Self-Reconfigurable Manufacturing System For Personalized Mass Customisation

Rotimi Ogunsakin, César A. Marín, Nikolay Mehandjiev
Alliance Manchester Business School
Booth St E, Manchester, M13 9SS
Email: {rotimi.ogunsakin, cesar.marin, nikolay.mehandjiev}@manchester.ac.uk

Abstract— **One of the expected benefits of Industry 4.0 is the ability of production systems to effectively produce personalized products for a relatively small lot size in the same assembly line without trade-offs for cost, delivery time and quality. In this case, individual product will be unique and require a production system that is optimized for single lot size (lot-size-of-one). Reconfigurable Manufacturing System (RMS) is an attractive choice for this. However, reconfiguration processes are not automated and not real-time, which sometimes requiring days, thereby making it economically and functionally unfit for personalized mass customisation. To achieve this, Self-Reconfigurable Manufacturing System (S-RMS) is proposed in this paper and implemented in-silico using nature inspired models and algorithms. In our implementation, resource re-configuration is both automatic and immediate. The system is evaluated by measuring total production output, system stability and average lead-time per order during production and during unexpected changes like resource breakdown. This approach is expected to proffer solution to the batch-size-of-one problem in manufacturing and engender personalized mass customisation.**

Keywords- Self-Recofigurable Manufacturing System; Reconfigurable Manufacturing System; Personalised Mass Customisation.

I. INTRODUCTION

Personalized mass customisation is the mass production of individually customised goods in the same production line [1]. This paradigm requires manufacturing systems capable of producing a relatively high volume of unique product options in the same assembly line without trade-offs in cost, delivery time and quality for both manufacturers and consumers [2]. The major challenge in developing a manufacturing system with such capability is the level of structural, control and software flexibility that is required [3]. Present days manufacturing systems cannot be modified to offer the level of structural, control and software flexibility that is required in such system and therefore requiring a new paradigm approach.

There are proposed manufacturing systems expected to give rise to personalized mass customisation, such as Reconfigurable manufacturing system (RMS), but these systems suffer from structural and control inflexibility and therefore functionally unfit for personalized mass customisation [3]. However, Evolvable assembly system (EAS) is an exception to this; it is demonstrated to achieve personalized mass customisation and address the batch-size-of-one problem, but the system achieves this by producing a single unit at a time [4]. In practice, personalized mass customisation scenarios are more complex. They require multiple products on the shop floor with different designs, styles, shapes and colors going through production process at the same time. EAS at present fails to demonstrate such practical complexity, which should be obtainable in a typical manufacturing system capable of personalized mass customisation.

This research proposes and demonstrates in silico Self-Reconfigurable Manufacturing System (S-RMS), a manufacturing system capable of addressing the batch-size-of-one problem in personalized mass customisation. This is achieved by borrowing natural self-organising rules from natural systems, extending and adapting them to reconfiguration in S-RMS to incorporate adaptive properties, such as self-reconfiguration for multiple product mix and self-recovery during machine (resource) failure.

The system is evaluated by measuring average production output, stability and average lead-time per unit during production and during unexpected changes like resource breakdown. The expected benefit of this approach is that, it will proffer solution to the batch-size-of-one problem in the manufacturing systems and engender personalized mass customisation.

The remaining of this paper is organised as follows. Section II reviews related work in reconfiguration and mass-customisation in manufacturing systems. Section III contains details of the approach used, which include models and algorithms, while Section IV contains the simulation and experimental evaluation of the S-RMS. The final section contains the discussion and conclusion.

II. RELATED WORK

Advancements in manufacturing systems have been mainly to increase efficiency and reduce production cost and lead-time. Two approaches have been dominant in achieving these, which are 1) Resource flexibility and 2) Resource layout.

Resource flexibility is the technological improvement on production machines for faster part production, wastage reduction and ability to produce more than one part variety. Resource layout is the spatial arrangement of resources on the factory floor in such a way as to optimize production process [5].

The proposed S-RMS is based on how personalized mass customization can be achieved through reconfiguration of resource layout in real-time, which is one of the major challenges in manufacturing systems. Therefore, reviewed work will be focused on resource layout and reconfiguration approaches.

Resources in manufacturing system can be arranged based on function or process requirement, which are referred to as functional and cellular layout respectively. Functional layout is a resource layout paradigm that is based on resource function. This implies having resources of the same type placed in the same location [6]. This provides economy of scales and simplicity in workloads allocation, but highly inefficient when there are constant or unpredictable changes in product mix and routings, a common scenario in personalized mass customisation.

Cellular layout is a layout configuration in which the factory is partitioned into cells, and each cell is dedicated to a product or part family with similar processing requirements [6]. This type of configuration simplifies workflow and is generally optimized for producing specific product set with stable demand and long product life cycle. However, reconfiguration process is usually expensive and time consuming, making such layout unattractive for personalized mass customisation [6].

As a result of the inefficiencies in both functional and cellular layouts for dynamic production environment, other layouts have been proposed for a more efficient production output in dynamic production environment [5]. Some examples of the proposed layouts are: Spine layout, Hybrid layout, Multichannel layout, Distributed layout and Modular layout (see Table 1).

TABLE I. LAYOUT TYPES AND DESCRIPTION [5]

Layout types	Description
Spine layout	Usually used by Original Equipment Manufacturers (OEMs). In this type of layout, products move along a main artery through the plant. Mini-assembly lines owned by independent suppliers are linked to the spine where additional modules are attached to the product as needed by these suppliers as it moves through the spine. This allows for change of suppliers without changes to the factory layout.
Hybrid layout	This is a combination of different production modules based on multiple production process. For example, a hybrid facility may contain flow-line component for manufacturing common parts and a job-shop component for customizing final products.
Multichannel layout	This involves having duplicate production lines that are shared across products. Products are allowed to move in and out of neighbouring production lines, thereby creating multiple lines and channels, and minimizing queue and congestion.
Distributed layouts	In a distributed layout, not all equipment of the same type is placed in adjoining location. Instead, equipment of the same type is placed individually throughout the factory, which can be quickly combined to form temporary cells dedicated to specific product line or job order.
Modular layout	This conceptualises layout as a network of basic modules. These basic modules may be based on different production process or layouts. For example, a modular layout may contain a network of flow-line, job-shop, cellular layout and functional layout basic modules.

However, these resource layout types are not optimized for production environment with requirement for single-lot-size production [4]. To address this limitation, a system with highly flexible layout is required, where reconfiguration can be done with a minimal amount of time and at no additional cost to both the manufacturer and the consumer. To achieve this level of manufacturing flexibility, smart, flexible and adaptive manufacturing system is proposed.

Manufacturing companies (such as GE, Airbus, Siemens) are observed to be investigating smart, flexible and adaptive manufacturing systems that are capable of autonomous self-healing, self-adaptation and self-reconfiguration, typified by the “batch-size-of-one” (BSo1) problem [3]. Examples of such system include: Reconfigurable Manufacturing System (RMS), Holonic Manufacturing System (HMS), and Evolvable Assembly System (EAS).

Reconfigurable Manufacturing System is a manufacturing system designed to enable rapid change in hardware and software component for quick response to sudden market changes by adjusting its functionality and production capacity [7]. RMS is engineered for mass production and therefore not effective for managing large product varieties and rapid changes in market. This makes RMS unsuitable for personalized mass customization.

Holonic Manufacturing System, which is inspired by Arthur Koestler’s holons concept [8]. Holons are autonomous self-reliant units with degree of independence, such that contingencies can be handled without being instructed by higher authority, and simultaneously subjected to control from single or multiple higher authorities [9]. This implies that Holons can exist in complex systems like manufacturing systems as both a whole and a part simultaneously.

The “whole” property ensures stability of forms in the system, while the “part” property signifies intermediate forms and ensure stability for higher form. Holons concept comparatively provides more flexibility for manufacturing systems, but immediate reconfiguration is still lacking, therefore making it not suitable for personalized mass customisation [9].

Evolvable assembly system is a production system whose components are designed to adapt to changing conditions of operations and also assist in the evolution of the component in time, such that processes utilizing the components will become self-evolvable, self-reconfigurable, self-tuning and self-diagnosing [10]. Examples of EAS implementation are plug and produce system, and Smart Manufacturing and Reconfigurable Technologies (SMART).

Plug and produce system is an implementation of Instantly Deployable Evolvable Assembly System (IDEAS), which is aimed at developing an industrially suitable EAS [10]. The plug and produce system was implemented on a mini scale called MiniProd [11]. It is based on multi-agent control paradigm and capable of real-time self-reconfiguration on the shop floor without higher-level instruction. This shows that real-time self-reconfiguration is possible at machine level using distributed control paradigm.

On the other hand, SMART is a demonstration of the application of adaptive agent control in the transformation of

 ISBN: 978-1-61208-610-1

legacy manufacturing system into a RMS. SMART is demonstrated to be capable of addressing the “batch-size-of-one” problem [4].

Both IDEAS and SMART are implemented using distributed control and they both addressed the batch-size-of-one problem, but only based on single unit production. This means the system can only produce one unit at a time. In practice, personalized mass customisation scenarios are more complex. They require multiple products on the shop floor with different designs, styles, shapes and colours going through production process at the same time. IDEAS and SMART failed to demonstrate this type of complexity, which should be obtainable in a typical manufacturing system with capability for personalized mass customisation.

III. SELF- RECONFIGURABLE MANUFACTURING SYSTEM (S-RMS)

Manufacturing systems with capability for personalized mass customisation are characterized by the potential for single-lot-size production [10]. This implies that individual product will require distinct production plan, schedule and process. Concurrent execution of these plans and processes during production suggests that different products will require different resources and routes at same or different time slot, depending on the production stage of the product.

The complexity expressed by this process necessitates the use of distributed coordination mechanism for dynamic and autonomous route selection; resource discovery, selection and negotiation; and scheduling. Therefore, applying the present manufacturing system’s design approach that is characterized by stationary machines and rigid conveyor belt with pre-defined route for products will be unfit for this purpose. This is because multiple products will be manufactured concurrently, which are unique with distinct and distributed production plan, schedule, process, resource and route requirement. Therefore, S-RMS is proposed to fill this gap by proffering solution to the batch-size-of-one problem.

S-RMS is a manufacturing concept whereby machines can autonomously move around during production, instead of remaining stationary. In this system, there is no conveyor belts, but instead parts and products are transported using mobile robot referred to as products. The underlying complexity of the proposed production system, which is a characteristic of the distributed and distinct nature of the products and physical mobility of machines, suggest the use of nature inspired approach in the design and implementation of the system.

Nature inspired approach is used in this case because the properties of the problems-space, which is the spatial arrangement of products and resources in the production system change constantly with time. This is as a result of the ability of both the product and resource to move freely without any constraints during production. Also, the process of arriving at optimal production strategy for individual product changes concurrently with the problem-space. This is because the process required by a product to discover the closest resource changes constantly as a result of constant changes in the spatial arrangements of these resources. Therefore, the coordination process has to constantly adapt to the problem dynamics. This suggests that a pure computational approach will be expensive if feasible compare to a heuristic approach proffer by taking inspiration from nature.

Ant colony and flock of birds exhibit one of the closest behaviors to the proposed S-RMS in term of underlying complexity and physical mobility of machines and products. Therefore, inspiration is taken from ants and birds.

In biological systems, such as ant colony and flock of birds, these systems are based on entities that exhibit simple behavior, made up of small set of simple rules, with reduced cognitive abilities, and global system of behavior emerges from a multiplicity and reinforcement of non-linear interactions [12]. In such complex natural systems, coordination emerges without a predicted plan or template, not driven by a central entity or global rules, and only become observable at a macro level when the resultant behavior of the whole are greater and more complex than the sum of the behavior of its part [14]. This makes the application of coordination, self-organization and emergence behavior in biological system to S-RMS for personalized mass customisation a very viable alternative solution to purely computational approach [15].

In some species of ants (social ants), pheromones are used as a coordination mechanism by means of indirect or environmental mediated coordination [13]. Information about food location are embedded in pheromones and when perceived by others, it is interpreted and the result of such interpretation informed the next action to be taking by the perceiving entity. With very limited intelligence, ants are able to effectively coordinate activities regarding foraging and construction. When viewed at a higher level, a well-organized and intelligent social system is perceived. This is purely as a result of indirect or environmental-mediated coordination through the use of pheromones.

On the other hand, some species of birds have to migrate from one region to another in search of food. This instinctive behavior can be viewed as a profit oriented strategy. The birds consumed energy in search or migrating to get food. Therefore, to guarantee survival while searching for food, they must use less energy than they will get from the food. To achieve this, birds have to migrate and stay close to food sources, which lower the cost of foraging in the long term. They also store the locations where foods are available including the time of the year in memory. As food availability changes due to weather conditions, this information is also updated.

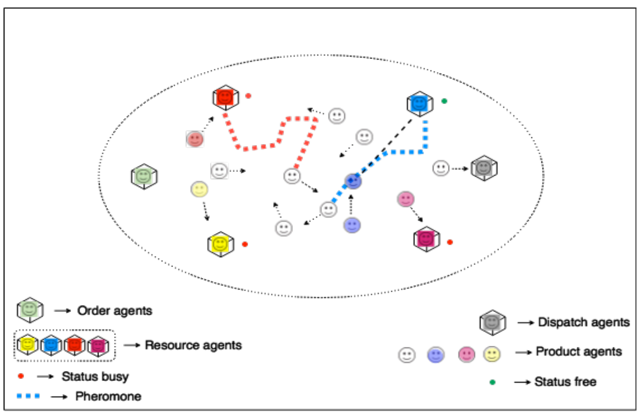
S-RMS is modeled as interaction between two distinct entities on the shop floor, which are products and resources (both products and resources are mobile). Products seek resources to execute production task in their production plans just like ants seek for food in their environment. While resources at the same time seek to be close to products just like birds migrate in order to stay close to food source. Resources achieve this by minimizing the average distance between them and corresponding products on the shop floor during production. Applying this model, a natural equilibrium is expected over a period. Natural equilibrium implies a situation where a resource is able to establish a location on the shop floor that is optimal for task execution and the product for plan execution.

A. S-RMS Multi-Agent System (MAS) Interaction

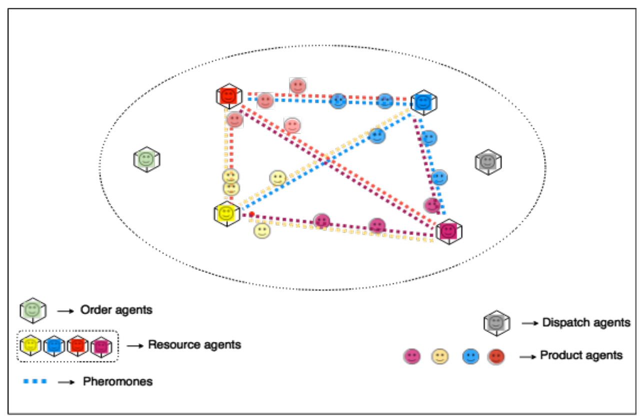
The concept of Multi-Agent System (MAS) is used in the design of S-RMS, where products and resources are both agents operating and interacting on the shop floor. These two agents make different observation about the production environment and therefore have different knowledge and belief about the production environment. The product-agents have knowledge of what to produce, processes required to produce them and type of resources required to carry out such processes. However, individual product-agent has no knowledge of the spatial location of these resources. Instead, product-agents discover resources through interactions with other product-agents and location of resources are communicated using pheromones. This is referred to as indirect or environmental mediated coordination [13].

Resource-agents possess knowledge of what production processes they can execute but have no knowledge of where such production processes are located. Instead, the resource-agents rely on foraging strategy, by following the Circulant Traversal Rule (CTR), which will be explained in later section.

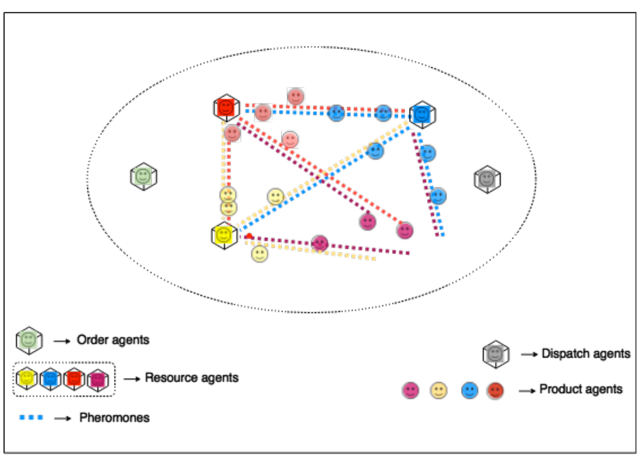
During agent interaction, when a product-agent discovers a resource, it drops pheromones (signals) in the production environment. Other product-agents close-by perceive (sense) these pheromones and if the pheromones lead to a required resource, the perceiving product-agent compute the shortest path to the resource and advance in that direction. If the product-agent is able to execute its plan successfully, it drops more pheromones to intensify the signals as demonstrated in Figure 1(a). The resource-agents on the other hand simultaneously and independently seek to stay close to corresponding product-agents using the CTR.



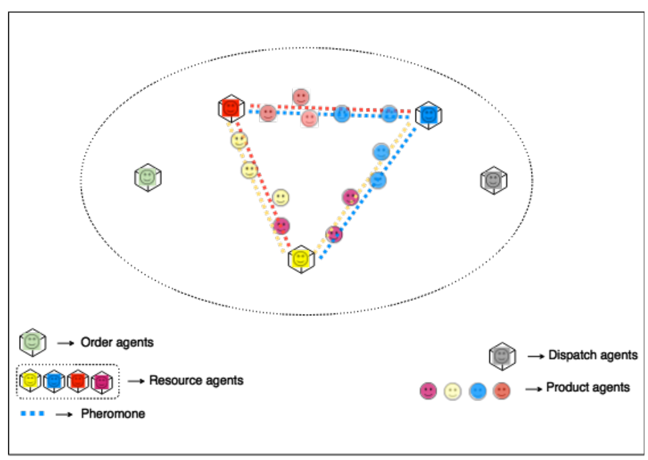
(a): No configuration at the beginning



(b): Optimal configuration achieved over a period



(c): Machine failure during production



(d): Reconfiguration after machine failure

Figure 1: High level observation of S-RMS convergence and reconfiguration after a resource failure

 ISBN: 978-1-61208-610-1

Over a period of interactions, coordination begins to emerge, the system gradually evolve from state of disorder as seen in Figure 1(a), to a state where order is observed as seen in Figure 1(b). Without a physically layout path (conveyor-belt), The system self-organizes itself and form an optimal path for production process using pheromones in the form of virtual marks (virtual conveyor belt) on the shop floor as seen in Figure 1(b). The system is however far from equilibrium as the configuration constantly changes in real time depending on the product mix, resource availability and so on. For example, if a resource suddenly becomes faulty, the system re-configures itself for optimal production from 4 resources as shown in Figure 1(c) to 3 resources as shown in Figure 1(d) without halting production process.

B. Pheromones For Product Agents in S-RMS

Applying the concept of pheromone in computational system requires formalization and abstraction of the natural concept into computational concept. To achieve this, we formalize our own concept of pheromones. The following two conditions are necessary for the pheromone properties and behavior in the S-RMS:

C1: Pheromone

$$\forall f_r \ \exists \theta_p \in (R,\theta) : f_r \in F \leftrightarrow \theta_p \in (R,\theta) \qquad \text{(I)}$$

For a particular pheromone fr in all pheromones F in the system leading to a particular Resource R with the capability of executing production plan θ, there exist a production plan θ_p belonging to a product P that is executable by the Resource R, such that f_r will always lead to Resource R. This implies that all pheromones in the system lead to at least one resource, and such resource is capable of executing at least one production plan.

C2: Pheromone decay rate

$$\forall f_r \in F \ \exists \ \delta_{f(t)} : \lim_{\to \infty} \delta_{f(t)} \to \lim_{\to 0} f_r \, , where \ t \in \mathbb{R}^+ \qquad \text{(II)}$$

For a particular pheromone f_r in all pheromones F in the system leading to a particular Resource R, there exist a pheromone decay rate $\delta_{f(t)}$, such that as $\delta_{f(t)}$ increases, the intensity of f_r decreases. This implies that all pheromones in the system have a decay rate, which is inversely proportional to the intensity of the pheromone.

The product-agents use pheromone for resource discovery, that is, to locate resources on the shop floor that is capable of executing their production plan(s) during production. This is achieved by using the pheromone resource discovery algorithm as shown in Figure 2.

```
--------------------------------------------------------------------------------
PHEROMONE RESOURCE DISCOVERY ALGORITHM (PODUCT-AGENT)
--------------------------------------------------------------------------------
Input: product_info, max_x, max_y
Variables: P:Production_Plans; ϕ:shop-floor;
Pa:Product_agent; Pa(x,y):Product_agent_location;
Ra:Resource_agent; Ra(x,y):Required_Resource_agent_loc;
RR:Required_resource; Loc:New_location;
Plan[i]:Production_plans;
Program:
Plans[i] ← create_plans(product_info)
While Pa(x,y) < max (ϕx,y) && i < max Do
  If Pa(x,y) == Ra(x,y);
    Break;
  Else
    Pa → Sense_pheromone (ϕx,y);
    If pheromone → within_range && Leads_to(RR)
       Loc ← shortest_path(Ra(x,y));
       Move_to (Loc);
       Break;
     End if
       Search_for(RR);
      If found(Ra(x,y)) == True && Pa(x,y) == Ra(x,y);
        Break;
      Else
        Continue_search(RR);
      End if
  End if
End While
Execute_plan;
Plan[i] ← Plan[i++];// move to the next plan
End.
```

Figure 2: Algorithm for resource discovery using pheromones

Product-agent P_a generates an array of production plans and corresponding required resources to execute generated plans. It searches for required resources on the shop floor ($\phi_{x,y}$) and if found, it executes production plan and seek resources for the next production plan and leaves pheromones leading to the resource in the environment.

If instead, a pheromone is sensed within the shop floor ($\phi_{x,y}$) while searching for resources and it leads to the location of required resource agent $R_{a(x,y)}$, then P_a computes the shortest path to $R_{a(x,y)}$ and set its destination (Loc) to the location of the resource agent $R_{a(x,y)}$. If the required resource is met at Loc, then P_a executes production plan, seek resources for the next production plan and leaves pheromones leading to the resource in the environment. Else, P_a continues searching for required resource agent. However, if the executed production plan is the last plan to be executed, then P_a ends production and exits system.

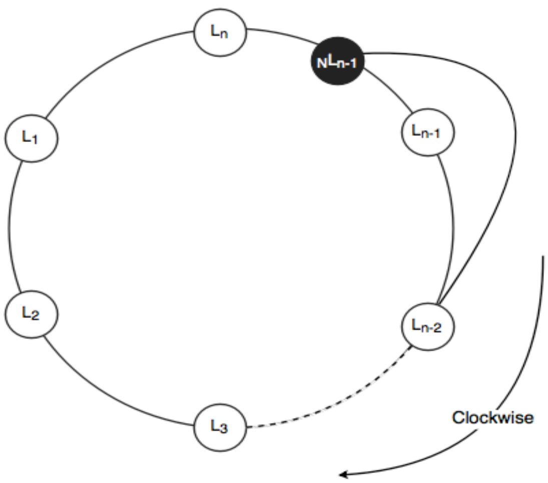
C. Foraging For Resource Agents in S-RMS

Foraging birds are abstracted as resource-agents. These are resources in the S-RMS for part production. These resource-agents are mobile and use the foraging strategy referred to as CTR for navigation as shown in Figure 3. The CTR provides a heuristic means by which resource-agents are attracted toward areas in the shop floor where there is a higher chance of executing production task. This is achieved computationally by rotating the task execution matrix one element to the right.

The CTR is used for two purposes, first to update spatial location where mobile resource-agents execute production task and secondly, as a tracking mechanism to track locations on the shop floor where chances of order executions are comparatively higher. This rule ensures each resource-agent is positioned close to product-agents with production plans

 ISBN: 978-1-61208-610-1

that it can execute. This process is achieved purely by heuristic using the Circulant Traversal Algorithm (CTA) as shown in Figure 4.



*Traversal Rule: If a location **L** is found between $\boldsymbol{L_n}$ and $\boldsymbol{L_{n-1}}$, then set new location as $_N\boldsymbol{L_{n-1}}$ and pop $\boldsymbol{L_{n-1}}$ out of memory*

Figure 3: Foraging Strategy: Circulant Traversal Rule

CIRCULANT TRAVERSAL ALGORITHM (RESOURCE AGENT)

```
Input: product_info, max_x, max_y
Variables: P:Production_Plans; ϕ:shop-floor;
Pa:Product_agent; Pa(x,y):Product_agent_location;
Ra:Resource_agent; Ra(x,y):Resource_agent_location;
RSx,y:Resource_agent_saved_location; RR:Required_resource;
Loc:Location; RSx,y[]:Array_of_Previous_locations;
Program:
While Ra(x,y)< max (ϕx,y) Do
 For (i=0, i<= length(RSx,y[]), i++)
  If RSx,y[i]!= NULL && EoF =! True;
    Set Ra(x,y) ← RSx,y[i];
    Move_to_(RSx,y);
     If Pa(x,y) == Ra(x,y)
      Execute_plan;
      Set RSx,y[i] ← RSx,y;
      Set i ← 0
      Repeat;
     Else
     Repeat;
    End if;
  End if;
 End For
Random(i);
Random(j);
Set Ra(x,y) ← Rx+i,y+j
 If Execute_plan == True;
  Set RSx,y[i] ← Rx+i,y+j;
 End if
End While;
End.
```

Figure 4: Algorithm for product discovery using foraging strategy

Resource agent R_a visits locations stored in memory `RSx,y[]` if it is not empty. These locations are previous places where production plans were previously executed. If they do not exist, it creates these locations by first moving randomly and if by chance a plan is executed for a product agent P_a, then such location is stored in the memory `RSx,y[i]` until the End of File is reached.

If while attempting to visit these stored locations, it reaches end of file and no production plan is executed, then it generates a random number i, j that is added to the present x and y coordinates of R_a respectively. It sets its next location to $R_{x+i,y+j}$ and advances towards this location. This process is repeated until a plan is executed. If a plan is executed this way, the oldest location in memory is replaced with the new location. The resource agent R_a defaults back to visiting stored location until End of file is reached before attempting another random movement.

IV. EXPERIMENTAL EVALUATION

The S-RMS is implemented using agent based simulation software (Netlogo) [16]. Simulation experiment is designed to investigate how changes in product mix, volume and resource unavailability (machine failure) during production process impact the system and its adaptive-behavior.

To evaluate the above, six settings were used. In each setting, maximum numbers of orders that can undergo production process concurrently are kept constant at 50, 100, 150, 200, 250 and 300 respectively. Total number of available product mix is kept constant at 24, and the probability p of a product mix being selected for production is also kept constant for all product mix, $p = \frac{1}{24}$. 30 simulation runs were performed for each of the six settings for 50,000 simulation steps each. At 20,000 simulation steps, a resource failure was introduced and brought back on at 30,000 simulation steps, totaling a period of 10,000 simulation steps. The simulation is allowed to run for an extra 20,000 steps after the resource is brought back. This gives enough time for the system to re-converge.

The following parameters were measured in each of the six settings to investigate how the system adapts to changes in product mix, volume and machine failure during production.

I. **Average lead-time per unit**: This is the average time it takes to manufacture a product, i.e., the average time a product spent in the production system

II. **Production rate:** This is the number of product produced per 1,000 simulation steps during the simulation (a total of 50,000 simulation steps).

III. **Stability:** This is a measure of the system's stability with respect to production input and output. The average distance moved by all resources is measured to quantify stability.

Average lead-time per unit during the simulation is observed to be initially high in all the six settings at approximately 500 simulation steps (see Figure 5(a)). When products and resources start to interact, average lead-time per unit decreases gradually, a sign of system's convergence

 ISBN: 978-1-61208-610-1

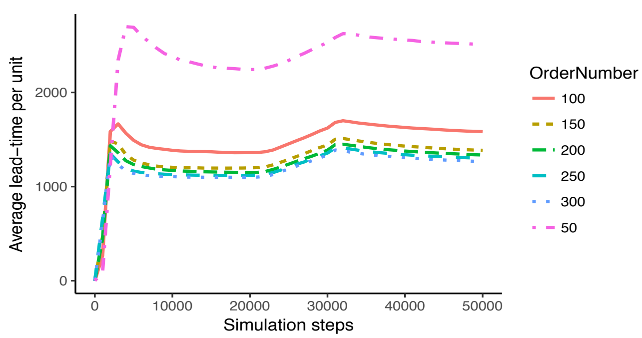
achieved through the use of pheromone as coordinating mechanism. At 20,000 simulation steps, when resource failure was introduced, lead-time per unit is observed to increase. This is the resultant effect of a machine failure. However, the system re-configures in order to adjust its processes to compensate for the failed resource, which is the rationale behind the continual production process observed in the system, though at a higher lead-time per unit compare to when all resources were present.

Average lead-time per unit is observed to decrease gradually at 30,000 simulation steps when the failed resource was re-introduced into the system. This is as a result of the system's ability to self-reconfigure its resources to integrate the new resource and immediately load-balanced production task. However, the rate of decrease in the lead-time per unit after re-introduction of failed resource (at 30,000 simulation steps) is slower compare to rate of increase when resource failure was initially introduced (at 20,000 simulation steps). This is because the coordination of the process for re-integrating failed resource is achieved through pheromones and thus takes time. On the other hand, introduction of resource failure immediately renders all information embedded in pheromones leading to the failed resource out-dated. Thus, the negative impact immediately propagates through the system, resulting in faster increase in lead-time per unit.

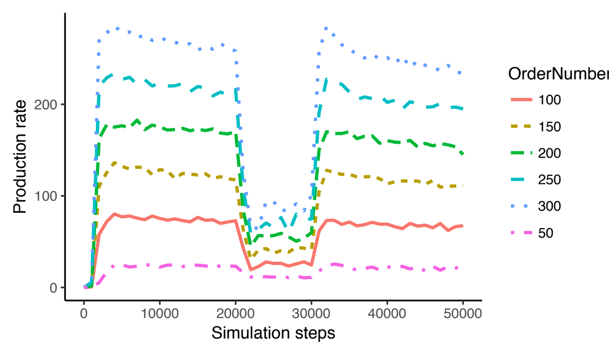
The setting with 50 maximum orders is observed to have the highest average lead-time per unit as a result of fewer product-agents in the system. Product-agents coordinate using pheromones produced by other product-agents. Therefore, the more product-agents that are available in the production environment, the higher the pheromone distribution and intensity, and the more effective the coordination mechanism.

Production rate is observed to increase from zero at about 800 simulation steps into the simulation when the system is observed to move from state of disorder to order (see Figure 5(b)). Production rate decreases between 20,000 and 30,000 simulation steps as a result of resource failure that lead to cascade of changes in the system. Production picks up soon after the failed resource was re-introduced at 30,000 simulation steps. This is because at this point, the system starts to self-reconfigure to accommodate the re-introduced resource and load-balance production task.

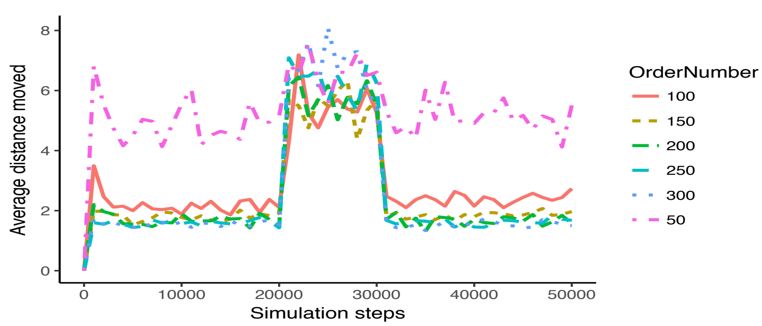
Stability of the system is quantified by average distance moved by all resource-agents. The resource-agents are observed to move around more frequently between 20,000 and 30,000 simulation steps when resource failure was introduced. This is to compensate for shortage of resources by moving more frequently to serve more products (see Figure 5(c)). At 30,000 simulation steps, when the failed resource is brought back on, the system is observed to return to its optimal state. This is as a result of the ability of the system to integrate the new resource and immediately load-balanced production task.



(a): How average lead-time per unit varies during production and resource failure



(b): How production rate varies during production and resource failure



(c): How average distance moved by resource-agent varies during production and resource failure

Figure 5: S-RMS experimental output

 ISBN: 978-1-61208-610-1

V. DISCUSSION AND CONCLUSION

Unexpected changes in the production environment, such as constant changes in product mix, volume and machine failure generate cascades of events that disrupt the system's behavior during production. Such disruptions are recorded in the three observations, which are average lead-time per unit, production rate and stability. The system is observed to be able to keep-up production irrespective of product volume and constantly changing product mix. However, introduction of machine failure disrupts the system, but does not halt production. This shows the adaptive property and self-reconfigurable capability of the proposed S-RMS.

The average lead-time per unit is observed to increase and decrease during and after disruption; this implies that the remaining resources were able to share the task of the failed resource (machine) without halting production. The production rate also decreased and increased during and after disruption instead of production coming to a stop. This implies that the system is capable of re-configuring immediately to compensate for the shortage of resources in the system. The average distance moved by all resource-agents is observed to increase during disruption. This is as a result of three resources executing production task of four resources, therefore requiring resource-agents to cover more distances on the shop floor.

Throughout the simulation, there was no observable instance where production rate equaled zero or resource-agents and product-agents lost coordination, even during unexpected changes in the production environment - like constant changes in product mix and machine failure. This is unlike a typical manufacturing system with reconfigurable capability, where production has to come to a halt for reconfiguration task to be carried out. The result obtained from the demonstration of S-RMS in silico shows that immediate self-reconfiguration of manufacturing system is possible without stopping production process using nature inspired approach. This proven concept will engender a new thinking in the design and implementation of production system with the capability for personalized mass customisation.

A future work for this research will be to compare S-RMS with other implementation of RMS without mobile products and resources, to evaluate efficiency and throughput gain due to product mobility, resource mobility, and self-reconfiguration. Also, the use of nature inspired coordination mechanisms are well known for slow convergence which may impact system performance, hence the use of machine learning algorithm for coordination in S-RMS may be more suitable. Therefore, a comparison of these two approaches based on efficiency and throughput will be explored in future research.

REFERENCES

[1] B. Pine II, Joseph (1993). *Masse Customization - The New Frontier in Business Competition*. Harvard Business School Press. ISBN 0-87584-372-7

[2] S. J. Hu, "Evolving paradigms of manufacturing: From mass production to mass customization and personalization," *Procedia CIRP*, vol. 7, pp. 3–8, 2013.

[3] A. Brusaferri, A. Ballarino, and E. Carpanzano, "Distributed intelligent automation solutions for self-adaptive manufacturing plants," *IFIP Adv. Inf. Commun. Technol.*, vol. 322 AICT, pp. 205–213, 2010.

[4] D. Sanderson, J. C. Chaplin, L. De Silva, P. Holmes, and S. Ratchev, "Smart manufacturing and reconfigurable technologies: Towards an integrated environment for evolvable assembly systems," *Proc. - IEEE 1st Int. Work. Found. Appl. Self-Systems, FAS-W 2016*, pp. 263–264, 2016.

[5] S. Benjaafar, S. Heragu, and S. A. Irani, "Next Generation Challenges Layouts : Factory and Recent Progress," *Interfaces (Providence).*, vol. 32, no. 6, pp. 58–76, 2002.

[6] H. Sarper and T. J. Greene. "Comparison of equivalent pure cellular and functional environments simulation," *International journal of computing, Integrated Manufacturing*., vol.6, no 4, pp 221-236, 1993

[7] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Brussel, "Reconfigurable manufacturing systems," *Annals of the CIRP*, vol. 48, pp.527-540, 1999.

[8] A. Koestler, The Ghost in the Machine, Arkana Books, London, 1989.

[9] A. R. Sadik and B. Urban, "A Novel Implementation Approach for Resource Holons in Reconfigurable Product Manufacturing Cell," *Proc. 13th Int. Conf. Informatics Control. Autom. Robot.*, vol. 1, no. Icinco, pp. 130–139, 2016.

[10] M. Onori, N. Lohse, J. Barata, and C. Hanisch, "The IDEAS Project: Plug & Produce at Shop-floor Level," Assembly Automation, vol. 32, no. 2, pp. 124–134, 2012.

[11] M. Onori, D. Semere, and B. Lindberg, "Evolvable Systems: An Approach to Self-X Production," International Journal of Computer Integrated Manufacturing, vol. 24, no. 5, pp. 506–516, 2011.

[12] A. Omicini, "Nature-Inspired Coordination Models: Current Status and Future Trends," ISRN Software Engineering, 2013, pp.1–13.

[13] A. Ricci, A. Omicini, M. Viroli, L. Gardelli, and E. Oliva, "Cognitive stigmergy: Towards a framework based on agents and artifacts," In Weyns, D., Parunak, H. V. D., and Michel, F., editors, Environments for MultiAgent Systems III, volume 4389 of LNAI, pages 124{140. Springer. 3rd International Workshop (E4MAS 2006).

[14] F. Heylighen, "The Science of Self-Organization and Adaptivity," The Encyclopedia of Life Support Systems, vol 5, no 3, pp. 253–280, 2001.

[15] P. Leitão, J, Barbosa, and D. Trentesaux, "Bio-inspired multi-agent systems for reconfigurable manufacturing systems," Engineering Applications of Artificial Intelligence, vol 25, no 5, pp. 934–944, 2012.

[16] U. Wilensky, Center for connected learning and computer-based modeling, Northwestern University, Evanston, IL, 1999, http://ccl.northwestern.edu/ netlogo/ (accessed 08.01.18).