical manner.

Classically, a hierarchical or centralized controlled system gives the best efficiency when all systems are predictable. When many disturbances occur, a distributed system with heterarchical aspects (where individual manufacturing systems act autonomously) could possibly achieve better results. Hence, for the production of high-mix, low-volume products, a distributed system, where autonomous systems are not dependent on each other, could be of interest.

For this purpose, low-cost manufacturing systems have been developed that can quickly be reconfigured and changed for the latest demand. These systems have been named *equiplets*, which are low cost, single purpose, and use agent technology to cooperate within a Multi Agent System (MAS) [1]. The MAS is used to represent all equiplets that work together in what is called a *grid* [2]. Products are also represented by these agents and interact freely within this MAS to negotiate scheduling of their products at the equiplets in the grid.

Since equiplets are autonomous and can directly interact with products, it is possible to use 'heterarchical' control, where every equiplet is equal to another. The products make use of each service the equiplets provide, depending on the capabilities of the equiplets configuration. This provides the possibility to manufacture a range of products on one grid of products, offering generic services.

Automated manufacturing systems control architectures can be divided into four basic types: centralized, hierarchical, modified hierarchical, and heterarchical [3]. This paper investigates if manufacturing could be more efficient using some aspects of hierarchical or heterarchical control systems. Since both have their advantages, it

* Corresponding author: E-mail: daniel.telgen@hu.nl, Linkedin: http://www.linkedin.com/in/danieltelgen

discusses the possibility to change between these approaches based on the current demand. Different cases will be researched using a simulator where a number of products are manufactured on a grid of equiplets.

2. LITERATURE OVERVIEW

Leitão [4] and Trentesaux [5] present a survey of the intelligent and distributed manufacturing control systems using the emerging paradigms. Both note there is a lack of proven methodology or tools used in industrial practice. Overall trends in various manufacturing sectors are the move from hierarchical management structures to more leveled structures reducing middle management, i.e. moving towards more modularity, autonomy, and self-sufficiency, at the lowest possible levels [6].

A good overview of Multi-agent scheduling is given by [7]. Brun et al. argue that distributed systems have an edge over centralized systems and propose a multi-agent system for simulation of shop-floor scheduling [8]. Moergestel proposes a multi-agent system for manufacturing that produces a successful robust schedule against disturbances when it is producing at a high load [9].

Even though research in this area becomes more available, these contributions are often informal and fragmented. Maturana focuses on modeling different agents in a manufacturing domain on JADE, where part machines and AGVs are modeled as agents [11]. The application leads to an agent-based simulation with a reactive agent architecture. Barbosa et al. state that simulations are crucial in analyzing behavior during the design phase, and present a simulation designed of studying agent-based control systems for deployment into real operation [10].

3. RESEARCH QUESTIONS

As shown in the literature overview, a variety of research exists on agent technology in manufacturing, and in general for heterarchical vs. hierarchical control systems. Since the use of grid manufacturing using heterarchical control has been created, it is of interest to research what impact this can have when using a grid of equiplets. Using a simulator, the goal is to investigate if it is useful to create classic manufacturing lines in a grid that is originally designed to operate completely heterarchically. This is done by reserving a number of equiplets to only operate batch products. While this lowers the overall flexibility of the grid it could potentially be more efficient when large batches are produced, since travel distance is shorter and the schedule of an equiplet is easier to fill completely because of the predictable sequential steps that have to be performed by the identical products.

To investigate if this could be potentially interesting to implement for grid manufacturing, some research questions need to be answered: 1. Are there cases that potentially increase the efficiency of the grid when equiplets are reserved for batches? 2. What impact do disturbances, e.g. hardware errors or logistic delays, have on a grid when equiplets are reserved by a hierarchical entity? 3. What would be the effect of switching between these strategies and when would this be of interest?

4. SIMULATION

To be able to research different approaches a number of cases will be investigated by developing a grid simulator. The simulator defined a number of equiplets, with specific capabilities and products that can be added dynamically over time. Transport is taken into account by using average travel time over bidirectional paths between equiplets, using distance information, which is calculated with the following formula:

$$travelTime = (abs(A_x - B_x) + abs(A_y - B_y)) \times travelTimePerHop \qquad (1)$$

Where A and B are equiplets, x, y are coordinates, and travelTimePerHop is the average travel time between neighbor equiplets. Other capabilities of the simulator are to add disturbances that delay or stop an equiplet. It is also possible to start production of a batch, which is done by creating a specified amount of identical products over time. The simulator also has a product generator, which can spawn random products that have a random amount of different steps that have to be taken to let it be produced. The system that spawns products can be set to deliver a certain ‘grid load’, taking into account the number of equiplets and the capabilities they provide to estimate the amount of products that are required to calculate the amount of products that will be randomly spawned. The spawn system uses a seeded randomizer together with a setting for the chance of delay of the product to create a spike of product demand.

4.1 *THE PLANNING ALGORITHM*

To research heterarchical vs. hierarchical control, a grid consisting of 9 equiplets is defined. Each of these equiplets has its own set of capabilities. Since equiplets are low cost and single purpose, the set of capabilities are small, varying from 1 to 3 capabilities per equiplet. In our simulation we look at simple products, based on three simple actions, e.g. α to add adhesives, β for rotation operations (bolting or drilling), and γ for a pick and place operation.

Now consider a product P1, with 4 product steps(σ), defined by a capability that can perform that step:

$$< \sigma_1\{\gamma\}, \sigma_2\{\alpha\}, \sigma_3\{\gamma\}, \sigma_4\{\beta\} >$$

This could for instance be the actions required to assemble a sensor. First the electronic parts are placed in a casing, and secondly, adhesive is added to fixate the parts. Then the other half of the casing is placed on top of the adhesive, and finally a screw is driven into the thread to fixate the sensor casing. To plan this sequence, a product agent is spawned that represents the product. The product agent will match all possible equiplets that can perform the necessary production steps that need to be performed, i.e. it compiles a collection, matching the capability of the equiplet with the required product steps. The resulting collection in this case is: $< E_1\{\sigma_2\}, E_2\{\sigma_4\}, E_3\{\sigma_1, \sigma_3\}, E_4\{\sigma_2\}, E_5\{\sigma_1, \sigma_3\}, E_6\{\sigma_4\}, E_7\{\sigma_2\}, E_8\{\sigma_4\}, E_9\{\sigma_1, \sigma_3\} >$

In order to optimize production within the grid, batch scheduling is introduced. Batch scheduling allows for reserving a 'section' of the grid to be used only for reserved batches. This can be seen as a classical assembly line controlled by a hierarchical entity (likely an agent) that reserves the equiplets. When a reservation is made, only products spawned in the annotated batch are allowed to schedule with the reserved equiplets. When applying batch scheduling on the simulation, reserving E1, E2, and E3 for batches, the collections become:

Batches	*Other products*
$< E_1\{\sigma_2\}, E_2\{\sigma_4\}, E_3\{\sigma_1, \sigma_3\} >$	$< E_4\{\sigma_2\}, E_5\{\sigma_1, \sigma_3\}, E_6\{\sigma_4\}, E_7\{\sigma_2\}, E_8\{\sigma_4\}, E_9\{\sigma_1, \sigma_3\} >$

Once this collection has been compiled, the product(agent) starts negotiating with the equiplets to be scheduled. This system was implemented partly based on the work of Moergestel [9], with the difference that we do not optimize for distances between equiplets and do not use a circular system for scheduling, but define a limited time window that moves over time.

In the implementation of the simulation the product agent inquires each of the equiplets to evaluate whether or not the step can be performed with the given parameters at the given equiplet. Once all equiplets have been queried and matched, the actual planning begins. The first step in the planning consists of reducing the transition time between product steps. This happens in several steps: First a production matrix is constructed where the rows represent the equiplets and the columns represent the product steps. It is filled with a score of where a product is best scheduled. The higher the number, the better the choice. This is done with the following steps: All equiplets that are capable of performing a step have their value raised to 1. The next step is to minimize transitions between equiplets. To prevent excess transitions between equiplets during manufacturing, all equiplets with sequential steps have their value raised by the length of the sequence -1. A sequence is defined as the number of steps that the product could perform at the same equiplet. In the current case we use a limited amount of equiplets with only one capability each. Another optimization is load balancing. All equiplets are responsible for their own schedule, this makes it feasible to calculate the load at any given time.

Consider a product querying an equiplet to calculate its load for scheduling its fourth step (σ_4). This step has to be carried out not at release time, but at release time + Δ, where Δ is the time the previous 3 steps will take to be performed, including travel time. The equiplet has to calculate load over that time period, which is called a load window. The length of the window is defined as release time + Δ, until the product's deadline. This is illustrated by the following two schedules:

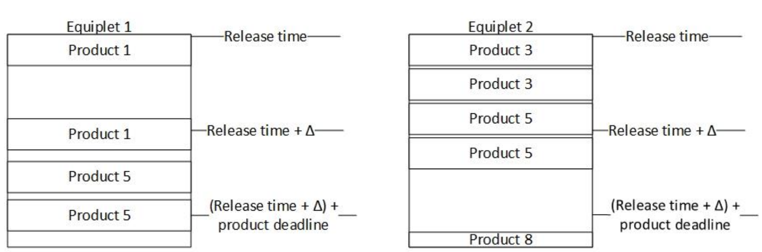

Figure 1: Two examples of equiplet schedules

If the equiplet would calculate load over release time + product deadline, it would not be a correct calculation. This can be seen in Figure 1, where equiplet 1 has a load of 90% (which is represented by 0.9) whereas equiplet 2 has a load of 20%. In order to favor the equiplets with least load, all numbers in the matrix are multiplied with (1 – load). Applying this to our current production matrix, and an imaginary constant load (E1, E2 & E3 – 60%, E4, E5 – 20%, E6, E7, E8 – 40% and E9– 10%), the matrix will result in:

Table 1: Planning with load - the values at E1, E2 & E3 are multiplied by (1 – 0.6) which results in 1 × 0.4

Equiplet{capability}	σ1	σ2	σ3	σ4
E1{α}	0.0000	0.4000	0.0000	0.0000
E2{β}	0.0000	0.0000	0.0000	0.4000
E3{γ}	0.4000	0.0000	0.4000	0.0000
E4{α}	0.0000	0.8000	0.0000	0.0000
E5{γ}	1.0000	0.0000	0.8000	0.0000
E6{β}	0.0000	0.0000	0.0000	0.6000
E7{α}	0.0000	0.6000	0.0000	0.0000
E8{β}	0.0000	0.0000	0.0000	0.6000
E9{γ}	1.0000	0.0000	0.9000	0.0000

As seen in Table 1, Equiplet 1 (E1), which has a load of 60%, is suddenly not the best option for step 2 anymore. This way, a proper load balance between the equiplets is achieved. Another optimisation is reducing the amount of time a product travels within the grid. This can be achieved by calculating the different paths through the grid over all steps, and applying these values to the production matrix. Because optimising travel time was not a priority in this simulation, it has not been implemented at this time.

5. SIMULATION TEST RESULTS

The simulation is performed using products which use three different capabilities, and is done over time, shown in timeslots. Every product step takes different time to perform. It takes 20 timeslots (relative period) to use capability α, 10 timeslots to use capability β, and 5 timeslots to perform capability γ. The simulation presents 2 different cases. Case 1 will show a heterarchical strategy vs. a hierarchical strategy, by using an entity that reserves equiplets for batch production, while case 2 will show what happens when switching from hierarchical control to heterarchical control when disturbances occur.

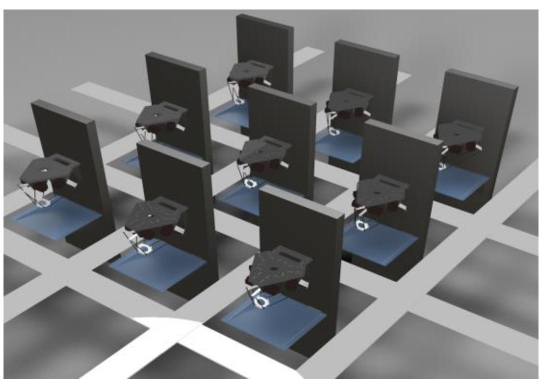

Figure 2: Type A grid = a 3 x 3 grid with heterarchical control

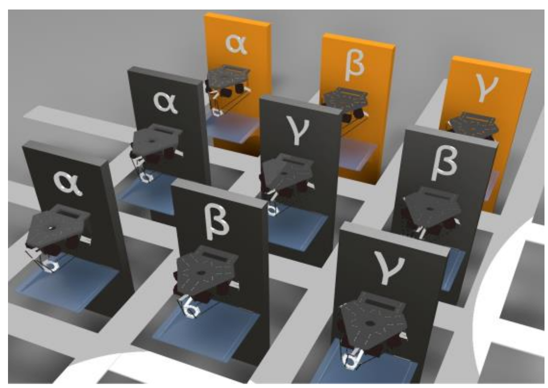

Figure 3: Type B grid = The same grid then in figure 2, but also showing capabilities and three reserved equiplets in the back that will only be used by specific batched. This is typical for this type hierarchical control

Figure 2 shows a grid of equiplets that will be used in the first case. In case 1 we try to investigate the difference between Grid A, as shown in Figure 2, and Grid B, as shown in figure 3. The only difference between these two grids is that with type B, three equiplets are specifically reserved for batch of products. In this case these reserved products are not optimized for the specific batch product.

5.1 CASE 1A - HETERARCHICAL CONTROL

In case A, all equiplets are equal, i.e. set for heterarchical control. All equiplets can be used by any product. To simulate these 351 timeslots (which represent 2 hours) are simulated, spawning 35000 random products, of which 3600 products in a batch. The products use all three capabilities that are available in the grid, and have a deadline of 86 seconds to be completed. This deadline and the amount of products are chosen such as that the grid performs at an estimated 80% average load, using the random product spawn system. Figure 4 shows the load of all 9 equiplets.

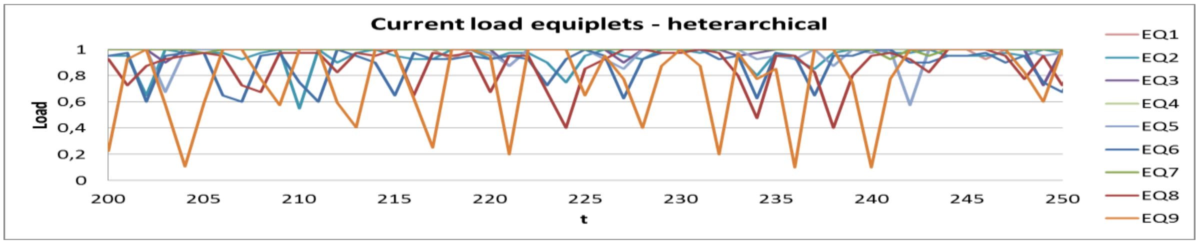

Figure 4: The load (1 represents 100%) of 9 equiplets in case 1A

Figure 4 shows, as expected, a high load on all equiplets. Basically the equiplets with capability α have the highest load, which could be expected since this action takes the longest to perform. This is made even clearer in Table 2, which shows the average load over the shown time period for each equiplet and the three equiplets with the same capability. This will be analysed further after the next case.

Table 2: Average load during shown time period, per equiplet and capability

E1{α}	E2{β}	E3{γ}	E4{α}	E5{γ}	E6{β}	E7{α}	E8{β}	E9{γ}	α	β	γ
1.0	0.94	0.99	1.0	0.97	0.89	1.0	0.88	0.77	1.0	0.90	0.91

5.2 CASE 1B - HIERARCHICAL CONTROL THROUGH RESERVATION OF EQUIPLETS FOR BATCHES

In case 1B we look at the same result while spawning products also around an estimated 80% load average, but then using three equiplets that are reserved specifically for the batches.

Figure 5: The load of 9 equiplets in case 1B – with reserved equiplets for batches

Table 3: Average load during shown time period

E1{α}	E2{β}	E3{γ}	EQ{α}	E5{γ}	E6{β}	E7{α}	E8{β}	E9{γ}	α	β	γ
1.0	0.5	0.5	0.84	0.91	0.80	0.77	0.81	0.69	0.87	0.70	0.7

Figure 5 and table 3 show the load in this case. Equiplet 1, 2 and 3 are now reserved for batches. Equiplet 1 has a 100% load, as was also the case in case 1A. This can be explained, since the α capability takes the longest to perform. However, all other equiplets (both reserved and non-reserved) have a lower average load. Also Figure 5 shows that the reserved equiplets have a stable load, since they are continuously manufacturing identical batch products.

5.3 CASE 1 – ANALYSIS

From image 4 and 6 it can be determined that reserving equiplets provides a stable, but also lower, load on the other non-reserved equiplets when using the random spawning system. The cause of this is the lower choice of free equiplets to the random products, which leads to less possibilities to be manufactured in time and an expected lower throughput (finished products over time). However, as shown in figure 6 there are more differences.

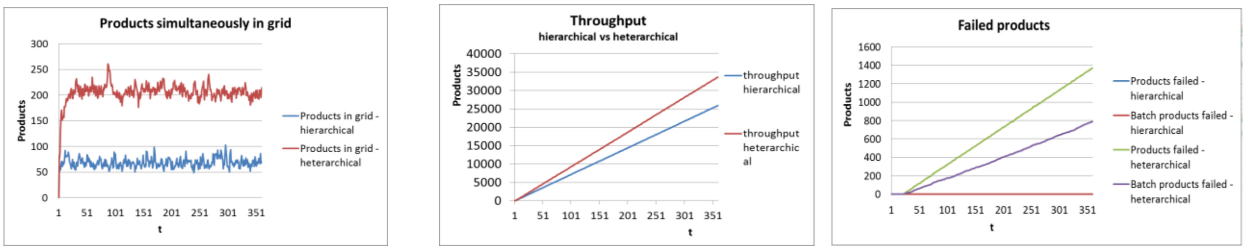

Figure 6: 6A-left: Active products in grid. 6B-middle: throughput (completed products). 6C-right: failed products

Figure 6A shows that the number of active products in the grid is substantially higher when no equiplets are reserved, while throughput, as shown in Figure 6B, is just slightly lower). This can be explained by a shorter travel distance and therefore a faster completed product per equiplet. Figure 6C also shows that some equiplets were overloaded when using heterarchical control, which led to more failed products that were unable to reach their deadline.

It can be concluded that heterarchical control has a potential higher throughput in some case, at the cost of production time per equiplet and potential overload of the equiplets. Also, in a practical case using real equiplets, the higher amount of active products in a grid might give challenges for logistics, with possible delays due to the longer travel time.

Case 1 shows that both hierarchical control, where an entity reserves some equiplets for batches, and heterarchical control, when all equiplets can be equally used by any product, can have benefits. Hierarchical gives less chance of failed products, a more predictable load on the system if there are many batches, and has less active products in the grid. Heterarchical control has a potentially higher throughput when the reserved equiplets are not optimized for the batch product. This gives the grid the ability to manufacture more products in the presented case at the cost of some products not meeting their deadline and predictability.

We did not show a test case where reserved equiplets would be chosen based exactly on the needs of the batch product, since it this would clearly show the reserved equiplets running near 100% capacity as long as there was a demand for batch products, at the cost of the heterarchical part of the grid that would have limited capacity for the random products.

6. CASE 2– ERROR BEHAVIOR – SWITCHING FROM HIERARCHICAL RESERVED TO HETERARCHICAL

Case 1 showed that both hierarchical control and heterarchical control with reserved equiplets have different advantages. However, it is evident that disturbances will have a larger effect on hierarchical control. If the reserved equiplets encounter any kind of disturbance the batches will stop producing, which leads to the failure of all batch products until the problem that initiated the disturbance can be resolved. However, since equiplets can be reserved by a hierarchical entity, but are designed to be autonomous, it is of interest to investigate if switching from hierarchical to heterarchical provides the ability to mitigate these disturbances. This could also open future research for dynamical reservations when large batches are expected.

To investigate this case we take the same simulation settings as in the previous case. This means an average of 80% load is maintained on the grid by using random products, whilst spawning 1 batch product every 2 seconds. The same grid was used as in case 1b, as was shown in Figure 3. After timeslot 180, equiplet 3 encounters a disturbance and goes into an error state which renders it unavailable for further production. As a result of the disturbance, all reserved equiplets that are able (without error) will change to a heterarchical state. The batches will also be able to make use of all available equiplets.

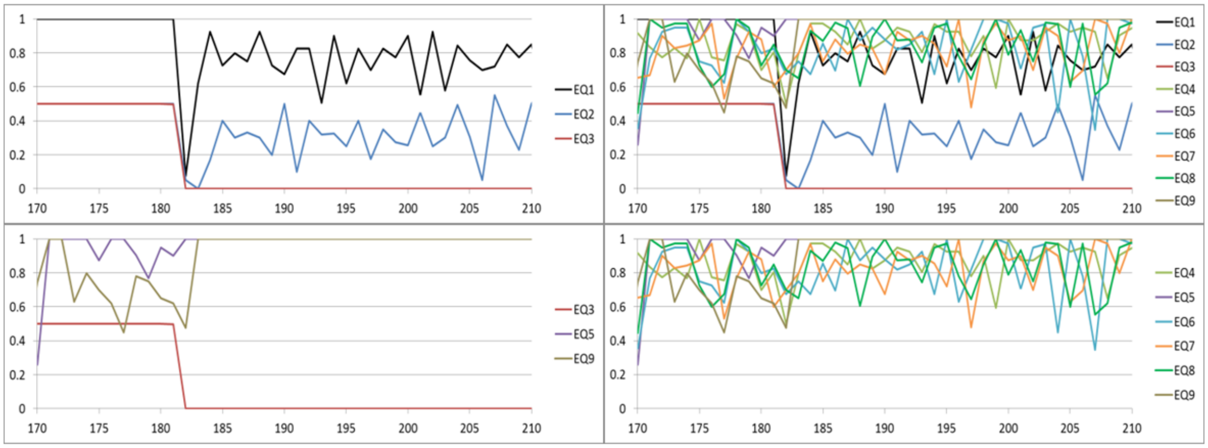

Figure 7: load on the Y axis, timeslots on the X axis, 7A-top left: load of the reserved equiplets, 7B-top right: load of all equiplets. 7C-bottom left: all equiplets with capability γ, 7D-bottom right: original heterarchical equiplets

At figure 7A the disturbance that stops equiplet 3, with capability γ, can be clearly seen. This would normally stop all batch production. However, in this case equiplet 1 and 2 are immediately switched to heterarchical mode and batches are allowed to be rescheduled at any available equiplet in the grid. While this could potentially lower efficiency (depending if the reserved equiplets are efficiently chosen for the batch products), the problems of the disturbance are mitigated since equiplet 5 and 9 immediately compensate for the disturbance. This can clearly be seen in figure 7C, which shows equiplets 5 and 9, which have a similar capability as equiplet 3, immediately increase their load to a 100%. As shown in Figure 7D the impact on the equiplets that were originally heterarchical is only slightly noticeable.

6.1 CASE 2 ANALYSIS

Case 2 shows the use of being able to switch back to heterarchical manufacturing. Figure 8 shows how many products would be unable to reach their deadline if the batches were not allowed to change to the heterarchical system and if they would be allowed to.

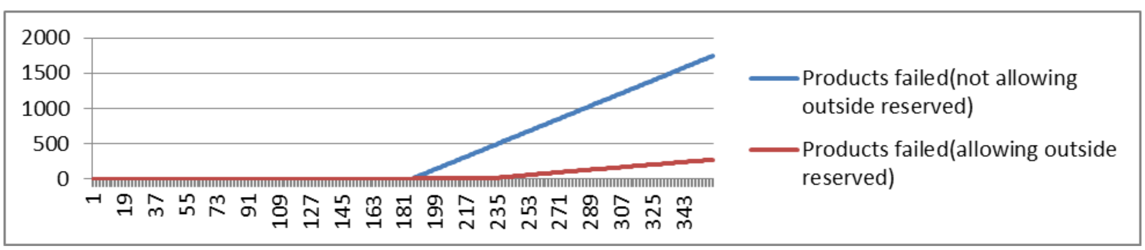

Figure 8: Shows the amount of failed products

As expected, many more products fail after the disturbance occurs. Switching to heterarchical clearly provides a lower amount of products that are unable to reach their deadline. However, some products still fail after the disturbance due to the increased travel time and higher load of the other equiplets that have to take over for the disabled equiplet.

7. DISCUSSION

Case 1 might be considered slightly arbitrary, since the results are highly dependent on a large number of variables including: size of grid, composition of capabilities, required steps for the product, the deadline and their randomized starting point. However, this was the intended point. In grid manufacturing, products can be dynamically added to the grid and the simulation shows that grid manufacturing can deliver this large range of possibilities. Since equiplets are also reconfigurable in nature, it would be possible to quickly reconfigure the hardware to accommodate large batches of identical products and form an efficient line using the hierarchical entity that reserves equiplets for these products. This would even be substantially more efficient than shown in case 1B, since here the reserved equiplets were not specifically updated for the used batch product.

While the reserving of equiplets does make the grid more susceptible to error, case 2 proves that the possibility to switch back to full heterarchical use of all equiplets provides the ability to deal with disturbances. Considering grid manufacturing was originally intended as a fully heterarchical way of manufacturing, adding the hierarchical entity for reserving batches provides many interesting possibilities.

8. CONCLUSION

To answer the original research questions: This paper shows that there are cases that could benefit from both hierarchical and heterarchical strategies. Based on the needs (size of batches, acceptable failure to meet the deadline and the grid capabilities) choices can be made which strategy to choose. However, disturbances do have a high impact on batch production, which can be mitigated through switching back to heterarchical control. Hence, the conclusion is that it is of high interest to consider both the heterarchical and hierarchical approaches to optimally utilize the possibilities of grid manufacturing.

This paper impacts the way autonomous manufacturing systems are used in a grid. While the original goal of grid manufacturing was to create the maximum amount of flexibility, this paper shows that it is of importance to be able to limit this flexibility to create a higher efficiency for any given situation. While this might seem intuitively right, no systems in manufacturing have been known to utilize this possibility. Hence, it can be concluded that the development of a hierarchical entity that can reserve equiplets for batch production is a valid and interesting research subject for the future.

9. FUTURE WORK

Considering the results of this paper, and the generic possibilities that grid manufacturing provides, it would be of interest to continue the research in several steps: 1. Analyse a large amount of cases to find a heuristic that could determine when reserving equiplets would be beneficial. 2. Use the heuristic to develop an automatic system that could select equiplets to be reserved when this would be beneficial. 3. Integrate this system into the real (physical) grid using agent technology to test the practical implications of the proposed system.

REFERENCES

[1]. *M. Wooldridge, "An Introduction to Multi Agent Systems", Second Edition, Wiley & Sons, 2009.*

[2]. *E. Puik and L.J.M. van Moergestel, "Agile multi-parallel micro manufacturing using a grid of equiplets", IPAS 2010 proceedings, pp. 271-282, 2010.*

[3]. *D. Dilts, N. Boyd, and H. Whorms, "The evolution of control architectures for automated manufacturing systems," Journal of Manufacturing Systems, vol. 10, no. 1, pp. 79 – 93, 1991.*

[4]. *P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey", Eng. Appl. Artif. Intell., vol. 22, pp. 979–991, Oct. 2009.*

[5]. *D. Trentesaux, "Distributed control of production systems," Engineering Applications of Artificial Intelligence, vol. 22, no. 7, pp. 971–978, 2009, Distributed Control of Production Systems.*

[6]. *M. Mehrabi, A. Ulsoy, and Y. Koren, "Reconfigurable manufacturing systems: Key to future manufacturing," Journal of Intelligent Manufacturing, vol. 11, no. 4, pp. 403–419, 2000.*

[7]. *M. de Weerdt and B. Clement, "Introduction to planning in multiagent systems," Multiagent Grid Syst., vol. 5, pp. 345–355, Dec. 2009.*

[8]. *A. Brun and A. Portioli, "Agent-based shop-floor scheduling of multi stage systems," Computers & Industrial Engineering, vol. 37, no. 12, pp. 457–460, 1999, Proceedings of the 24th international conference on computers and industrial engineering.*

[9]. *L. van Moergestel, E. Puik, D. Telgen, and J-J. Meyer, "Production scheduling in an agile agent-based production grid", Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on, vol. 2, pp. 293 – 298, 2012.*

[10]. *J. Barbosa and P. Leitao, "Simulation of multi-agent manufacturing systems using agent-based modelling platforms," in Industrial Informatics (INDIN), 2011 9th IEEE International Conference on, pp. 477–482, 2011.*

[11]. *F. Maturana, R. Staron, K. Hall, P. Tich, P. lechta, and V. Mak, "An intelligent agent validation architecture for distributed manufacturing organizations," in Emerging Solutions for Future Manufacturing Systems (L. Camarinha-Matos, ed.), vol.*

159 of IFIP International Federation for Information Processing, pp. 81–90, Springer US, 2005.