Journal of Ambient Intelligence and Smart Environments 13 (2021) 157–180
DOI 10.3233/AIS-210593
IOS Press

Towards trustworthy Cyber-physical Production Systems: A dynamic agent accountability approach

Richárd Beregi [a], Gianfranco Pedone [a,*] and Davy Preuveneers [b]
[a] *Centre of Excellence in Production Informatics and Control, Institute for Computer Science and Control, Eötvös Loránd Research Network, Kende u. 13–17, 1111 Budapest, Hungary*
E-mails: richard.beregi@sztaki.hu, gianfranco.pedone@sztaki.hu
[b] *imec-DistriNet, KU Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium*
E-mail: davy.preuveneers@cs.kuleuven.be

Abstract. Smart manufacturing is a challenging trend being fostered by the Industry 4.0 paradigm. In this scenario Multi-Agent Systems (MAS) are particularly elected for modeling such types of intelligent, decentralised processes, thanks to their autonomy in pursuing collective and cooperative goals. From a human perspective, however, increasing the confidence in trustworthiness of MAS based Cyber-physical Production Systems (CPPS) remains a significant challenge. Manufacturing services must comply with strong requirements in terms of reliability, robustness and latency, and solution providers are expected to ensure that agents will operate within certain boundaries of the production, and mitigate unattended behaviours during the execution of manufacturing activities. To address this concern, a *Manufacturing Agent Accountability Framework* is proposed, a dynamic authorization framework that defines and enforces boundaries in which agents are freely permitted to exploit their intelligence to reach individual and collective objectives. The expected behaviour of agents is to adhere to CPPS workflows which implicitly define acceptable regions of behaviours and production feasibility. Core contributions of the proposed framework are: a manufacturing accountability model, the representation of the *Leaf Diagrams* for the governance of agent behavioural autonomy, and an ontology of declarative policies for the identification and avoidance of ill-intentioned behaviours in the execution of CPPS services. We outline the application of this enhanced trustworthiness framework to an agent-based manufacturing use-case for the production of a variety of hand tools.

Keywords: Agent accountability, agent behavioural confidence, security policies, Cyber-physical Production System, smart manufacturing

1. Introduction

The rise of Industry 4.0 (I4.0) [33] and the convergence of the digital and physical worlds are increasingly, if not radically, transforming production requirements and corresponding strategies. I4.0 is all about information and data exchange: between people, machines, materials and systems. The exploitation of the I4.0 paradigm requires the availability of adequate information across all engineering and production value chains, which is the result of aggregation and fusion of data from various heterogeneous sources, often under real-time conditions. This represents

*Corresponding author. E-mail: gianfranco.pedone@sztaki.hu.

a challenge for the provision of an efficient, secure and reliable information management infrastructure. There is a strong demand for novel manufacturing architectures in order to address new emerging industrial requirements for globally interconnected Cyber-physical Production Systems (CPPS).

According to [28], CPPS consist of autonomous and cooperative elements and subsystems that are inter-connected on the basis of a mutual context, within and across all levels of production, from processes through machines up to production and logistics networks. Seeking for analogous perspectives among computational paradigms, past decades have already elected Multi-Agent Systems (MAS) as an alternative way to design and develop intelligent decentralised control systems [26] based on autonomous and cooperative entities (agents), which exhibit characteristics of modularity, robustness, flexibility, adaptability and reconfigurability [18]. From a conceptual point of view, MAS based architectures consist of solving problems between agents and control mechanisms, and represent a natural candidate for realizing also CPPS [43]. Agencies of agents can act as a collaborating society that aims at solving complex problems [23]. The coordination of MAS comprises those methods for obtaining an efficient and consistent behaviour at system level, through adequate planning techniques. This can be described as an approach to coordination, where the early design of action sequences oriented to a global purpose must be taken into account in peer-to-peer agent interactions [8]. Flexibility can be exploited as a basis for future intelligent and automated production systems as well, which are able to control the entire production chain [12] and in which machinery can be viewed, in a simplified way, as a collection of workstations whose role is well established from the outset and the authorization level upon organizational assets. Each station is equipped with component shops and stations are seen in this work as agents with eventually autonomous functions and goals. Methods based on multi-agent programming can be applied between flexible manufacturing processes and cooperation with agents, leading to the development of strategies to control and optimise production planning [31], often obtained by simulations [37].

In [21], the authors offer a review of the development and use of multi-agent modeling techniques and simulations in the context of manufacturing systems and Supply Chain Management (SCM). They also identify and evaluate some key issues involved in using MAS methods to model and simulate manufacturing systems, such as modeling competitive manufacturing systems, capturing the evolutionary development of manufacturing systems, qualitative analysis emerging from quantitative methods and developing a real-time-based simulation framework and real-life applications.

One of the most exciting and crucial properties of an agent is conceptualised in terms of "delegation", granted for the execution of its tasks. The essential component of delegation is trust [25] and, in this sense, agents are, next to the possibility they have to exploit their freedom in pursuing "personal" objectives, mandatorily expected to achieve the goals defined in the production work-flows.

1.1. Motivation

Customer satisfaction is one of the factors contributing to the success or failure of a business. From a general point of view, it derives directly from the requirements explicitly defined by the customer, but the creation of a supplementary added-value in the end-product (or service) can further increase it [2]. It is hard to quantify the effective ratio between identified and unidentified needs which together determine the overall customer satisfaction. An example could be long life products in which higher profits are driven by unique extras and luxury components. Imagine for instance modern cars with all the optional extras, hard to plan for every future situation, especially if one has never heard of those options before. There are also circumstances in which customers have only an abstract idea of their own requirements. In this sort of "elastic" scenarios we aim to show that the adoption of agent-based technology, in which the accountability of agents' action is constantly re-evaluated, can lead to a higher level of customers' satisfaction.

Nevertheless, reports on industrial deployment of agent-based systems indicate that, despite the strong academic and industrial involvement in this field, the full potential of agent technology has not been fully utilised, yet, and that not all of the developed agent-oriented concepts and techniques have been completely exploited in industrial practice [44,45]. This is mainly due (next to other key factors like design, technology, standardization, hardware, application and cost) to the insufficient level of "trust" in the full exploitation of the paradigm, as security and safety in manufacturing require that communication and interactions among the agents be secure and trustworthy [20]. Industry is in general "concerned" about risks related to MAS behaviours emerging without any central unit, and

there are no formal algorithms or procedures guaranteeing that the distributed systems would behave as desired. The usual way to verify this is to apply simulation, but it is infeasible to simulate all possible, emerging configurations.

1.2. Contribution

With centralised control and in the absence of a thoughtful survey, unidentified requirements of the customer can remain unsatisfied [6]. Keeping in mind all of the resource and technological constraints while giving more freedom to the automatised shop-floor in adding extra, not ordered values (like features or even processes) can still keep the profit within acceptable margins but generate a considerable increase in the customer satisfaction. Giving too much freedom, on the contrary, can undermine production goals.

To this end we propose the Manufacturing Agent Accountability Framework (MAAF), whose fundamental elements can be listed as follows: (i) an agent accountability model, (ii) a dynamic authorization-based mechanism to convey agent accountability into actionable decisions by agents, and (iii) the Leaf Diagram (LD) for the representation and evaluation of deviations in agents behaviours from the expected ones in the manufacturing process. MAAF merges concepts of security, feasibility and trust into a single correlated dimension, which orchestrate limitations in agent behaviours at each production step and regulate them by means of a policy-based authorization middleware, whose rules are easily modifiable by the manufacturer.

MAAF is intended to represent a possible compromise between the rigidity of production expectations and the elasticity necessary for the exploitation of agent "liberty" in pursuing a higher level of intelligent collaborations. Trust is considered essential to make agent interactions effective, and the lack thereof, is a key challenge addressed in this work. MAAF introduces a dynamic accountability based approach, in which the behavioural model of collaborating agents is identified by agent goals' trajectories and measured in terms of acceptable behavioural "distance" within so called *Confidence Region* (CR). It represents the space of additional resources and technology tolerance combination within which an agent can freely act according to its internal goals but pursuit, at the same time, its manufacturing commitments. Access to resources will be linked to the authorization profiles by dynamically catalyzing the effects of behaviours emerged within the CR into accountability auditing for the agent.

1.3. Paper outline

The rest of the manuscript is organised as follows: Section 2 investigates the existence of related works with respect to trustworthy distributed control and security. In Section 3 the MAAF is presented, highlighting the targeted manufacturing scenario and its requirements, and introducing the LD. In Section 4 outcomes of the MAAF realization and application to the use-case of interest are reported. The agent interactions, the policy-based simulation, the experimental setup are also discussed. Conclusions and future work close the paper.

2. Related works and background

In this section we investigate the existence of similar research works in literature, mainly in relation to control mechanisms in distributed manufacturing architectures (included MAS) and trust-oriented models, access controls and policy based authorization.

2.1. Distributed control and agents

As mentioned before, distributed systems with an appropriate supervisory system can improve their overall performance over rigid hierarchical systems. A thorough overview of cooperative controls enumerates possibilities for production and logistics environments, addressing multiple shortcoming [29]. Technologies and applications presented in [30] emphasise a relatively low industrial acceptance because of the risk of consistent global operations and the difficulties deriving from legacy system integrations. In accordance with requirements from the production domain, agent-based manufacturing seems to be one of the most effective theories and modeling approaches for distributed production control [41]. Agents are defined in a manufacturing environment according to their ability to

autonomously select appropriate settings and find their own strategies towards the production goals. The strength of MAS based organizations (societies) is that they enable the construction of very complex systems that are nonetheless efficient in the use of resources, highly resilient to disturbance (both internal and external), and adaptable to changes in the environment in which they exist [41]. Within a society, agents may dynamically create and change hierarchies although in agent-based manufacturing models four basic types of agents can initially be found: *resource*, *order*, *product* and *staff agent*, each one representing an important aspect of production processes [42].

Implementing architectures that analyse the behaviour of agents and, according to this analysis, deliver information can become an important component in decision making process [15]. MAS approach is suitable to support the current requirements for modern control systems in industrial domains, providing flexibility, robustness, scalability, adaptability, reconfigurability and productivity [21,38]. Works on the dynamics of industrial systems and supply chains have attempted to describe the networks of relationship that characterise contemporary businesses' trading situations and internal functional structures [3,27]. Despite this apparent growth in interest in MAS from the manufacturing systems research community, it is clear that research has not yet been able to fully exploit the potential of MAS in the past decades. This is partially because of the difficulties in increasing the level of trust perceived in industry when dealing with autonomous intelligence, one of the major roadblocker for the current weak adoption of this technology in industry.

Trust management has been approached from two different perspectives: policy based trust and reputation based trust [5,11]. The policy based approach tries to describe trust using predefined rules while relying on strong cryptographic assurances found in security certificates. Reputation based approaches establish trust based on direct experiences, which might be shared through a community. Many reputation based trust evaluation models have been proposed and implemented in different areas [17,36,39] but most of them focused on the application of algorithms for trusters to model the trustworthiness of trustees in order to make effective decisions about which trustees to select. For this purpose, many reputation based trust evaluation models use third party information sources such as witnesses. Also in [4], after giving a comprehensive overview of trust evaluation models, the authors propose a scalable model to locate a set of witnesses and combine a suspension technique with reinforcement learning in order to improve the model responses to dynamic changes in the system.

The approach introduced in this paper, on the contrary, leverages a policy based trust management framework that extends the MAS paradigm with the enforcement of strong boundaries in which agents can freely exploit their intelligence. This framework provides strong guarantees that reputation based models cannot offer, and we will present and evaluate its application in a manufacturing use-case for the production of tools (hammers). Considering its adherence to the specific application domain, the authors of this paper referred to PROSA methodology, the Reference Architecture for Holonic Manufacturing Systems [40,42], as to the starting point for building the agent accountability model.

2.2. Trust related concepts

Concepts like responsibility and reliability in manufacturing execution environments [19] have evolved along the decades from a traditional human-oriented approach [1] to a more distributed, MAS-based society [7]. Reliability of production processes is a key issue that ensures the stability of production system operation, as it improves product quality and reduces production losses. But while reliability tries to predict the fault analysis of production process trough the provision of corrective actions for the elimination of critical faults in machinery, accountability is used as a mechanism to measure the effects of freedom granted to each single agent in making alternative decisions. In this sense accountability will be utilised to characterise MAS based CPPS systems, without the need to create explicit relationships among the involved agents [7].

2.3. Trust models, access control and policy-based authorization

Policy based trust models allow to define what evidence should be presented by an entity, in what circumstances, in order to establish a trust relationship. Blaze et al. [9] first defined the concept of *trust management* as *a unified approach to specifying and interpreting security policies, credentials, and relationships which allow direct authorization of security-critical actions*. They argued that the question answered by a trust compliance checker is not *'Is*

the key that signed this request authorised to take this action?', but rather *'Does the set* C *of credentials prove that the request* r *complies with the local security policy* P*?'*. Furthermore this question should not be answered by the application itself, but by a general trust compliance checker. In the following years several implementations of trust management frameworks as defined by Blaze et al. followed, such as *Keynote* [10], *REFEREE* [14] and *RT* [22].

Cheminod et al. [13] motivated why access control policies are fundamental in the process of securing networked industrial control systems and critical infrastructures. Liu et al. [24] raised similar security concerns particularly for collaborative manufacturing systems, and argued that configuring and enforcing an access control model in a collaborative manufacturing system is a challenging task. This concern was addressed in [34,35] with a framework that enables policy-based access control to maintain trust relationships that cross the boundaries of collaborating organizations, while lifting the burden to manually align diverse sets of authorization policies of different organizations.

In this work, we go beyond the state-of-the-art by extending such a framework to support dynamic accountability for smart collaborating agents, not only to enforce trust boundaries in which agents can operate freely under certain conditions, but also to enable accountability of misbehaving agents.

2.4. Production routing use-case

The realization of this work will be demonstrated starting from an easily understandable case of hand tool production (a hammer in the specific) and then proceed by extending results to the general use-case implemented via simulation. The activities in the production routing depicted on Fig. 1 will be the base for validating the evolution of the manufacturing execution task through the MAAF.

3. Manufacturing Agent Accountability Framework

The approach proposed in this paper is centered around the Manufacturing Agent Accountability Framework (MAAF), whose core elements are summarised as follows:

- an agent-based *Factory Agent Accountability Model*, aiming at addressing the emerging of unpredictable demand changes in production scenarios;
- the *Leaf Diagram*, which encompasses both a visualization instrument for the representation of an agent behavioural deviance, and provides an evaluation mechanism for both an agent's behavioural autonomy and the system dependency (the extent to which each agent's goal influences and limits the freedom of the overall system), based on the agent authorization profiling; and
- an *Authorization Middleware*, whose goal is to bridge the conceptualization of agent authorization policies, by merging security, feasibility and trust into a single dimension.

Assumption: it is important to emphasise that the focus of this framework is drawn only onto the production control level of manufacturing. Production planning and scheduling related aspects are implicitly taken into consideration in the model but assumed to be managed in a centralised way, so to ensure the sustainability of the overall manufacturing process in the company. The distribution and the policy-based control of such levels will demand further research, which is out-of-scope in this paper.

3.1. On the MAAF accountability mechanism

The concept of *accountability* proposed in the paper is aligned with the following consideration: effects deriving from an agent delegation (i.e. its freedom) in taking alternative decisions are catalysed through the implicit concept of responsibility. It is furthermore clear that this concept, for actions in CPPS, needs to be adapted as, in general, there is no human twinned with the agent, to whom the effects of responsibility could be propagated. Instead we linked them to the concept of reliability, as the capacity of an agent not only to do what it is expected to do, but also to deviate from the strict production workflows and yet being able to improve manufacturing end-results through the exploration of alternatives. Accountability aims at being a simplified aggregation of complex and dynamic modulation of agents' behaviour, also through the ontological policies. Accountability will act, in the end, as a sort of reward-enabling mechanism in the exploitation of their autonomy.

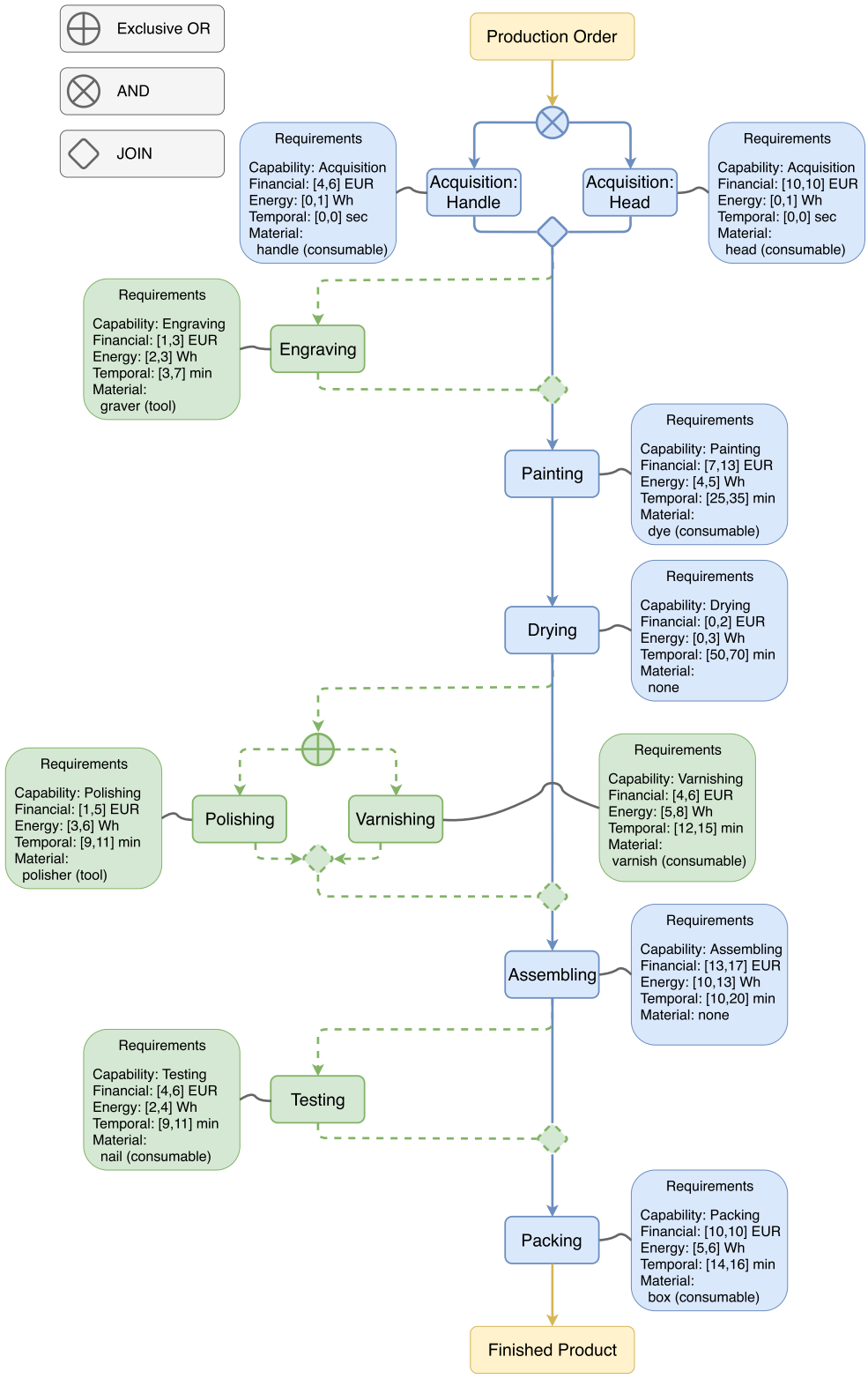


Fig. 1. Hand tool production routing (hammer).

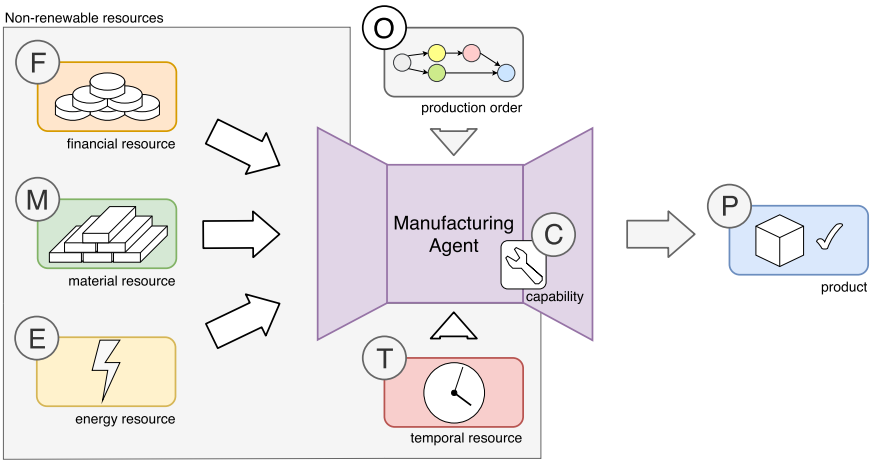


Fig. 2. Realization of a product with the utilization of non-renewable resources.

3.2. *Factory Agent Accountability Model*

In manufacturing theory, production is basically the transformation of multiple resources into products according to demands from customers. Manufacturing processes take time, consume energy, cost money and have commodity and equipment needs, which can be simplified, in generic terms, into *temporal, energy, financial* and *material resources* (Fig. 2 reports a schematic view of production from the point of view of such resources). This list of four non-renewable resource examples can be further extended according to the specific manufacturing processes of the factory, but it is hereby sufficient for the general understanding of the model. In this sense, also machinery facility can be seen as a "special" resource, thanks to its capability to combine and utilise the previously mentioned cumulative resources for the specified production purposes.

The manufacturing tasks contained in the routing (know-how derived from the production order; Fig. 1) are then executed by manufacturing agents with the utilization of their own resources and capabilities. Generally, production orders also have different levels of priority: they can be resource-based, with the goal to minimise costs for instance, or target the addition of completely new features or functionalities to the products based on the customers' and facility's requirements.

The main objective of the system is to direct the facility towards the achievement of the primary goal: a sustainable production with maximised customer satisfaction. As introduced earlier, distributed systems with the appropriate supervisory mechanisms can improve performance over rigid hierarchical systems. In accordance with this vision and taking into account requirements from the manufacturing domain, the *Factory Agent Accountability Model* has been created (Fig. 3).

Agents represent the manufacturing staff and have the knowledge to manage and regulate parts or aspects of the facility via their own methods and the resource assignments. Namingly: *Resource Manager*, *Production Planner*, *Shop-floor Manager* and *Manufacturing Agents*.

A *Customer* represents the demand of a product with both specified (explicit) and hidden (implicit) properties. The undefined demand can be a feature of the product or a priority for the *Customer*, and usually measures his attitude against the demand (quickness, cheapness or high quality). These unknown properties make the simulation model realistic and diverse. The main goal in the model is the maximization of customer satisfaction, which is determined by the *Shop-floor Manager* agent by receiving the *Customers*' feedbacks on the realised product, after the completion of production. The *Shop-floor Manager* can discover undefined demands (or requirements) from the feedback and match them with historical information of production (logs and reports) in order to develop the accountability profile for each *Manufacturing Agent*. Aim of the *Production Planner* is to create a production order, which will be an assignment for the *Manufacturing Agents* of the specified product, based on the requirements identified in *Customer*'s demand. The order contains the actual production routing (workflow). Every product relates to a production routing and might eventually permit (from the point of view of the facility strategy) acceptable alterna-

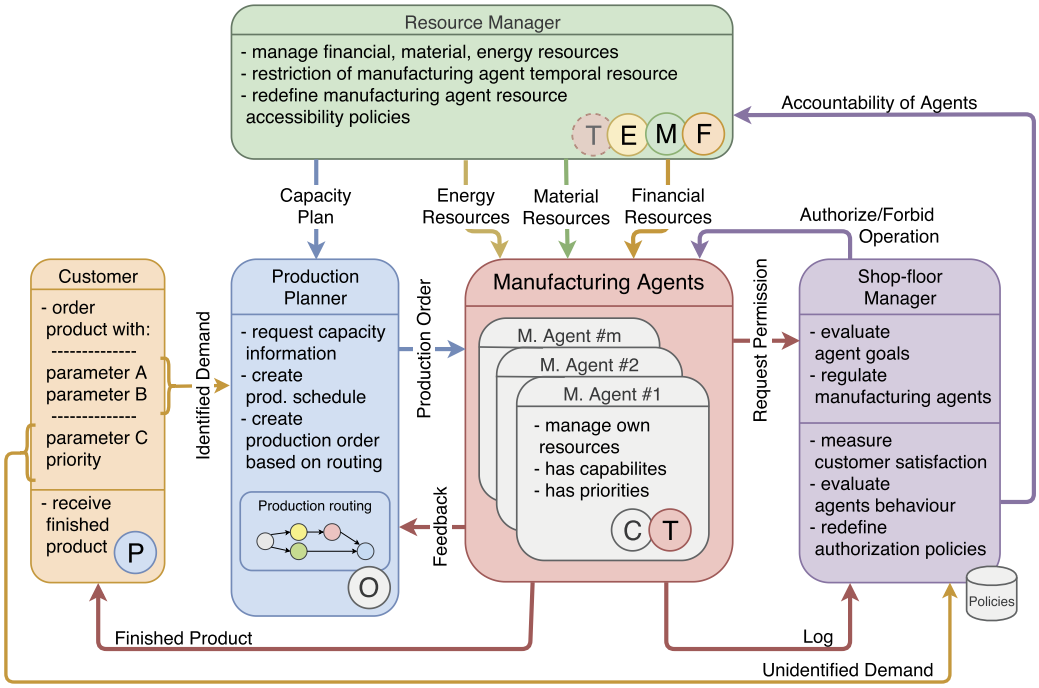


Fig. 3. Factory Agent Accountability Model.

tives and extra steps (as graphically depicted in Fig. 1). Between two mandatory tasks (blue nodes) there is always a constraining relation defined (direct, substitutional or order-independent precedence), with the possibility to add alternative operations (green nodes). Every task has a catalogue which defines the necessary resources involved (financial, energy, material and temporal), as well as the capability required from a machine resource to execute it.

The *Production Planner* is in charge to forward and employ orders via the *Manufacturing Agents* in a profitable and terminable manner, so to keep production sustainable in the long run. To achieve this purpose, and in order to plan production schedules, it has to take into considerations also directives and resource capacity plans provided by the *Resource Manager*. The latter has multiple regulatory options, like restricting and/or indirectly influencing the availability of resources based on its stock level and recommendation shaped by the *Shop-floor Manager*. The actual manufacturing work is done by the *Manufacturing agents*, as they represent the resource with the necessary capabilities to fabricate the product. The *Shop-floor Manager* identifies the management of trust towards agents, based on the methodology presented in this section. It is its responsibility to permit or deny agent actions and evaluate their performance after the realization of a product. The *Manufacturing Agents* are also regulated by the *Shop-floor Manager*, which orchestrates the specific selection of machines in concurrent situations.

Production is generally followed by an evaluation phase, in which the main focus is drawn on the whole system and is based on the customer expressed satisfaction: this arises both from the production logs and from all of the requirements of the client, calculated by the *Shop-floor Manager*. Based on the evaluation results every *Manufacturing Agent* gains a new reliability (and so accountability) level, utilised by the *Resource Manager* in order to adjust decisions related to policies on the availability of resources, by leveraging also the recommendations from the *Shop-floor Manager*. The sequence diagram illustrated in Fig. 4 highlights the interactions occurring among these agents, and can be categorised in instructions, knowledge or resource transfers, and permission or resource requests.

The factory agent *accountability model* itself does not grant the expected feasibility of the production, but it aims at reaching such a goal in conjunction with behavioural constraints detailed in policies of the authorization middleware.

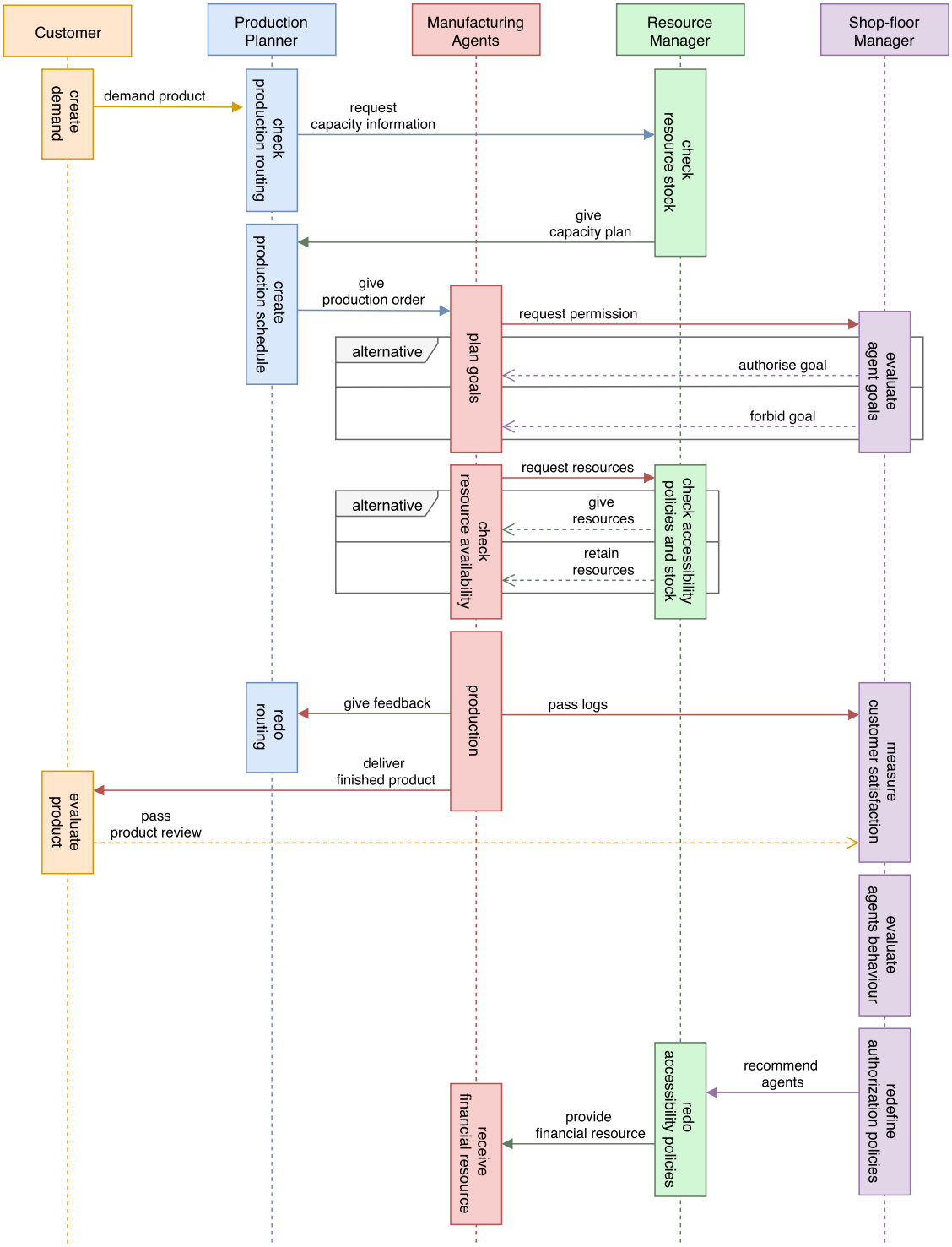
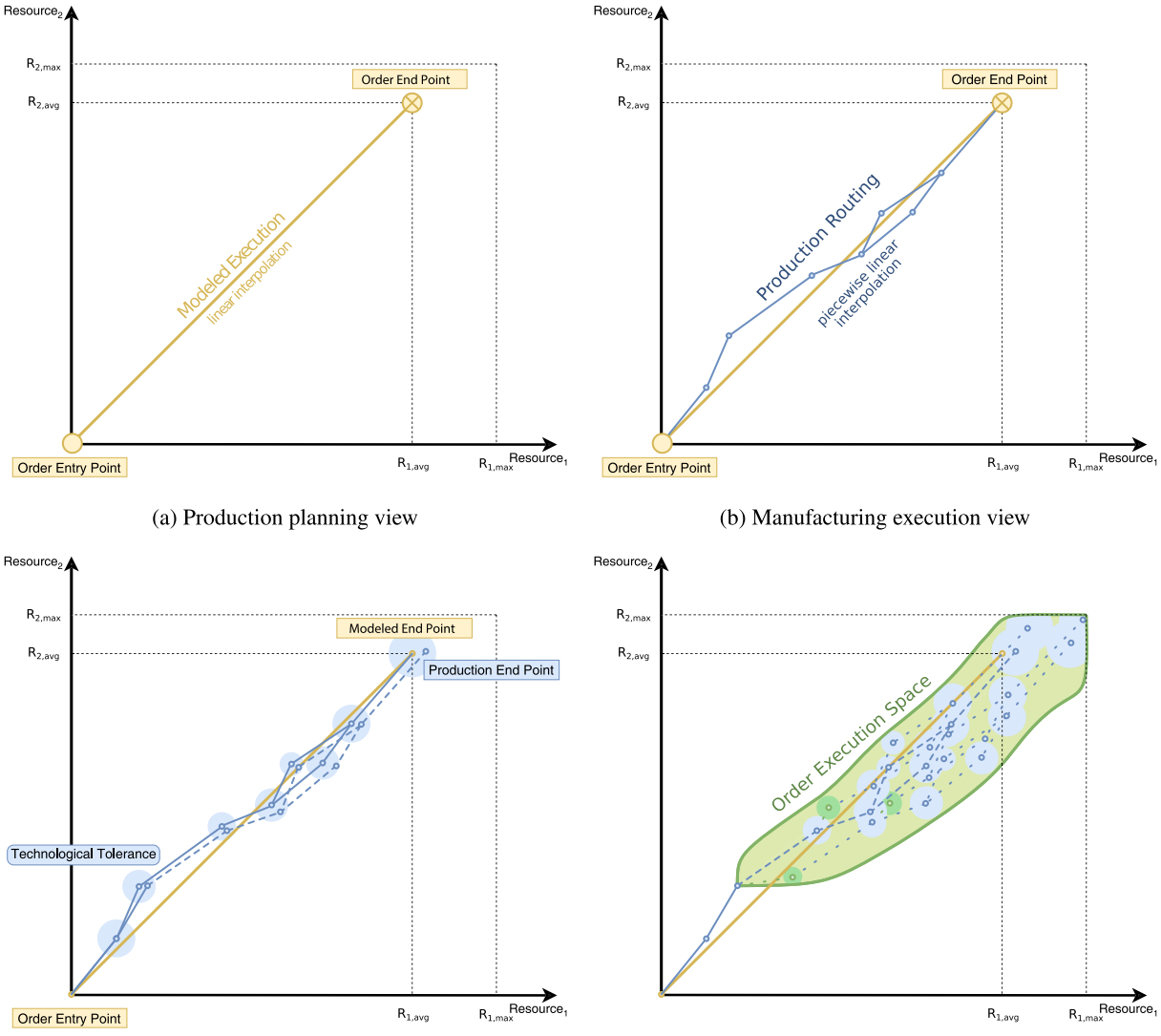


Fig. 4. Sequence of agent interactions in the studied production work-flow.

3.3. Leaf Diagram

The *Leaf Diagram* (LD) is a representation, visualization and thus monitoring tool of the MAAF for capturing and governing an agent's behavioural tendency to exploit its autonomy in pursuing its goals while converging towards the manufacturing objectives of the factory. The aim of the *LD* is to visualise and evaluate the evolution of an agent's behaviour in the multidimensional space of production resources: we have selected a two-dimensional representation for an easier explanation of concepts but this can be easily extended also to an n-dimensional case. Each axis represents a cumulative resource (temporal, energy, financial, material and energy), whose utilization is typically a monotone increment along the time.

Every production process evolves along an *Order Execution Space*, between a starting and finishing point, named *Order Entry Point* and *Order End Point* (represented with crossed circles on Fig. 5(a) or 5(b)). The path connecting these two points, from a purely economical modeling point of view, tend to show a linear combination of resources (*Modeled Execution routing*). In reality, every production order has so-called technological steps, which are defined



(a) Production planning view

(b) Manufacturing execution view

(c) Actual manufacturing execution gap

(d) Order Execution Space

Fig. 5. Derivation of the Order Execution Space.

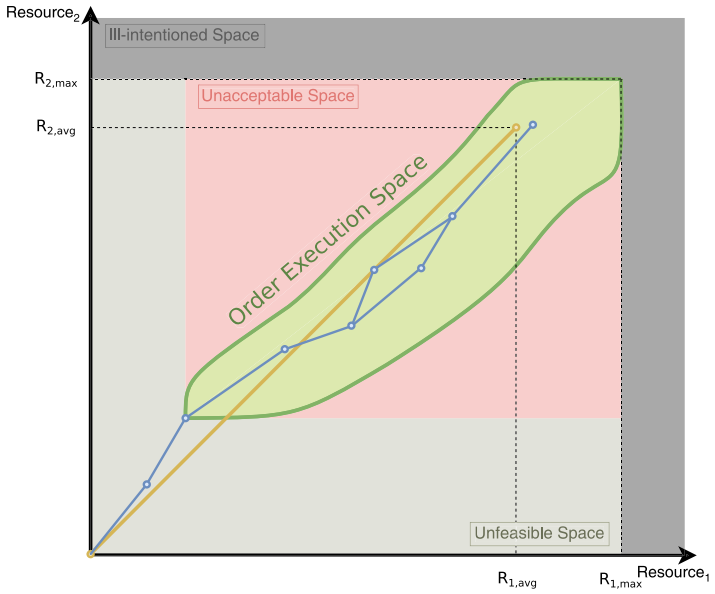


Fig. 6. Agent behavioural evaluation tool: the Leaf Diagram.

as routing points with the intention to create a vendible output. The prescribed *Production Routing* is composed of nodes representing the states of the product over the manufacturing process, while edges are the actions necessary to reach the next state (traced in blue). Edges generate trajectories, whose projection on the axis indicates the actual consumption of resources required for that specific action. It is worth highlighting how the actual execution of a manufacturing step (feasibility) usually requires a slightly different combination of production resources (manufacturing execution gap) compared to the one indicated in the prescribed route (Fig. 5(c)): this is due to the effective state of technologies and resources at that specific moment of the process (blue-filled circles on Fig. 5(c)). The union of all regions for the technology deviation generates the *Technology Tolerance* (TT). Considering that a production routing usually contains also alternative manufacturing steps (green-filled circles on Fig. 5(d)) the overall dimension of the *TT* has to be evaluated also in the light of such combinations. In the end, the *Order Execution Space* will be identified by the region of a *LD* which encompasses both the prescribed production routing and the overall space of the *TT* estimated for it.

The *LD* has been divided into four major areas characterizing the positioning of agent actions within them (Fig. 6). They have been formalised taken into consideration the global perspective of the system, as listed below:

- the *Order Execution Space* refers to all feasible manufacturing actions (green-colored region). Feasibility is intended here in terms of order realizability (e.g. none or acceptable delay, profitable production);
- complementary to the Order Execution Space are the areas of the *Unacceptable* (red colour area in Fig. 6) and *Unfeasible Space* (light grey area in Fig. 6). The Unfeasible Space refers to areas which are physically unreachable considering the cumulative nature of the involved resources. The Unacceptable Space refers to the combination of both technical and behavioural limitations for an agent, in a specific production step;
- the *Ill-intentioned Space* represents the region where malicious behavioural attempts are captured in the system (coloured in dark grey on Fig. 6): any action pointing to the *Ill-intentioned Space* is determinably against the manufacturing system goals and is punishable with the exclusion of the involved agent from any other production operation.

3.3.1. Agent CR and Accountability

Central in the LD is the definition (1) of the *Agent Accountability* (AA_m), which is here represented as a numeric value ranging from 0 (lowest accountability) to 1 (highest accountability) for each machine agent (e.g. turning,

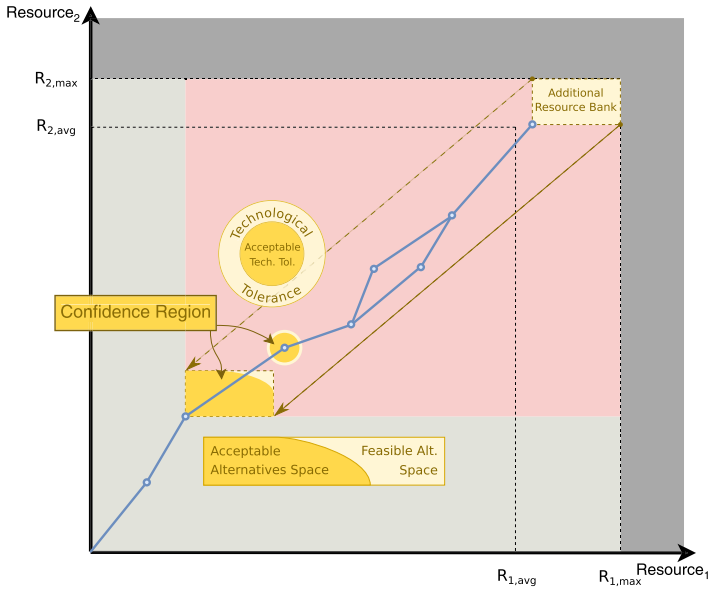


Fig. 7. *Confidence Region* of an agent at a hypothetical routing step.

milling, drilling machine, and so forth):

$$AA_m \in [0, 1], m \in \text{Machine agents} \tag{1}$$

AA_m is a value provided by the Auditor for each Machine Agent, hidden to the involved agent and recalculated at the end of a production order (see model depicted in Fig. 3). AA_m plays a fundamental role in the delimitation of an agent's CR, as detailed hereafter.

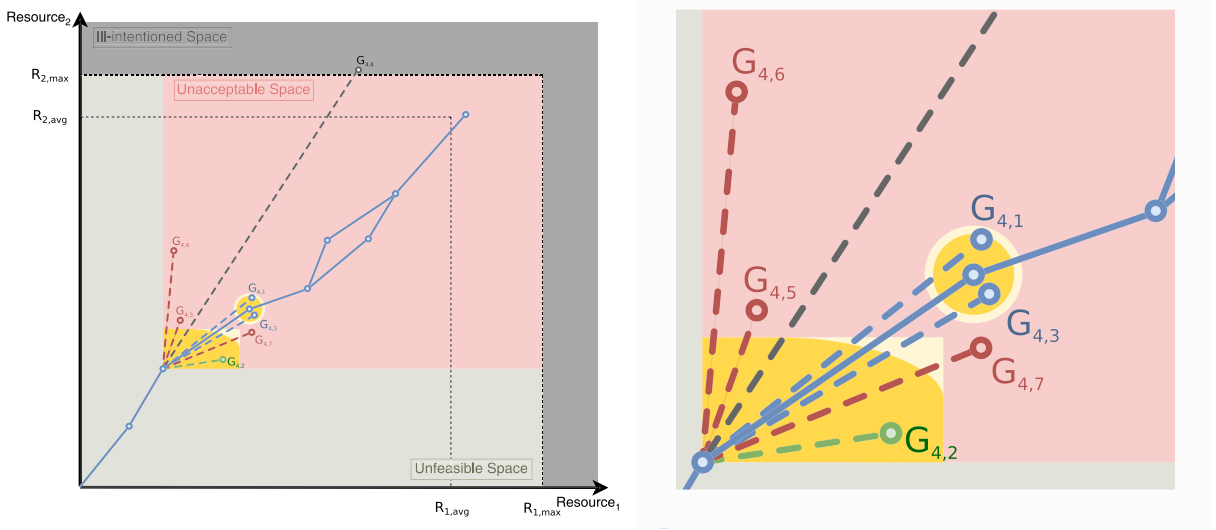
As already seen, the CR is for an agent the space of resource combinations within which it can freely exploit its autonomy and eventually decide for different alternatives in the production routing (see Fig. 7 for details). It is defined (2) as the union of two regions on the diagram: the TT (light-yellow circle) estimated for the next production step(s) and the space of *Feasible Alternatives*, leveraged taken into account the level of additional resources still available at the current production step (*Additional resource bank*). CR represents the maximum degree of freedom for an agent, in that specific step, in the selection of production resources, proportionally rescaled according to its the accountability level (AA_m). In this manner, we obtain the portion of CR in which the agent goal is also eligible for the execution of the task:

$$\text{Confidence Region}(AA_m) = \text{acceptable Alternatives} \cup \text{acceptable TT} \tag{2}$$

As $AA_m \in [0, 1]$, the result of (2) is consequently a reduction of the CR, as illustrated also on Fig. 7 (darker-yellow contours within it). The CR of an agent simultaneously conveys the concepts of production feasibility, agent autonomy and agent accountability. AA_m conveys on CR geometry the level of reliability of an agent, according to its past decisions and generated outcomes, and so by giving form to the concept of trustworthiness in the agent and its capability of taking alternative decisions. The agent's CR reported on Fig. 7 is just an illustrative example of how AA_m might be interpreted for the geometric re-definition of its boundary: any other shape would be eligible as well, provided that (i) it is contained within the maximal space of the CR and that (ii) its area is proportionally reduced by an AA_m factor.

3.3.2. Agent goals and behavioural freedom

We see agent actions as generators of behavioural trajectories, whose end-points on the diagram represent their own goals (Fig. 8). An agent *Goal* has been defined, for a given machine agent, as one of the possible combinations



(a) Example of a machine agent goals

(b) Enlarged view of an agent's goals and its *Confidence Region*

Fig. 8. Agent goals and behavioural trajectories.

of cumulative production resources (F, M, T and E on Fig. 2) targeted by the agent within the manufacturing organization:

$$G_{m,g} = (x_1, x_2, \ldots, x_r) \tag{3}$$

Where:

m as defined in (1) and $x_{1,\ldots,r}$

$$\in \textit{ Non-renewable production resources} \tag{4}$$

At each state of the production order there is the possibility to choose an alternative or extra step according to the agents goals and their relevance for the production routing. As already depicted on Fig. 6 the green-colored area contains additional possible operations while the red or dark-grey colored show the verdict for that action's outcome.

On the basis of precedent definitions, a quantitative evaluation of an agent's behavioural freedom has been introduced through the concept of *Freedom Ratio* ($FR_{r,m,g}$), as depicted in Fig. 9. Given the triplet <*resource, machine agent, agent goal*>, the $FR_{r,m,g}$ is calculated in relation to the two different portions of the agent CR (the acceptable alternatives and the acceptable technology tolerance), depending on whether the agent goal is actually (i) a feasible alternative step after the current one (p_F and green-colored circle) or (ii) the next step prescribed by the routing (p_T and blue-colored circle). These two regions are mutually exclusive, as an agent goal cannot be simultaneously an alternative and a strictly prescribed manufacturing action.

$FR_{r,m,g}$ is given by the exclusive evaluation of its components in p_F and p_T spaces:

$$FR_{r,m,g} = FR_{r,m,g}(p_F) \textbf{ or } FR_{r,m,g}(p_T) \tag{5}$$

Each component of the $FR_{r,m,g}$ is defined as the remaining resource capacity ratio along with the specific dimension, that is the difference between the maximum available in the CR and the proportion of two segments indicating

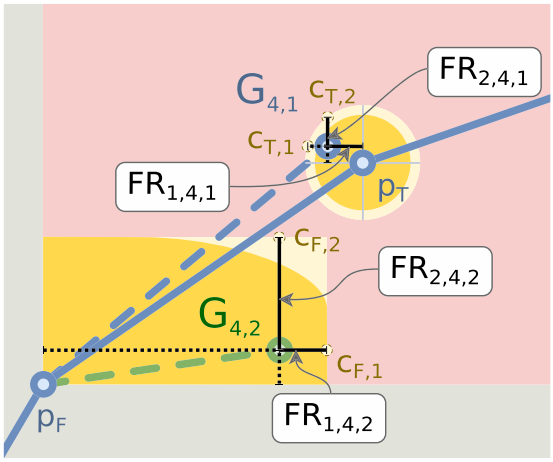


Fig. 9. Behavioural freedom ratio.

the variation of resources utilised with respect to the prescribed routing step (black segments of the CR in Fig. 9):

$$FR_{r,m,g}(p_F) = \left[1 - \frac{(G_{m,g}, p_{F,r})}{(p_{F,r}, c_{F,r})}\right] = \frac{(G_{m,g}, c_{F,r})}{(p_{F,r}, c_{F,r})} \in [0, 1] \tag{6}$$

$$FR_{r,m,g}(p_T) = \begin{cases} \frac{(G_{m,g}, c_{T,r})}{(p_{T,r}, c_{T,r})} \in [0, 1] & \text{if } r_G \geqslant r_{p_T} \\ [\frac{(G_{m,g}, c_{T,r})}{(p_{T,r}, c_{T,r})} - 1] \in [-1, 0] & \text{otherwise} \end{cases} \tag{7}$$

Where: r_G and r_{p_T} represent the amount of resource required by the agent goal $G_{m,g}$ and the prescribed routing step in the *Accepted Technology Tolerance* space, respectively; c_F and $c_T \in \partial r S_{confidence}$, p_F and $p_T \in$ Prescribed Routing, $p_F \neq c_F$ and $p_T \neq c_T$.

In practice, the estimation of $FR_{r,m,g}$ is calculated by positioning the agent goal on the straight lines parallel to the resources axes and measuring the variation of demand (compared to the one in the prescribed route step, p_F or p_T) from the boundary point of the $\partial r S_{confidence}$ for the given resources combination (c_{F_r} and c_{T_r}). A lower utilization of resources by an agent goal is considered as an increase of freedom, in the same resource dimension, for the next steps (this is a characteristic of p_T region, only).

For instance, the illustration depicted in Fig. 9 reports an example of *Behavioral Freedom Ratio* in relation to two resources (namely no. 1 and 2), for the machine agent no. 4 and its goals no. 1 and 2. According to this case and simply based on proportions evidenced on the LD, estimations would report the following values (if equal to 1 indicates that the agent performance complies with the prescribed one in both spaces of the *CR*; in p_F values cannot be negative as the agent is executing a further production step, and so demanding for additional resources):

- $FR_{1,4,1} = -0.6$ (or -60%): lower requirement of resource no. 1 compared to the prescribed routing step. The value indicates an *increase* of freedom for the next steps;
- $FR_{2,4,1} = 0.6$ (or 60%): a higher requirement of resource no. 2 compared to the prescribed routing step. The value indicates the *remaining* freedom;
- $FR_{1,4,2} \approx 0.17$ (or 17%): the value indicates the *remaining* freedom with respect to resource no. 1;
- $FR_{2,4,2} \approx 0.73$ (or 73%): the value indicates the *remaining* freedom with respect to resource no. 2.

Next to the evaluation of single agent actions, the LD also aims at providing an estimation of freedom for the entire system, or otherwise said, the level of freedom (in terms of resources) left by the agents for the achievement of remaining production objectives. This has been specified, for a given machine agent and its goals, by the *System Freedom* value ($SF_{m,g}$), which defines to what extent each agent goal (i.e. its sequence of actions) might influence

the entire manufacturing strategy. It is the difference between 1 (maximum freedom) and the average of $FR_{r,m,g}$ for a given agent:

$$SF_{m,g} = 1 - \overline{\begin{bmatrix} FR_{1,m,g} \\ \vdots \\ FR_{r,m,g} \end{bmatrix}} \tag{8}$$

Negative values indicate an increase, while positive ones produce a decrease of freedom in the orchestration of production resources. Continuing with the previous example, $SF_{m,g}$ can be calculated as the mean of all $FR_{r,m,g}$ values, as follows:

$$SF_{4,1} = 1 - \overline{\begin{bmatrix} -0.60 \\ 0.60 \end{bmatrix}} = 1 - 0.0 = 1\ (100\%) \tag{9}$$

$$SF_{4,2} = 1 - \overline{\begin{bmatrix} 0.17 \\ 0.73 \end{bmatrix}} = 1 - 0.45 = 0.55\ (55\%) \tag{10}$$

In this illustrative example values of the *System Freedom* indicate that the freedom still available in the system for the orchestration of the remaining manufacturing process after the execution of agent goals $G_{4,1}$ and $G_{4,2}$ is respectively 100% and 55% (compared to the prescribed routing, i.e. 100%). $G_{4,1}$, despite its deviation from the prescribed routing, leaves the overall system freedom unaltered. On the contrary, a general decrease of 45% in the system freedom is caused by the execution of $G_{4,2}$: this will influence the decision strategy of agents in the remaining routing path. It is worth recalling that a given order can have (or is influenced by levels of) priorities: if they are resource-based (temporal, financial and material), such as cost minimization for instance, then the framework can leverage the $FR_{r,m,g}$ as it can focus on single dimensions of interest. On the contrary, if the priority dictates the addition of new features or functionalities (which are referred to as an implicit combination of all production resources) then the proposed approach can utilise the $SF_{m,g}$ as a more effective, global performance indicator.

3.4. Dynamic authorization as a boundary in smart manufacturing

The previous section discussed the LD and the computations behind the degrees of freedom that an agent can have in selecting and completing the next task of the workflow. Note, however, that the mathematical background of the LD expresses only necessary but not sufficient constraints. For example, consider a production routing with a paint job. In such a scenario, the LD models time as any other resource. According to the LD, an agent would be allowed not to wait – i.e. do not spend any of the time resource – to achieve a higher freedom ratio. However, it is clear that an agent should wait for the paint to be dry before continuing with the next task. These additional constraints between tasks and their execution order are enforced by the authorization metamodel described in this section. It is this metamodel that provides the foundation for the dynamic accountability.

Within the frame of this manufacturing use case, the scope of authorization extends to all the actions undertaken by any entity in the system, including human operators and manufacturing agents. The authorization decisions are governed by the roles, responsibilities, permissions and other attributes associated with these entities. Our authorization middleware not only enforces authorization policies to constrain access to production information (including read, write, update and delete operations), but also evaluates access control decisions to permit or deny agents of undertaking certain manufacturing operations. The former goal addresses information security concerns which we implemented with contemporary and state-of-practice identity and access management (IAM) software solutions, such as RedHat Keycloak.[1] The latter goal of our authorization middleware aims to prohibit manufacturing agents of

[1] https://www.keycloak.org

overstepping their trust boundaries, and this is the scope of the remainder of this section. To achieve this objective, the middleware builds upon an authorization meta-model that interacts with the manufacturing meta-model.

This authorization metamodel maps with concepts and properties of the manufacturing meta-model describing workflows and production routings, such as the one depicted in Fig. 1: for each step in the production order advancement the illustration highlights the capability required to perform the step, as well as the requirements related to all of the resource types (previous Fig. 2). To simplify the mapping, both the manufacturing and authorization meta-models are formally grounded in separate ontologies, and the description logic foundation of ontologies allows us to reason about the implicit order of tasks, permissions, resource boundaries, and other process execution constraints.

The example authorization rules depicted below are merely used to explain the overall authorization concept, and are by no means representative of the more sophisticated access control policies of the manufacturing use case. The authorization policies reuse semantic concepts from other external domain-specific ontologies that, for example, define concepts to instantiate workflows. This ontology declares how a workflow can be instantiated and composed from individual tasks, whereby each of these tasks declares its own restrictions (dependencies on other tasks, mutual exclusion, etc.). These details were omitted from the following examples as the authorization policies and meta-models are semantically grounded, and the underlying OWL 2 ontology reasoner takes care of inferring all implicit relationships to semantically evaluate the authorization rules.

In the examples below, we assume a scenario where multiple agents with different capabilities can work on production orders following a particular production routing as depicted in Fig. 1. In a separate ontology we model the different agents and their capabilities, as well as the different production steps required to complete the order. Even if an agent has a certain capability, the task associated with that capability is only allowed to be carried out under certain conditions which are specified in authorization policies. The authorization policies are expressed in the Semantic Web Rule Language (SWRL) specification, such that inferences of allowed tasks can be done with rules like the following:

```
Agent(?a) ^ hasCapability(?a,?t)
    ^ ProductionRouting(?p)
    ^ hasTask(?p,?t)
    -> hasPermittedTask(?a,?t)
```

This is a simplified rule which states that an agent a with a capability to execute task t is allowed to pursue that task for a product if required according to the production routing p. Contrary to what is common in most policy based access control languages, the evaluation of this rule does not result in a single permit or deny decision for a particular task. Instead, it infers a list of all the agents and their respective capabilities that they are allowed to perform on any of the products that need to be manufactured. Indeed, traditional policy-based access control policies typically decide whether a subject is *permitted* or *denied* to execute a specific task on a resource. By leveraging the inferencing capabilities of the OWL 2 reasoner, our framework instead infers all tasks an agent is permitted to pursue. This is much more effective compared to an exhaustive search of every possible combination and checking whether each combination is permitted or denied.

Note that in the above rule we do not take any additional constraints into consideration, such as task dependencies in the production routing and the use of temporal, financial, material or energy resources. In practice, there are multiple rules to account for the fact that each task requires resources (temporal, material, financial, energy, etc.) and these constraints must be taken into consideration as well:

```
Agent(?a) ^ hasCapability(?a,?t)
    ^ ProductionRouting(?p)
    ^ hasTask(?p,?t)
    ^ demandsResource(?c, ?f)
    ^ FinancialResource(?f)
    ^ swrlb:lessThan(?f, 17)
    -> hasPermittedTask(?a,?t)
```

This second rule restricts the results of the first rule to those products, agents and tasks that require a financial resource of less than 17 units. This means that the agents responsible for production will be excluded due to the demand for a financial resource of 20 units.

Similar rules are required to model constraints that enforce dependencies, such as a task that can only be started whenever all the previous tasks are completed. The `hasTask(?p,?t)` predicate is further extended in the above rule to:

```
... ^ hasTask(?p,?t)
    ^ hasPreviousTask(?t, ?t0)
    ^ hasTaskStatus(?t0, completed)
    ^ ...
```

Furthermore, additional rules must account for alternative tasks that are mutually exclusive. They must check that all previous tasks are finished and that no alternative task (Exclusive OR) has already begun. This is illustrated for two mutually exclusive tasks `polishing` and `varnishing`.

```
ProductionRouting(?p)
    ^ hasTask(?p, polishing)
    ^ hasTaskStatus(polishing, completed)
    -> hasUnfeasibleTask(?p, varnishing)

ProductionRouting(?p)
    ^ hasTask(?p, varnishing)
    ^ hasTaskStatus(varnishing, completed)
    -> hasUnfeasibleTask(?p, polishing)
```

Additional description logic inference rules will guarantee that the permitted task set and unfeasible task set are mutually exclusive. This way, a task can not be present in both sets at the same time.

Beyond the inferencing capabilities, our authorization framework offers querying capabilities. For example, with the following simple SQWRL query [32], an agent a can request all allowed tasks t and corresponding freedom ratio r, and then decide on its own which strategy it will pursue:

```
Agent(?a) ^ hasPermittedTask(?a, ?t)
    ^ hasFreedomRatio(?t, ?r)
    -> sqwrl:select(?a, ?t, ?r)
```

In fact, the above query will return an overview of all agents and their allowed tasks, thereby representing the *CR* of each agent. The strategy of the agent will depend on the *Freedom Radio* as discussed in the previous section. Two such strategies are:

- **Highest Freedom Ratio**: Rank the tasks from the above query according to freedom ration, and select any task with the highest value.
- **Median Freedom Ratio**: Rank the tasks from the above query according to freedom ration, and select any task with a value above the median (or average) value.

The difference in the above strategies is that the latter allows more options to the agent, allowing to select a task with a lower freedom ratio which in the end may lead to a greater benefit. The full instantiation of the manufacturing metamodel at design time represents the whole production routing, from *Order Entry Point* to *Order End Point* as well as all intermediate production states, as a sequence of concurrent and alternative *Task* instances.

At runtime, our framework leverages the HermiT OWL 2 reasoner [16] – also used in Protégé – and an SWRL reasoner to process the semantic models and rules. By modelling task dependencies and leveraging the inference power

of the HermiT ontology reasoner, any implicit dependencies and constraints are automatically derived whenever an agent wants to initiate a task.

In addition to the semantic rules presented in Section 3.4, accountability-specific rules were used to infer tasks that would lead to ending up in the *Unacceptable Space*, the *Ill-intentioned Space* and the *Unfeasible Space*:

```
Agent(?a) ^ hasUnacceptableTask(?a, ?t)
    -> sqwrl:select(?a, ?t)

Agent(?a) ^ hasIllIntentionedTask(?a, ?t)
    -> sqwrl:select(?a, ?t)

Agent(?a) ^ hasUnfeasibleTask(?a, ?t)
    -> sqwrl:select(?a, ?t)
```

By using OWL 2 primitives to model these spaces of the LD, we can define these task classes as mutually exclusive so that an agent can evaluate for each task that it is capable of, whether it would end up in any of the spaces.

Whenever an agent continues with a particular task, the task and resources consumed are logged so that in a next iteration the thresholds of acceptable behaviour can be updated.

4. Realization

In this section the outcomes of the MAAF application are presented, providing a possible pattern of agent guidance evolution through the LD. The selected industrial use-case aims at providing an understandable evaluation of permitted agent actions. Guidelines from the policy-based simulation, the experimental setup and a performance evaluation are presented as well.

4.1. LD application to the industrial use-case

As mentioned in Section 2.4, we started from an simpler case of hand tool production – hammers to be more specific – and then proceed by extending results to the general use-case. The implementation was initially supported only by simulation but we are currently proceeding with the physical deployment and demonstration of the MAAF in an excellence research lab for manufacturing, a learning factory and an open ecosystem for students and researchers to perform research on such topic (Fig. 10). It is an ideal place to study the challenges and to understand the benefits of elevating CPPS to a mature level of interoperability in a production context. The different illustrations of Fig. 11 guide over the conceptional evolution of the LD.

Figure 11(a) depicts both views of the production planning and the manufacturing execution paradigm of a hammer production. The former shows a linearised connection between the used resources. The requirements for a hammer is one handle and one head on the material demand side, which are consumed during the manufacturing process. Next to this, there is an expected preparation time and cost, which are derived from the calculated average. This cost consists of work labour, equipment and tool amortization and also energy consumption tariff. A wider perspective can be achieved closer to the shop-floor in the manufacturing execution view, where the production gets more details in respect the manufacturing operations. After the acquisition of materials, transforming tasks can start, followed by the assembling and packing ones. Every single task has an average production cost and execution time. The activities in the production routing depicted in Fig. 1 are in tune with the production tasks reported on Fig. 10(a).

Figure 11(b) presents a new representation of agents trustworthiness in manufacturing. Every production state has prescribed goal(s) and, if the routing permits it, alternative goals intended by the machine resources. The *Order Execution Space* is shaped as the region encompassing every possible routing alternatives and considering the technological tolerances of every task. The green-boundary curve will accepts (agent) operations if the tolerance

(a) SmartFactory Lab (b) e-Learning Factory lab

Fig. 10. Physical environment of experiment.

area is partially or fully contained within the maximum resource limits, while sorting all the feasible variations into a set for a better visual representation. This means, in practice, that one goal is unacceptable if it is located in the *Unacceptable Space* or if it is outside the acceptable CR of the agent (based on its current accountability level) in the current production state.

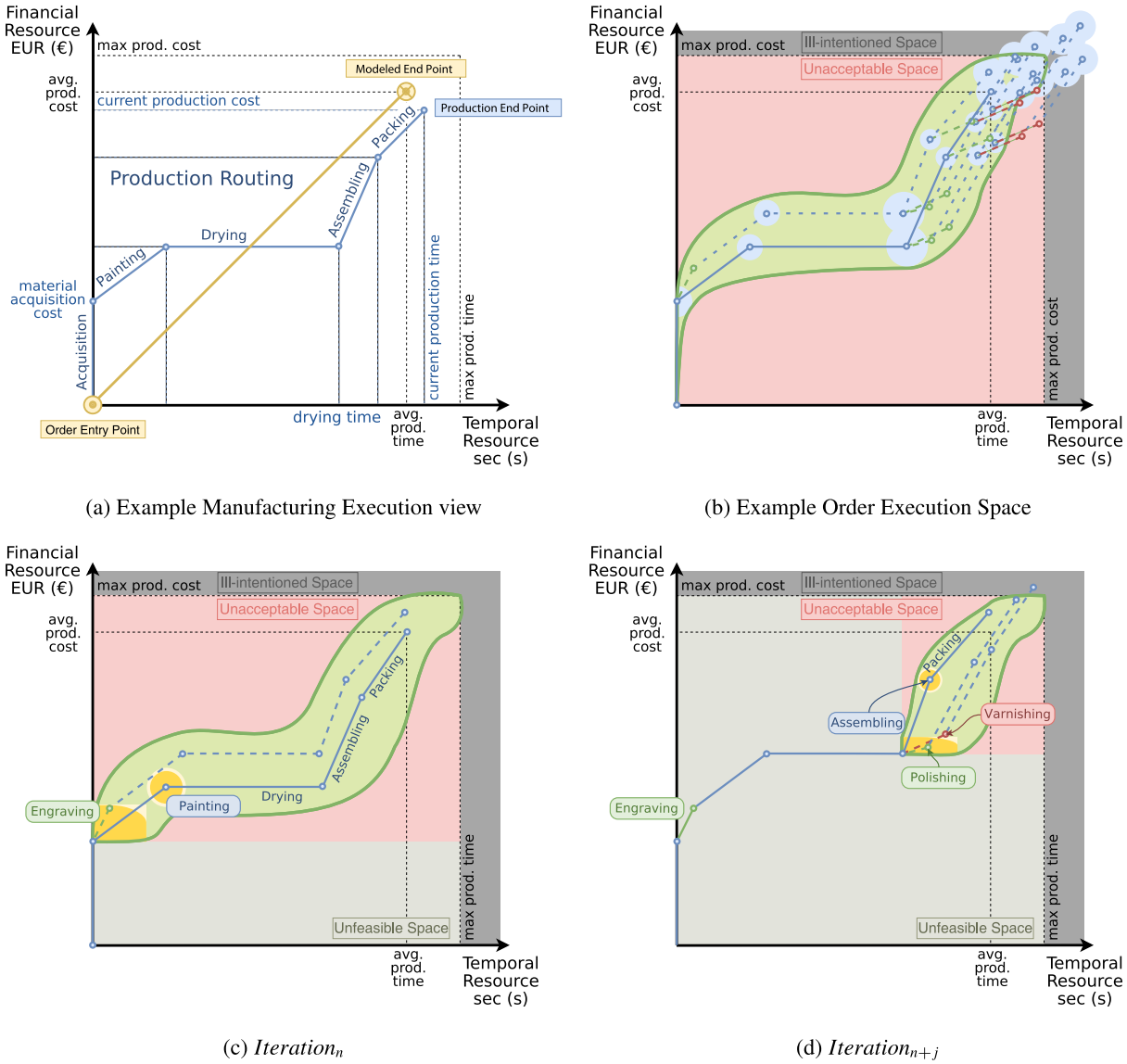
The last two illustrations (Fig. 11(c) and 11(d)) aim at depicting possible evolutionary stages of the *LD* in the manufacturing use-case. After each production state, the CR is recalculated on the basis of the additional resource bank still available and on the machine agent's accountability value. The acceptance of an alternative operation generates an overall shift of the production routing, as it probably requires a consumption of additional resources. This expendability of resource requirements necessitates the introduction of a well synthesised accountability value, as the execution of precedent alternative tasks (despite their feasibility) might preclude this possibility for later ones, and so for the entire manufacturing order execution. The choice made by the policy-based trust manager takes into consideration the constantly recalculated overall System Freedom.

4.2. Experimental setup and results evaluation

The relevance of hammers production in the overall production use-case can be unfolded if focusing on bigger batch sizes. With high fluctuation in the personnel staff of a construction company, identifying tools and toolsets can be mandatory for safety reasons and regulations. Adding a mid-quality RFID tag to a hammer (which does not really increases the cost of the product and keeps its usability unchanged) and so determining personnel responsibility can help significantly improve work conditions, safety and ethics. The construction company's procurement probably never thought about this possibility but if they start using the augmented tool, and the customer satisfaction feedback is positive, then the production company can think of standardizing this customization in future productions; otherwise it will be ignored. In both cases, the MAAF will be able to learn it.

The production company of the simulation use-case is a high quality hand tool producer. Its product portfolio consists of multiple kind of hammers, screwdrivers, saws, wrenches and electric screwdrivers. The ordered batch sizes range from one to several thousands and in order to serve all the different demands and keep the stock level low, they modularised their product part set (e.g. different tools with the same handle). The shop-floor consists of multiple human operated assembler stations, automatised manufacturing facilities and surface treatment machineries. The insular process oriented layout provides an opportunity of flexibility in the production control required by distributed control mechanisms.

The setup of our experimental environment starts with one hundred sets of randomised orders containing both defined and undefined demands in the ratio of four to one of the customer for one year ahead. Both the agents and the policy system were fine-tuned and optimised to the maximum production service level. Based on the defined expectations and the fixed policy rule sets each experiment runs for one year with eight hours work shifts a day.



(a) Example Manufacturing Execution view

(b) Example Order Execution Space

(c) $Iteration_n$

(d) $Iteration_{n+j}$

Fig. 11. Evolution of a LD during an order execution process.

Table 1 reports the diversity of products in Small- to Medium-size Enterprises (SMEs). The shop-floor consist of thirty machines, each directed by a corresponding agent. The configuration is listed in Table 2. The execution progress of our experiments was basically measured by iterating over the permissible maximal Agent Accountability value granted by the policy-based trust manager from zero to one, with an incremental step of one-hundredth.

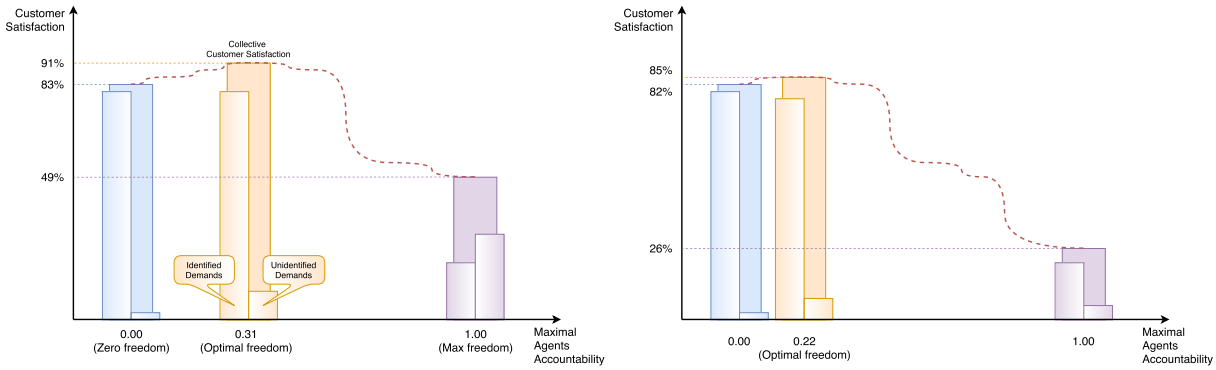
Collective customer satisfaction has been represented as the sum of its components derived form both the defined and undefined requirements, whose ratio was preset as four to one. In the first experimental setup (Fig. 12(a)) we simulate thirty agents, with all of them well-behaved, which means that they try to achieve a positive self-goal from the system's perspective. After ten thousand experimental runs we were able to pinpoint the ideal value for the Agent Accountability to 0.31 for the featured environment. A first observation is that the system performance in relation to customer satisfaction can increase with satisfactory freedom given to the agents. A second observation is that there is a huge drop in the satisfaction after a definite freedom (before that the variation is not so significant) which would

Table 1
Product variety

Product	Number of variety	Base material requirement	Obligatory routing tasks	Possible, extra routing tasks
Hammer	6	2	6	4
Screwdriver	11	2	5	6
Saw	4	3	6	2
Wrench	21	2	5	5
Electric Screwdriver	5	12	14	8

Table 2
Shop-floor configuration

Machines	Units on the shop-floor	Capabilities	Requirements
Manual operated painting station	3	Painting, Drying	Dye (consumable)
Manual operated assembler station	16	Assembling, Packing	Box (consumable)
Manual operated drilling station	5	Drilling, Polishing, Varnishing, Engraving	Drill (tool), Polisher (tool), Varnish (consumable), Graver (tool)
CNC lathing machine	2	Turning, Drilling, Polishing, Varnishing	Lathe tool (tool), Drill (tool), Polisher (tool), Varnish (consumable)
CNC milling machine	2	Milling, Drilling, Engraving	Milling tool (tool), Graver (tool), Drill (tool)
Automatic operated painting line	2	Painting, Drying	Dye (consumable)



(a) Level of optimal agent accountability

(b) Setup results in presence of ill-intentioned agents

Fig. 12. Effect of agent accountability on the collective customer satisfaction.

indicate that it is not advisable to tune too precisely the maximal accountability value to reach the beneficial zone in the utilization of this framework.

In the second experimental setup (Fig. 12(b)) we set the number of ill-intentioned agents to five, with the constraint that no task could be exclusively performed by ill-intentioned agents. The malicious agents try to initiate tasks out of order, or consume too many resources for the tasks at hand. As a consequence, even well-intentioned agents may be denied to execute certain tasks if the assets available to complete them are not sufficient, as an indirect effect of the propagation of malicious agents' behaviour. The goal of this experiment was also to measure the impact, from a performance point of view, for the requests submitted by the well-behaved agents. First observations indicate that the response times from the trust manager for ill-intentioned agents are lower than those for benign agents. This can be explained by the fact that finding a violation against a set of rules and constraints is usually faster than verifying that all conditions have been met. The second conclusion is that the trust manager is able to identify the ill-intentioned agents and reduce their accountability to zero, thus practically excluding them from production.

Examining the two charts together we conclude that even at zero maximal Agent Accountability the second setup performs worse, because of the increased unacceptable attempts of the ill-intentioned agents. The optimal freedom is lower also at the value of 0.22, which derive from the transitional time necessary to find and eliminate the malicious agents, during which they can harm the systems performance. The lower the maximal freedom, the shorter the period they work independently. Concluding it can be stated that, even in presence of malicious agents, from the point of view of customer satisfaction, the system's overall performance can benefit from the usage of the MAAF.

5. Conclusion and future work

When applied to the strategic level of authorizations agent-oriented mechanisms are able to capture a number of emergent features in real-life manufacturing scenarios. Exploitation of agent intelligence is enabled by the proper condition to freely act and reason on personal and global objectives, but this, in turn, generates industry concerns about their trustworthiness. In order to mitigate this perception, we have presented a dynamic accountability based framework which combines the advantages of an authorization-based middleware applied to the possible evolution of agent behavioural models. The proposed *Manufacturing Agent Accountability Framework* (MAAF) defined and enforced permission boundaries, in the light of which agents were freely permitted to exploit their intelligence in order to both reach individual and collective objectives. Declarative policies of the framework helped identify and avoid ill-intentioned behaviour in the execution of CPPS services. The MAAF also provided a visual representation of the overall system behaviour, which could be constantly monitored and evaluated for accountability audit against regions of acceptance by means of so-called *Leaf Diagrams*. Agents behavioural reliability was estimated against pre-modeled combinations of production resources, the *Order Execution Space*, which encompassed all of the possible behavioural deviations allowed to an agent via its CR.

The presented approach has been validated on a real manufacturing scenario for the production of a variety of hand tools, whereas the physical environmental model has been twinned with agent-based simulations. Nevertheless, the implementation of a support system for the visualization and interpretation of the agent accountability by LD has been initiated. In the next phase of this research work we plan to analyze how human capital, intended both as a production resource or an agent for a company, can be embodied in the accountability mechanism of the proposed framework. Nevertheless, an extension of the policy-based middleware will be investigated in order to prioritise aspects related to regulations or safety issues both at factory and shop-floor level.

Acknowledgements

This research has been supported by the GINOP-2.3.2-15-2016-00002 grant on an "Industry 4.0 research and innovation centre of excellence" and by the ED_18-2-2018-0006 grant on a "Research on prime exploitation of the potential provided by the industrial digitalisation". This research was also partially funded by the Research Fund KU Leuven and by the Flemish Government's Cybersecurity Initiative Flanders.

References

[1] E. Adamides, Responsibility-based manufacturing, *International Journal of Advanced Manufacturing Technology* **11** (1996), 439–448. doi:10.1007/BF01178970.

[2] M.M. Akbar and N. Parvez, Impact of service quality, trust, and customer satisfaction on customers loyalty, *ABAC Journal* **29**(1) (2009).

[3] J.M. Allwood and J.-H. Lee, The design of an agent for modelling supply chain network dynamics, *International Journal of Production Research* **43**(22) (2005), 4875–4898. doi:10.1080/00207540500168295.

[4] A.M. Aref and T.T. Tran, A decentralized trustworthiness estimation model for open, multiagent systems (DTMAS), *Journal of Trust Management* **2**(1) (2015), 3. doi:10.1186/s40493-015-0014-4.

[5] D. Artz and Y. Gil, A survey of trust in computer science and the semantic web, *Web Semantics: Science, Services and Agents on the World Wide Web* **5**(2) (2007), 58–71. doi:10.1016/j.websem.2007.03.002.

[6] L.D. Bacon, Using LISREL and PLS to measure customer satisfaction, in: *Sawtooth Software Conference Proceedings*, La Jolla, California, 1999, pp. 2–5.

[7] M. Baldoni, C. Baroglio, K. May, R. Micalizio and S. Tedeschi, Computational accountability in MAS organizations with ADOPT, *Applied Sciences* **8** (2018), 489. doi:10.3390/app8040489.
[8] Y. Bergner, J.J. Andrews, M. Zhu and J.E. Gonzales, Agent-based modeling of collaborative problem solving, *ETS Research Report Series* **2016**(2) (2016), 1–14. doi:10.1002/ets2.12113.
[9] M. Blaze, J. Feigenbaum and J. Lacy, Decentralized trust management, in: *Security and Privacy, 1996. Proceedings, 1996 IEEE Symposium on*, 1996, pp. 164–173. doi:10.1109/SECPRI.1996.502679.
[10] M. Blaze, J. Ioannidis and A.D. Keromytis, Experience with the KeyNote trust management system: Applications and future directions, in: *Trust Management: First International Conference, iTrust 2003 Heraklion, Crete, Greece, May 28–30, 2003 Proceedings*, P. Nixon and S. Terzis, eds, Springer, Berlin, Heidelberg, 2003, pp. 284–300. ISBN 978-3-540-44875-4. doi:10.1007/3-540-44875-6_21.
[11] P. Bonatti, C. Duma, D. Olmedilla and N. Shahmehri, An integration of reputation-based and policy-based trust management, *Networks* **2**(14) (2007), 10.
[12] S. Bussmann, N.R. Jennings and M.J. Wooldridge, *Multiagent Systems for Manufacturing Control. A Design Methodology, Springer Series on Agent Technology 1(14)*, 2004. doi:10.1007/978-3-662-08872-2.
[13] M. Cheminod, L. Durante, L. Seno and A. Valenzano, On the description of access control policies in networked industrial systems, in: *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*, 2014, pp. 1–10. doi:10.1109/WFCS.2014.6837594.
[14] Y.-H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick and M. Strauss, REFEREE: Trust management for Web applications, *Computer Networks and ISDN Systems* **29**(8) (1997), 953–964. doi:10.1016/S0169-7552(97)00009-3.
[15] A.M. Coroiu, Emotional intelligent agent in decision-making process with implications in manufacturing, *International Journal of Modern Manufacturing Technologies* **7**(2) (2015), 43–47.
[16] B. Glimm, I. Horrocks, B. Motik, G. Stoilos and Z. Wang, HermiT: An OWL 2 reasoner, *J. Autom. Reason.* **53**(3) (2014), 245–269. doi:10.1007/s10817-014-9305-1.
[17] T.D. Huynh, N.R. Jennings and N.R. Shadbolt, An integrated trust and reputation model for open multi-agent systems, *Autonomous Agents and Multi-Agent Systems* **13**(2) (2006), 119–154. doi:10.1007/s10458-005-6825-4.
[18] N.R. Jennings, P. Faratin, T.J. Norman, P. O'Brien, B. Odgers and J.L. Alty, Implementing a business process management system using adept: A real-world case study, *Applied Artificial Intelligence* **14**(5) (2000), 421–463. doi:10.1080/088395100403379.
[19] T. Karaulova, M. Kostina and E. Shevtshenko, Reliability assessment of manufacturing processes, *International Journal of Industrial Engineering and Management* **3** (2012), 143–151.
[20] S. Karnouskos and P. Leitão, Key contributing factors to the acceptance of agents in industrial environments, *IEEE Transactions on Industrial Informatics* **13** (2017), 696–703. doi:10.1109/TII.2016.2607148.
[21] J.-H. Lee and C.-O. Kim, Multi-agent systems applications in manufacturing systems and supply chain management: A review paper, *International Journal of Production Research* **46**(1) (2008), 233–265. doi:10.1080/00207540701441921.
[22] N. Li, J.C. Mitchell and W.H. Winsborough, Design of a role-based trust-management framework, in: *Security and Privacy, 2002*, Proceedings. 2002 IEEE Symposium on, 2002, pp. 114–130. doi:10.1109/SECPRI.2002.1004366.
[23] W.-Y. Liang and C.-C. Huang, The agent-based collaboration information system of product development, *International Journal of Information Management* **22**(3) (2002), 211–224. doi:10.1016/S0268-4012(02)00006-3.
[24] Q. Liu, X. Zhang, X. Chen and L. Wang, The resource access authorization route problem in a collaborative manufacturing system, *Journal of Intelligent Manufacturing* **25**(3) (2014), 413–425. doi:10.1007/s10845-012-0690-1.
[25] F. Mannhardt, S.A. Petersen and M.F. Oliveira, A trust and privacy framework for smart manufacturing environments, *Journal of Ambient Intelligence and Smart Environments* **11**(3) (2019), 201–219. doi:10.3233/AIS-190521.
[26] M. Marques, C. Agostinho, G. Zacharewicz and R. Jardim-Gonçalves, Decentralized decision support for intelligent manufacturing in Industry 4.0, *Journal of Ambient Intelligence and Smart Environments* **9**(3) (2017), 299–313. doi:10.3233/AIS-170436.
[27] H. Min and G. Zhou, Supply chain modeling: Past, present and future, *Computers & Industrial Engineering* **43**(1) (2002), 231–249. doi:10.1016/S0360-8352(02)00066-9.
[28] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn and K. Ueda, Cyber-physical systems in manufacturing, *CIRP Annals - Manufacturing Technology* **65**(2) (2016), 621–641. doi:10.1016/j.cirp.2016.06.005.
[29] L. Monostori, P. Valckenaers, A. Dolgui, H. Panetto, M. Brdys and B.C. Csáji, Cooperative control in production and logistics, *Annual Reviews in Control* **39** (2015), 12–29. http://www.sciencedirect.com/science/article/pii/S1367578815000024. doi:10.1016/j.arcontrol.2015.03.001.
[30] L. Monostori, J. Váncza and S.R.T. Kumara, Agent-based systems for manufacturing, *CIRP Annals* **55**(2) (2006), 697–720. http://www.sciencedirect.com/science/article/pii/S1660277306000053. doi:10.1016/j.cirp.2006.10.004.
[31] L. Nie, Y. Bai, X. Wang, K. Liu and C. Cai, An agent-based dynamic scheduling approach for flexible manufacturing systems, in: *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2012, pp. 59–63. doi:10.1109/CSCWD.2012.6221798.
[32] M. O'Connor and A. Das, SQWRL: A query language for OWL, in: *Proceedings of the 6th International Conference on OWL: Experiences and Directions*, OWLED'09, Vol. 529, CEUR-WS.org, Aachen, Germany, 2009, p. 208–215.
[33] D. Preuveneers and E. Ilie-Zudor, The intelligent industry of the future: A survey on emerging trends, research challenges and opportunities in Industry 4.0, *Journal of Ambient Intelligence and Smart Environments* **9**(3) (2017), 287–298. doi:10.3233/AIS-170432.
[34] D. Preuveneers and W. Joosen, Access control with delegated authorization policy evaluation for data-driven microservice workflows, *Future Internet* **9**(4) (2017). doi:10.3390/fi9040058.
[35] D. Preuveneers, W. Joosen and E. Ilie-Zudor, Policy reconciliation for access control in dynamic cross-enterprise collaborations, *Enterprise Information Systems* **12**(3) (2018), 279–299. doi:10.1080/17517575.2017.1355985.

[36] J. Sabater and C. Sierra, REGRET: Reputation in gregarious societies, in: *Proceedings of the Fifth International Conference on Autonomous Agents, AGENTS '01*, ACM, New York, NY, USA, 2001, pp. 194–195. ISBN 1-58113-326-X. doi:10.1145/375735.376110.

[37] Y. Sudo and M. Matsuda, Agent based manufacturing simulation for efficient assembly operations, *Procedia CIRP* **7** (2013), 437–442. doi:10.1016/j.procir.2013.06.012.

[38] A. Surana, S. Kumara, M. Greaves and U.N. Raghavan, Supply-chain networks: A complex adaptive systems perspective, *International Journal of Production Research* **43**(20) (2005), 4235–4265. doi:10.1080/00207540500142274.

[39] W.T.L. Teacy, J. Patel, N.R. Jennings and M. Luck, TRAVOS: Trust and reputation in the context of inaccurate information sources, *Autonomous Agents and Multi-Agent Systems* **12**(2) (2006), 183–198. doi:10.1007/s10458-006-5952-x.

[40] P. Valckenaers, Perspective on holonic manufacturing systems: PROSA becomes ARTI, *Computers in Industry* **120** (2020), 103226. http://www.sciencedirect.com/science/article/pii/S0166361520302530. doi:10.1016/j.compind.2020.103226.

[41] P. Valckenaers and H. Brussel, Design for the Unexpected, From Holonic Manufacturing Systems towards a Humane Mechatronics Society, 2015, ISBN 9780128036624. doi:10.1016/C2014-0-04226-8.

[42] H. van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts and P. Peeters, Reference architecture for holonic manufacturing systems: PROSA, *Computers in Industry* **37**(3) (1998), 255–274. doi:10.1016/S0166-3615(98)00102-X.

[43] B. Vogel-Heuser, J. Lee and P. Leitão, Agents enabling cyber-physical production systems, *Automatisierungstechnik* **63**(10) (2015), 777–789. doi:10.1515/auto-2014-1153.

[44] S. Wang, J. Wan, D. Zhang, D. Li and C. Zhang, Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination, *Computer Networks* **101** (2016), 158–168. http://www.sciencedirect.com/science/article/pii/S1389128615005046. doi:10.1016/j.comnet.2015.12.017.

[45] R.Y. Zhong, X. Xu, E. Klotz and S.T. Newman, Intelligent manufacturing in the context of industry 4.0: A review, *Engineering* **3**(5) (2017), 616–630. http://www.sciencedirect.com/science/article/pii/S2095809917307130. doi:10.1016/J.ENG.2017.05.015.