Nat Comput (2012) 11:417–430
DOI 10.1007/s11047-011-9295-4

Degeneracy and networked buffering: principles for supporting emergent evolvability in agile manufacturing systems

Regina Frei · James Whitacre

Published online: 5 November 2011

Abstract This article introduces new principles for improving upon the design and implementation of agile manufacturing and assembly systems. It focuses particularly on challenges that arise when dealing with novel conditions and the associated requirements of system evolvability, e.g. seamless reconfigurability to cope with changing production orders, robustness to failures and disturbances, and modifiable user-centric interfaces. Because novelty in manufacturing or the marketplace is only predictable to a limited degree, the flexible mechanisms that will permit a system to adequately respond to novelty cannot be entirely pre-specified. As a solution to this challenge, we propose how evolvability can become a pervasive property of the assembly system that, while constrained by the system's historical development and domain-specific requirements, can emerge and re-emerge without foresight or planning. We first describe an important mechanism by which biological systems can cope with uncertainty through properties described as *degeneracy* and *networked buffering*. We discuss what degeneracy means, how it supports a system facing unexpected challenges, and we review evidence from simulations using evolutionary algorithms that support some of our conjectures in models with similarities to several assembly system contexts. Finally, we discuss potential design strategies for encouraging emergent changeability in assembly systems. We also discuss practical challenges to the realization of these concepts within a systems engineering context, especially issues related to system transparency, design costs, and efficiency. We discuss how some of these difficulties can be overcome while also elaborating on those factors that are likely to limit the applicability of these principles.

Keywords Networked buffering · Degeneracy · Agile manufacturing · Assembly systems · Robotics · Multi-agent systems · Complexity · Evolvability

Regina Frei received a fellowship for prospective researchers from the Swiss National Science Foundation while this article was written.

R. Frei (✉)
South Kensington, Imperial College London, London SW7 2AZ, UK
e-mail: work@reginafrei.ch

J. Whitacre
University of Birmingham, Edgbaston B15 2TT, UK
e-mail: jwhitacre79@gmail.com

1 Introduction

Developments in engineering and technology have repeatedly taken cues from properties found in biology, mainly for designing individual systems. However, there may be an even bigger potential for collective and networked systems-of-systems to adopt principles observed in nature. In addition to self-* properties and emergence (Frei and Barata, 2010), degeneracy and networked buffering are promising characteristics that if adopted by engineered systems may improve their adaptability towards novel stresses.

But why should designers, engineers, planners, analysts, and decision makers care about the concept of *emergent engineering*?

The hypothesis explored in this article is that there is an important (and growing) set of problems for which traditional engineering paradigms are now known to be insufficient and where new biological paradigms can be shown to be more effective. These problems are characterised firstly by the presence of unpredictability that arises across multiple timescales and necessitates that systems display an inherent propensity to be modified and adapted to novel conditions. Mass customisation, operational volatility, and strategic uncertainty are common features of these problems and subsequently require systems to display reconfigurability, robustness, and evolvability.

We reflect on the conditions under which distributed self-organised systems can display new emergent properties at the system level which are congruent with system objectives yet driven largely by boundedly rational individuals undergoing short-sighted and possibly selfish decisions.

Emergent properties often involve some element of surprise and are not necessarily beneficial. In this article, we describe the necessary conditions for the realisation of a particular emergent property that directly contributes to a system's reconfigurability, robustness, and evolvability and thus represents a potentially important example whereby emergent properties directly contribute to the performance of engineered systems-of-systems.

Organisation of this article Section 2 details some of the challenges that are faced by agile assembly systems. Section 3 discusses complexity in biology and engineering. Section 4 explains how complex systems in nature use degeneracy, networked buffering and evolvability to cope with challenges associated with uncertainty, and applies these concepts to agile assembly systems. Our arguments for the realisation of a particular form of engineering emergence are supported from simulations of evolutionary processes. Because these have a close correspondence to standing challenges in nature-inspired computing, we use Sect 4.2 to discuss how previous simulation results are relevant to that field of study and how the challenges being faced in that field have important similarities to those faced in the design of evolvable technological artifacts. Finally, Sect. 5 discusses some of the benefits and open questions surrounding the approach proposed in this article, and Sect. 6 concludes.

1.1 Challenges in agile assembly

The production and assembly of small mechanic, electronic and mechatronic products such as mobile phones, ipods, computer mice, remote controls, watches, washing machine handles and coffee machines is today mostly automated; robots in the industrial shop floor assemble the product parts according to the customers' orders.

Assembly is an important component of the manufacturing process that is common to almost all modern industrially produced goods. In some cases, the concepts developed for assembly are applicable to manufacturing in general; in particular for operations such as milling, drilling, turing, painting, marking, quality checks, packing, and others.

The robots used for the automation of assembly and other manufacturing processes are either complete industrial robots of diverse types or, increasingly, composed of versatile modules. Prior to the presence of modularity in manufacturing and mass-customisation in products, most manufacturing and assembly lines (Koren et al. 1999) were dedicated and custom-built facilities that mass-produced a specific product. It typically takes several months to build and program such a system, which will then operate under central top-down control. These systems cope poorly with unexpected failures and disturbances; if they can be adapted or changed at all, the re-engineering and re-programming is a work-intensive and error-prone procedure. Such manufacturing facilities are increasingly unsuitable for today's dynamic, customer-driven markets; innovative solutions for coping with new products/services are becoming a standard system requirement. Over the last two or three decades, numerous collaborative projects involving academia and industry have proposed alternative conceptual approaches for the design of adaptive manufacturing facilities. These projects have broadly followed four main paradigms (and variations thereof), some of which are at least partially complementary: flexible, reconfigurable, holonic and agile manufacturing. The following overview of these paradigms is brief—more extensive discussion is available in (Frei 2010; Frei and Di Marzo Serugendo 2011c).

Flexible manufacturing systems (FMS) (Buzacott and Shanthikumar 1980; Kaula 1998; Onori and Groendahl 1998) are composed of machines that display a predefined set of manufacturing capabilities, which makes them highly sophisticated and potentially difficult to manage (Barata et al. 2005). The likelihood of paying for unutilised/wrong capabilities is high if such systems were to be implemented within a dynamic manufacturing environment. On the other hand, for companies that are confident that their capability requirements will not change over several years, FMS may provide a suitable solution.

Reconfigurable manufacturing systems (RMS) (Koren et al. 1999; Mehrabi et al. 2000; ElMaraghy 2006) aim to develop modular systems in which an engineer can add / remove functionalities according to current demands. Modularity is viewed as an important precondition for promoting shop-floor level agility and recent efforts in the area of RMS focus on reconfigurable machines (Katz 2007) and the evolution of product characteristics (ElMaraghy et al. 2008). While conceptually promising, the elaboration of guidelines for these design principles and the associated control strategies are thus far lacking.

Holonic manufacturing systems (HMS) (Valckenaers and Van Brussel 2005; Marik et al. 2007) follow a paradigm based on the so-called *holarchies*, as suggested by Koestler (1967): every item is a whole as well as a component of a larger whole. At their inception, holonic systems were strongly inspired by evidence that many natural systems are organised into dynamic hierarchies; however, with time, these approaches have primarily become top-down solution strategies and consequently have become less suitable for facilitating rapid adaptation. ADACOR (Leitão 2004; Leitao and Restivo 2008) combines holonics with the concept of self-organisation by using principles based on pheromone attraction for task attribution.

Agile manufacturing systems are distributed autonomous systems. This paradigm was developed to cope with frequently changing requirements, low production volumes, multiple product variants, as well as perturbations and failures. Mechanical system reconfigurations are facilitated by modular hardware, but (re-)programming manufacturing systems often remains as a tedious, manual, work-intensive and error-prone procedure.

The *Architecture for Agile Assembly* (AAA) (Rizzi et al.1997; Kume and Rizzi 2001; Hollis et al. 2003) considers a distributed system of self-representing cooperative modules equipped with information about their own capabilities and negotiation skills to communicate with their peers. The programming (Gowdy and Rizzi 1999) is agent-based, but does not consider self-* properties. Recent advances (Hollis et al. 2003; Niemeyer 2006) in AAA concern mechanic modules with a concept where not only the robot moves with two *degrees of freedom, DoF*), but also the carriers are planner motors which move on a platen (two additional DoF). Research into AAA has mainly presented technological solutions for specific manufacturing tasks such as visual gripping, cascaded lenses and special algorithms for object recognition. A similar concept to AAA is seen in a German project known as *MiniProd* (Gaugel et al. 2004; Hanisch and Munz 2008)[1]; website in German, which involves a collaboration between several industrial and academic partners.

Some system designers have taken inspiration from natural complex systems to build agile manufacturing systems (Ueda 2006; Leitão 2008; Frei et al. 2007), with additional influence from Autonomic (Kephart and Chess 2003), Pervasive Adaptation[2] / Ubiquitous[3] and Organic (Wuertz 2008) computing concepts.

An agile manufacturing system can be considered as a multi-agent system, which needs to fulfil specific tasks. Manufacturing resources are agentified; similarly, product orders and parts are represented by agents. Numerous systems for multi-agent control systems in manufacturing have been proposed, reaching from enterprise resources management to order scheduling and shop-floor control (Marik et al. 2007; Shen et al. 2006; Vrba 2003). Some projects were deployed in industry (Bussmann 2000). Changes in the shop-floor can be automated through an ontology-based reconfiguration agent (Al-Safi and Vyatkin 2010); this is, however, a centralised top-down approach for managing an otherwise distributed system. For software agents which represent robotic modules to gain more autonomy in achieving their goals, they need a representation of their own body as well as their relations with their peers and the environment (Frei 2010; Vallée et al. 2009).

The following introductory subsections explain one of the agile approaches currently being developed—*self-organising evolvable assembly systems*—which focuses on facilitating evolvability and self-organisation.

1.2 Evolvable Assembly Systems (EAS)

Evolvable Assembly Systems (EAS) (Onori 2002; Barata 2005) consist of robotic modules of varying granularity. A module is either an entire industrial robot with several skills (i.e. screwing, rotating and linearly moving) or a simpler module such as a robotic axis, a gripper, a feeder, or a conveyor having a single skill only. Every module is an embodied agent with self-knowledge (about its skills and physical characteristics) as well as communication/coordination capabilities (to coordinate its work with other modules). Modules engage in coalitions (see Fig. 1) to provide the composite skills necessary to assemble the product at hand. For instance, a gripper able to seize and release parts forms a coalition with a linear axis to compose a *pick&place* skill.

Evolvability refers to a system's ability to continuously and dynamically undergo modifications of varying importance in order to maintain or improve competitiveness: from small adaptations, e.g. in the timing and placement of component interactions, to larger changes in system behavior (Frei 2010). To understand evolvability in an assembly systems context, it is important to take into

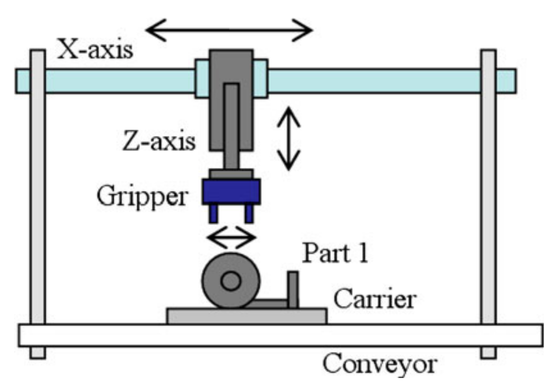

Fig. 1 EAS module coalition

---
[1] http://miniprod.com/frame_01.html.

[2] http://www.perada.eu.

[3] http://sandbox.xerox.com/ubicomp.

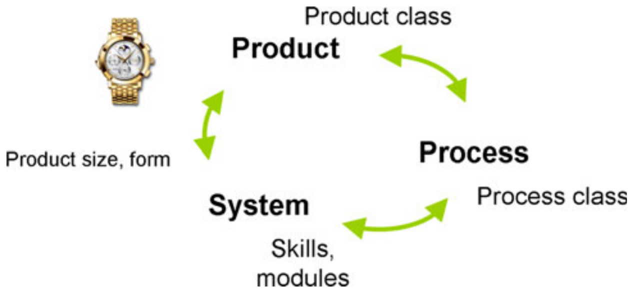

Fig. 2 Product, processes and system

account the mutual causal relations between product design, assembly processes and the assembly system itself, as illustrated in Fig. 2 and discussed in (Semere et al. 2008; Frei 2010). Each *product* belongs to a particular product class and each production *process* refers to a coherent suite of assembly operations which generate a finished product by assembling a set of parts. Production processes, the product design and the assembly system are intimately linked: any change in the product design has an impact on the processes to apply and on the actual assembly system to use. Similarly, any change in a process (for instance replacing a rivet by a screw) may imply a change in the product design, and will almost invariably impact assembly system requirements. Evolvability requires seamless integration of new modules independently of their brand or model.

Modules that comprise an assembly system either include local controllers, or are associated with separate virtual agents; either way, the modules have some degree of autonomy to make decisions based on local information. The heterogeneity of the modules (nature, type, vendor) does not prevent them from forming a homogeneous agent society at the software level. Software wrappers (also called Agent Machine Interfaces, AMI, in (Barata 2005)) allow the generic agents to represent any type of robotic module.

Feasible coalitions in EAS are statically created (offline) by an engineer. Modifying a coalition implies redesigning and re-programming the whole assembly system.

1.3 Self-Organising Assembly Systems (SOAS)

Self-Organising Assembly Systems (Frei et al. 2008; Frei 2010) extend EAS in the following way: given a product order *(generic assembly plan—GAP)* provided as input, the modules read task specifications and autonomously compose suitable coalitions with the goal of providing the required skills. Modules typically provide simple skills, and when forming coalitions, they provide composite skills, based on their compatibilities and specific composition rules (details in (Frei 2010)). Once each task is associated with a coalition—and this has been confirmed by the user—the coalitions arrange themselves in accordance with the shop-floor layout.

The modules also derive their *layout-specific assembly instructions—LSAI* themselves, based on the GAP. The result of this self-organising process is a new or reconfigured assembly system that will assemble the ordered product. An appropriate assembly system will *emerge* from a self-organisation process, modelled on the basis on the Chemical Abstract Machine (Berry and Boudol 1998), as follows.

Any new product order triggers a self-organising process, which eventually leads to a new appropriate system. There is no central control authority, although the user may intervene if necessary.

Similar to the formation of complex molecular assemblies within a cell (Kurakin 2009), robotic modules progressively aggregate with each other to fulfil the product order (Frei et al. 2010) (illustrated in Fig. 3). Because order specifications define the required task sequence, the self-organisation process becomes regulated so that, under ideal operating conditions, each formed coalition presents a required skill that is executed in the correct operation sequence.

This automated process extends beyond layout creation. During production, whenever a failure occurs in one or more of the currently used modules of the system, the process may lead to three different outcomes: (1) the current modules adapt their behaviour (change speed, force, task distribution, etc.) to cope with the failure, possibly degrading performance in order to maintain functionality; (2) the module fails to achieve the task and it is replaced by another module, thereby leading to a repaired system; (3) the system is unable to compensate for the failure. The actions taken by the system will depend on the availability

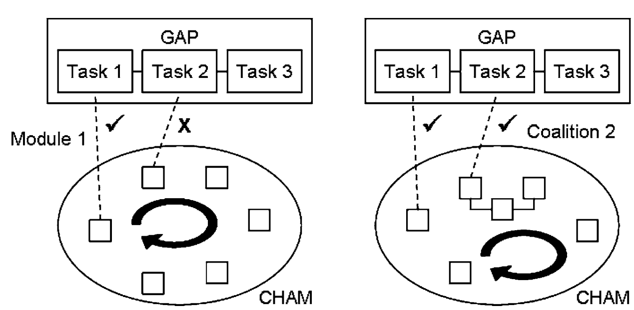

Fig. 3 The chemical abstract machine (CHAM) applied to self-organising assembly systems (SOAS), where GAP stands for the generic assembly plan which is composed of a set of tasks. The blocks in the "chemical soup" represent the modules which react with the GAP and each other to provide the requested skills and, if necessary, therefore form suitable coalitions

of resources and on specific production constraints (cost / speed / precision).

An SOAS is thus an EAS with two additional features: (1) modules *self-organise* to produce a suitable layout for the assembly of the ordered product and (2) the assembly system as a whole *self-adapts* to production conditions and *self-manages* its behaviour.

The realisation of the SOAS paradigm requires pervasive adaptation in the face of several inter-related types of uncertainty. This uncertainty originates from a lack of perfect knowledge about future system and environmental states and can result in the emergence of sub-optimal system configurations or the creation of sub-optimality through unexpected changes in the environment. Resolutions to such challenges are broadly relevant across complex systems science in general and emergent engineering studies in particular. Importantly, SOAS must incorporate strategies that can allow a system to remain evolvable under complex and ever novel conditions. The remainder of this article outlines concepts that are intended to resolve several important evolvability preconditions. We argue that these concepts can help to facilitate adaptation at the configuration, operation, and design levels of assembly systems and thus could prove invaluable to agile manufacturing paradigms in general and SOAS in particular.

2 Complexity in biology and engineering

In biological evolution, continued species survival requires that incremental adaptive design changes can be discovered that do not lead to a propagation of redesign requirements in other components in the system, i.e. macro-mutation is a negligible contributor to the evolution of complex species. Instead, single heritable (design) changes are found that lead to (possibly context-specific) novel interaction opportunities for a component, flexible reorganisation of component interactions (that can robustly preserve other core system functionalities), and in some cases a subsequent compounding of novel opportunities within the system (Kurakin 2009). In other words, the requirement is one of incremental changes in design and compartmentalised, but not necessarily incremental, changes to system behaviour. While occasional slowdowns in the tempo of biological evolution are known to take place (e.g. under stabilising selection), there is no evidence from paleontology or population genetics studies to suggest that biological systems display the same built-up tension or sensitivity to incremental genetic changes as technological systems display towards incremental engineering design changes.

The dynamic attributes of biological evolution are perplexing to engineers, especially considering that sophisticated services in biological systems involve the execution of many distinct sub-functions and process pathways. Importantly, the building blocks that generate these sophisticated biological services/traits are not single purpose devices with predefined functionality but instead display a high degree of functional plasticity and degeneracy.

Functional plasticity refers to the presence of multifunctional components (e.g. proteins, molecular assemblies, cells, organisms) that change what function they execute depending on their local context. Primarily observed in biological systems, degeneracy refers to the existence of functionally plastic components (but also modules and pathways) that can perform similar functions (i.e. are effectively interchangeable) in certain conditions, but can perform distinct functions in other conditions, i.e. components are partially overlapping in their functionalities; see Fig. 4 (Whitacre et al. 2011). Degeneracy contributes to local compensatory effects because it provides a biological system with different options for performing a given function, which can be used to compensate for the failure of a component class and helps in dealing with small changes in the requirements associated with the realisation of a particular functional outcome (Edelman and Gally 2001).

As we discuss in Sect. 4, degeneracy affords a weaker coupling between the functions performed and the components involved in achieving them (Kirschner and Gerhart 1998), and can lead to emergent forms of system flexibility that increase a system's options for dealing with novel stresses. Within an abstract design space or fitness landscape, one could say that traditionally designed systems find themselves on isolated adaptive peaks while biological systems reside on richly connected neutral plateaux. While complex systems research has repeatedly used the rugged fitness landscape metaphor to advocate for greater emphasis in disruptive/explorative design changes, this is neither required nor observed in biological evolution. To clarify these points, we next discuss conflicts that arise between a system's complexity and its adaptability in designed systems and then we discuss how these conflicts might be resolved through degeneracy.

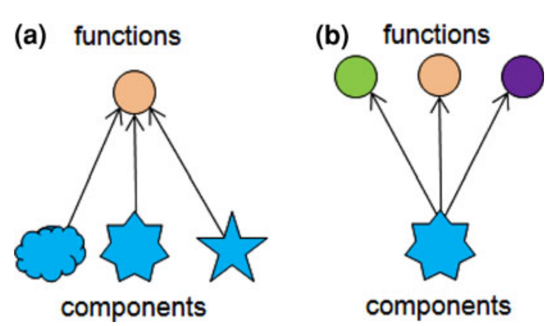

Fig. 4 **a** Functional redundancy, **b** functional plasticity

2.1 Evolvability-complexity conflicts in engineering

While the term *complexity* generally relates to the interdependence of component behaviour / actions / functions, it is an otherwise ambiguous term and there is no consensus as to its meaning or measurement. In engineering, complexity is often used to describe sophisticated services involving several entities and their multi-lateral interactions (Brueckner 2000; Leitão 2004; Frei and Di Marzo Serugendo 2011a, b). Below we recount a typical narrative surrounding the tension between complexity and design adaptation in systems engineering.

Starting with a single device, the number and exactness of functional requirements / specifications / constraints generally influences the proportion of operating conditions that can meet these requirements. While the trade-off between operating constraints and operational feasibility is not always simple (e.g. linear, monotonic) even for a single device, it is widely acknowledged that multi-component services tend to become more sensitive to novel internal and external conditions as more components are added to the system that co-specify the feasible operating conditions of other system components. In particular, the operating requirements placed on each component become more exacting as its function becomes more reliant on the actions / states / behaviours of others, e.g. through direct interaction, through sharing or modifying the same local resources, or indirectly through failure propagation. These challenges are broadly observed and represent important heuristic knowledge for engineers in industries such as biotechnology (e.g. bioreactors, biologic purification), nanotechnology (e.g. production of electrostatically sensitive devices), precision assembly (e.g. propagation of tolerance exceeding), or rigidly automated production lines (e.g. one blocked machine can take the entire system down).

Services achieved through complex engineering artifacts tend to be more fragile to atypical component behaviours or atypical events because a greater proportion of events will exceed the operational tolerance thresholds in at least one device, with the propagation characteristics of these threshold-crossing events determining the likelihood of sub-system and system-wide failure.

These common operational challenges contribute to difficulties associated with changing product specifications, changing production processes and changing the design of assembly systems. With systems designed from single purpose devices that are uniquely suitable for a process-critical function, this establishes a tight coupling between system performance metrics, the reliability of a function, the continued normal operation of the device providing that function, and the continued compatibility of that device with other interacting devices.

To reduce the frequency of failures, a design approach is often taken that assumes predictability (e.g. in requirements, operations, perturbations) or relies on empirically driven placement of backup devices, storage / maintenance / preservation facilities, and related fail-safe principles. These design principles often introduce inefficiencies but have proven important in complex engineered systems for reducing failure propagation, for ensuring products are not lost, and for ensuring that adequate time is available to undertake any required system reconfiguration / retooling / repair / redesign.

How to achieve adaptation while maintaining higher efficiency in a complex operational setting is not straightforward or obvious. A number of discussions in the literature have implied that technological artifacts reside near a Pareto optimal adaptability-efficiency trade-off surface and that the comparatively higher propensity for adaptation in biology is only achievable due to lower levels of efficiency. In the following sections, we discuss conceptually simple principles that are believed to facilitate efficient adaptation in complex biological systems. Along with reviewing these principles, we also discuss recent simulation studies that thus far support the relevance of these principles for adaptation within several classes of complex systems. We then discuss how these concepts can be directly transferred to assembly systems and also comment on their broader relevance within systems engineering.

3 Strategies of natural complex systems: adaptation through degeneracy and networked buffering

3.1 Experimental evidence of degeneracy and networked buffering

To adapt, a system must be provided with access to many distinct options for changing its output or behavior and the system must be able to take these options and transform them into innovations that are useful within the context of a particular environment.

Theoretical arguments have been put forth over the last decade suggesting that degeneracy supports both of these prerequisites for adaptation (Edelman and Gally 2001; Whitacre and Bender 2010a; Whitacre 2010a) and recently there has been some evidence from simulation studies that has provided some empirical support for these conjectures. For instance, in simulations of genome:proteome mappings and in agent-based models, degeneracy has been found to considerably enhance the number of accessible design change options for a system (see heritable phenotypic variation in Whitacre and Bender (2009, 2010a)). Further studies have found an unexpectedly high proportion of these options can be utilised as positive adaptations

(Whitacre 2010b; Whitacre et al. 2010) and can sometimes afford further opportunities when these systems are presented with new environments (Whitacre 2010c).

In attempting to understand how random novelties are transformed into adaptations, it was shown in (Whitacre and Bender 2010b) that high levels of degeneracy lead to the emergence of pervasive flexibility in how a system organises itself and in this way can allow a decoupling between the robustness of some functions and the modification of others. The means by which this can be achieved has been described as the networked buffering hypothesis in Whitacre and Bender (2010b) and is conceptually illustrated using the diagrams in Fig. 5. Shown in each of the panels in Fig. 5 are systems of agents, which could represent proteins within a cell, species in a food-web, or devices comprising an assembly system. For educative purposes, we simplify the illustration so that each agent is only capable of performing one of two distinct functions.

The agents are depicted as pairs of connected nodes and the nodes are positioned in such a manner such that spatial distance within the diagram indicates similarity in function. In Panels a–c, we show high levels of system degeneracy, i.e. many multi-functional agents that are partially related to one another in function, while Panel d displays a system with no degeneracy. In Panel b, we consider a situation in which an agent has failed to perform function Z and the system now needs to attempt to perform this function by other means, i.e. by having another agent attempt to take its place. Because there are more agents assigned to function X than are needed, and because of the degeneracy in the system, the agents can undergo a series of role reassignments (as indicated by the arrows with the switch symbols), which provides the system with the means by which to attempt a response to this challenge. In other words, degeneracy allow extra resources related to one function to indirectly support entirely unrelated functions in a system. As shown in Panel c, depending on where we have extra agent resources, there are potentially many different ways in which the system could respond to this unexpected change (as indicated by the additional arrows with switch symbols) and thus there is a greater chance that the system can be reconfigured to deal with novel conditions. Conversely, consider a situation where we now have excess resources related to function Z. There are many different ways in which these resources can be used to support unrelated functions in the system, which is seen by reversing the flow of arrows in panel c. This implies that small amounts of excess resources can be used in a highly

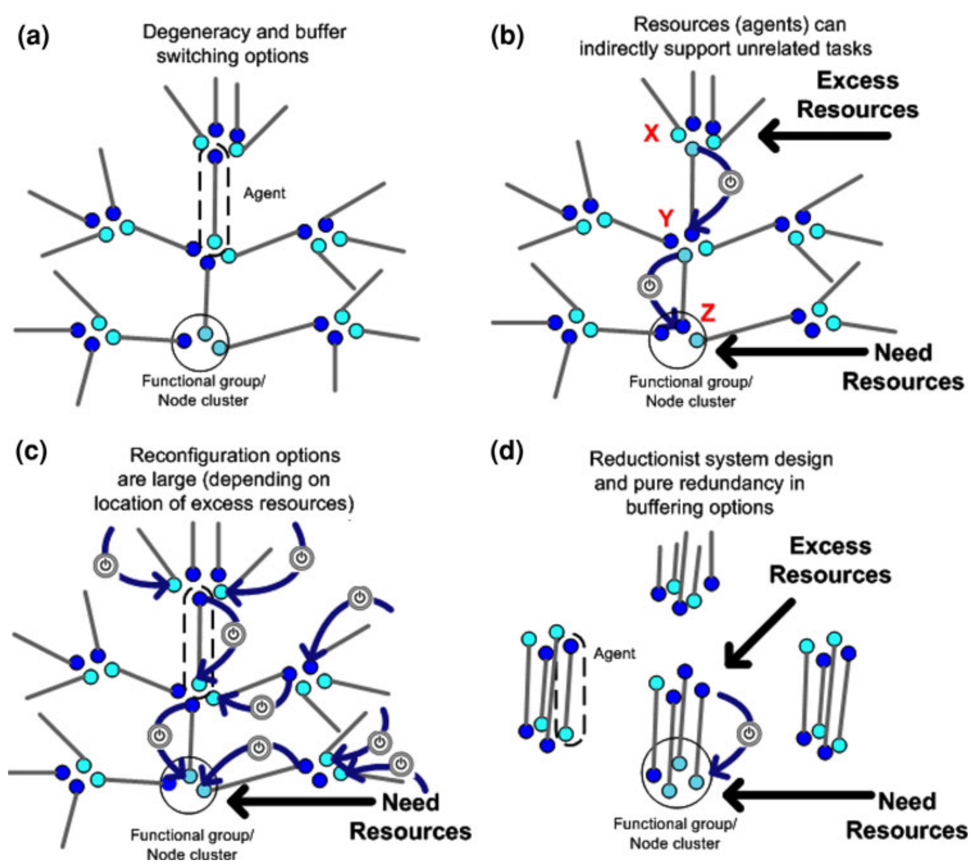

Fig. 5 The networked buffering hypothesis illustrated for a multi-agent system

versatile manner with a multiplier effect on overall system robustness. In other words, there is dramatically lower amounts of inefficiency that are required to achieve a high level of system robustness. We have confirmed such attributes within simulation studies of genome:proteome mappings where it was found that degeneracy approximately doubles the robustness that is gained from excess resources in comparison with systems where degeneracy was entirely absent (Whitacre and Bender 2010b).

In accordance with Ashby's Law of Requisite Variety (Ashby 1956), robustness is intimately tied to the number of response options that are available to a system. One can immediately see from Panel d that the number of reconfiguration options becomes greatly reduced when degeneracy is replaced by pure redundancy. While pure redundancy is costly in technological systems because redundant components remain idle, degeneracy allows a system to use its components in different ways, so that they are more consistently utilised under different system-level requirements.

On the other hand, systems which are composed of a wider range of components exhibiting degeneracy are more complex and thus may require greater efforts for system design and control. There are reasons however why degeneracy may not necessarily incur such high design and control costs, which we discuss in detail in Whitacre et al. (2011) and mention here briefly. First, manufacturing systems are often integrated within socio-technical systems and in some circumstances humans may interfere to manage decisions that the control system is not yet capable to handling. Second, at least for the simulation conditions have been explored thus far, it appears that the desirable properties associated with degeneracy can arise through local and boundedly rational decision making and does not require centralised control. This is significant because technical components are becoming increasingly autonomous, (i.e., self-regulated, self-directed) and the effort for managing such systems might become reduced as a result of these technological advancements. Although design costs may increase from degeneracy (e.g. due to lost economies of scale), we have found that changes in only a small percentage of component designs can quickly lead to a networked buffering effect (Whitacre et al. 2011). Reductions in design costs may also arise from modularity-facilitated mass-customisation as briefly discussed in Sect. 5.

By exploring the concepts of degeneracy in several studies, we have found that the networked buffering property shown in Fig. 5 readily emerges whenever degeneracy is allowed to occur in a system that is forced to repeatedly adapt to novel changes in conditions. Importantly, this network-based flexibility does not need to be explicitly encouraged or planned for in order to arise in these simulations. Such findings are particularly relevant for systems that are forced to respond to novel conditions on a fairly regular basis. While no systems can adequately respond to all possible perturbations and all systems display degraded performance under conditions that are strongly disconnected from previous experiences (Whitacre 2010b), these simulations have found that high rates of adaptation combined with quantitatively higher robustness (Whitacre and Bender 2009) places degenerate systems at a considerable advantage in competitive environments that are not fully predictable (Whitacre et al. 2011).

One of the most important conclusions drawn from these studies of degeneracy and networked buffering is that only certain types of robustness will support system evolvability. For instance, in recent experiments, we have found degeneracy provides opportunities for design and operational novelty that are not simply random variations (Whitacre et al. 2011). Instead, the flexibility afforded by degeneracy can facilitate the emergence of new highly adaptive system configurations that are responses to new environmental and internal requirements. In other words, degeneracy may provide a means for what Kirschner and Gerhart (1998) refer to as "facilitated variation" and might be described as a constrained but more evolvable version of Kauffman's "Adjacent Possible" (Kauffmann 1993).

Together these findings are potentially significant to balancing the needs for stability and change in systems engineering contexts and they have been used to propose a mechanistic basis by which random variations can be transformed into useful innovations, as is discussed in Whitacre et al. (2011). We have argued that such findings should be relevant to evolution theory and the application of evolutionary principles to other domains. One domain where we have explored these ideas is in evolution-inspired optimisation. Below we describe the role that degeneracy can play in problem representation evolvability and then we focus the remainder of the article to explain how these basic findings can be transferred to the design of self-organising assembly systems.

3.2 Degeneracy and networked buffering in nature-inspired computing

Studies from several disciplines have attempted to determine those conditions that lead to the positive relationships observed in natural evolution between *mutational robustness* and *evolvability*. In computational intelligence, these issues relate directly to concepts of fitness landscape neutrality and the search for high-quality solutions. Fitness landscapes are used extensively in the field of combinatorial optimisation to describe the structural properties of the problem to be optimised. The fitness landscape results directly from the choice of representation as well as the choice of search operators. Subsequently, different representations lead to different fitness landscapes and hence to problems of different difficulty (see Rothlauf (2006) for an overview). Much research has focused on developing and analysing different problem representations. Inspired by earlier developments in theoretical biology, neutrality—the concept of mutations that do not affect system fitness—has been integrated into problem representations using various approaches such as polyploidy, i.e. introducing multiple copies of the same gene (see Banzhaf (1994), Yu and Miller (2001), Rothlauf and Goldberg (2003), Knowles and Watson (2003), Jin et al. (2009)). However, there are theoretical reasons as well as some experimental evidence to suggest that only particular representations of neutrality will support the discovery of novel adaptations.

When considering discrete local changes (mutations) in the decision variables of a single solution, the number of distinct accessible solutions is trivially constrained by the dimensionality of the solution space. Under these conditions, any increase in fitness neutrality—i.e. *mutational robustness*—will reduce the number of accessible alternative solution options (Jin and Trommler 2010). While more explorative/disruptive variation operators can increase solution options, nature almost always takes a different approach. In gene regulatory networks and other biological systems, mutational robustness often creates a neutral network that improves access to solution options over long periods of time, e.g. by drifting over neutral regions in a fitness landscape (Ciliberti et al. 2007). With solution options being a prerequisite of evolutionary adaptability, a strong case has been made that this positive correlation of mutational robustness and solution options is important to the evolvability of biological systems (Ciliberti et al. 2007; Whitacre and Bender 2010a; Whitacre 2010a).

Motivated by these developments in biology, some computational intelligence studies have investigated whether increasing neutrality (e.g. designing a many-to-one mapping between genotypes and phenotypes) influences the evolvability of a search process (Banzhaf 1994; Yu and Miller 2001; Rothlauf and Goldberg 2003; Knowles and Watson 2003; Jin et al. 2009). A common approach is to introduce genetic redundancy so that more than one copy of a gene performs the same function, i.e. genes that impact the fitness function in the same way (Banzhaf 1994; Yu and Miller 2001). Although early studies suggested that redundant forms of neutrality improve evolvability, others have questioned the utility of fitness landscape neutrality generated through redundant encodings (Knowles and Watson 2003; Whitacre and Bender 2009; Whitacre and Bender 2010a).

One problem with previous representation studies is that neutrality was introduced as a means for exploring a largely already determined fitness landscape and not as a property that arises as a consequence of development, i.e. genotype:phenotype mappings that are guided by feedback from an external environment. In biology, a considerable amount of neutrality (i.e. mutational robustness) is actively constructed through components that are partly interchangeable, i.e. conditionally compensatory or degenerate. This means that components might appear interchangeable in one environment or a particular genetic background, but lose this functional redundancy in other backgrounds, i.e. interoperability is context dependent. One important consequence is that different points in a neutral region within the fitness landscape will have mutational access to different phenotypes (Whitacre and Bender 2009; Whitacre and Bender 2010a). Recent Evolutionary Computation studies have found that phenotypes accessed in this way can have an adaptive significance in both static and dynamic environments (Whitacre 2010c; Whitacre 2010b; Whitacre et al. 2010). In the latter case, it was further shown that degeneracy enables the emergence of useful forms of genetic diversity in a population whereby few phenotypic differences are observed in a stable environment but many phenotypic variants can be revealed in the same population after a change in the environment (Whitacre 2010c). This conditional robustness in traits is analogous to a phenomena observed in natural populations known as cryptic genetic variation (Whitacre 2010c; Whitacre 2011).

3.3 Degeneracy in assembly systems

One of our primary claims in this article is that the same phenomena observed in the evolutionary simulations and Evolutionary Computation studies just discussed can be realised in a systems engineering context. For this to occur, it is an important requirement that agents are capable of functional plasticity and degeneracy.

In EAS and SOAS, modules have a fine granularity, which means that the functionalities of an industrial robot are broken down into many small modules (as opposed to only a few in a system with coarse granularity), as illustrated in Fig. 6. Finely granular modules may be defined at the level of tools or robotic axes; medium granularity is at robot or machine level, and coarse granularity at manufacturing cell level. Also conveyors may be divided into smaller or bigger units. Logically, the finer the granularity, the more varied the possibilities for recombining the modules to build different systems.

This means that, on the one hand, some modules can provide different functionalities in different contexts (functional plasticity), and on the other hand, several types of modules may provide the same functionality (functional redundancy). It all depends on the context and on how coalitions are composed.

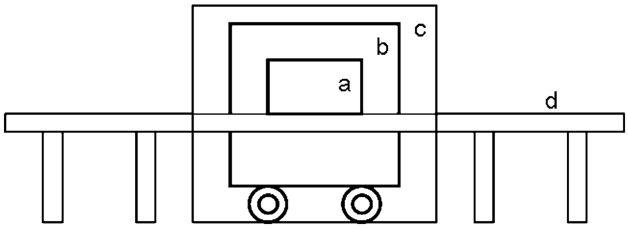

Fig. 6 Modules of varying granularity: *a* tool or axis, *b* robot or machine, *c* cell, *d* conveyor tables

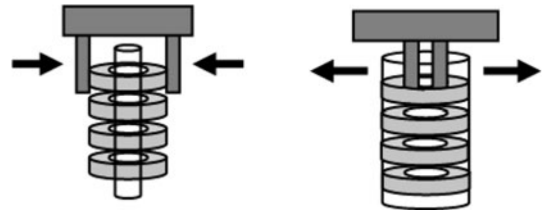

Fig. 8 The same gripper grabbing parts, once delivered on a stick (*left* the gripper fingers grab it from outside, which is the usual procedure) and once in a tube (*right* the gripper fingers grab the part from inside)

As an example of functional plasticity, a rotational axis and a vertical linear axis may provide a helicoidal movement (screwing movement), but they may also be part of a Scara-type robot, as illustrated in Fig. 7, composed of two rotations around a vertical axis with a vertical translation (thus requiring more partners that can provide an additional rotational axis). As for functional redundancy, a Scara-type functionality may be provided by a coalition of simpler modules, as explained, or it may be provided by an industrial Scara robot as a whole.

As another example, functional plasticity is demonstrated when a gripper, usually thought to grab a part between its fingers, lifts a part from inside (see Fig. 8), or closes its fingers to push apart. Similarly, functional redundancy means here that not only a finger gripper can handle a part, but also a vacuum gripper (using suction) or an electromagnetic gripper (in cases where the part contains a magnetic material).

Furthermore, a robot which is made for rapid pick&-place operations could incorporate a riveting tool to temporarily take over for a failing riveting robot; this would make the robot slower, but not otherwise disturb the system. This is at the same time functional plasticity and redundancy, because the robot can execute functions it was not intended to, and the required function can be achieved in more than one way.

3.4 Networked buffering in manufacturing and assembly

The concept of networked buffering appears to be broadly applicable to manufacturing and systems engineering.

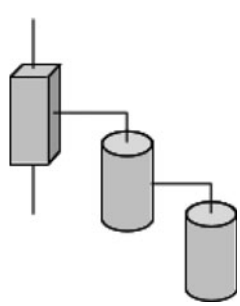

Fig. 7 Kinematic diagram of a Scara, where the cuboid stands for a vertical translation and the cylinders for rotations rotation around vertical axes

Consider any system comprising a set of multi-functional entities or agents which interact with each other (Whitacre and Bender 2010b). Each agent performs a finite number of tasks where the types of tasks performed are constrained by an agent's functional capabilities and by the environmental requirement for tasks ("requests"). A system's robustness is characterised by the ability to satisfy tasks under a variety of conditions. A new "condition" might bring about the failure or malfunctioning of some agents or a change in the spectrum of environmental requests. When a system has many agents that perform the same task then the loss of one agent can be compensated for by others, as can variations in the demands for that task. Stated differently, having an excess of functionally similar agents (excess system resources) provides a buffer against variations in task requests. While the utilisation of such local compensation appears to require ubiquitous excess resources, a buffering network of partly related agents can allow for a distributed response to local perturbations that utilises a small number of excess resources to respond to a variety of seemingly unrelated stresses (see description of Networked Buffering).

Besides manufacturing systems, which are discussed subsequently, examples where networked buffering could be applicable include self-deploying emergency task forces (Ulieru and Unland 2004), self-organising displays (Puviani et al. 2011), supply networks (Choi et al. 2001), fleets of transportation vehicles (Whitacre et al. 2011), as well as telecare for eldery persons and families (Camarinha-Matos and Afsarmanesh 2004). Each of these systems rely on a myriad of devices and/or persons which deliver services and interact on the basis of local rules and incomplete information. The system these agents form would be more stable under unexpected failures if other coalition members could substitute for failing ones and if the topology of these compensatory effects formed a connected network. Moreover, the flexibility in who does what means that the addition of only a few excess agent resources can confer a system with exceptional versatility, i.e. robustness can be achieved at relatively higher efficiencies.

The translation of this concept to assembly systems is immediate, given the previously described functional

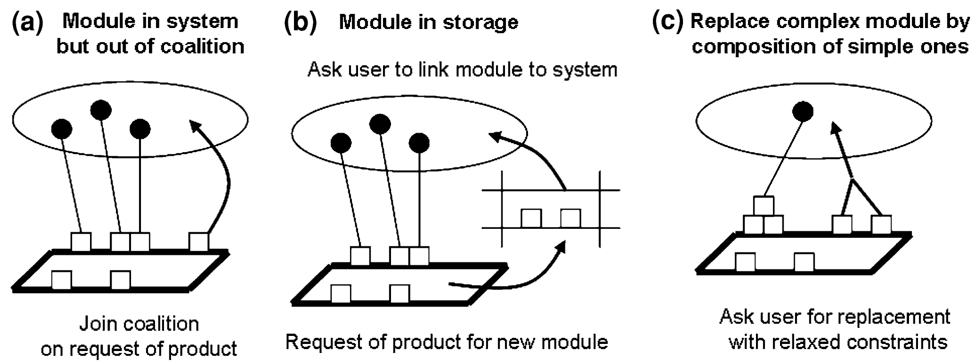

Fig. 9 Scenarios of networked buffering in a manufacturing coalition composed of modules

redundancy and functional plasticity together with the agents interactions and self-organisation described in SOAS. Once a module fails (or its neighbour notices that it is no longer responsive), either the module itself or one of its coalition partners looks for a replacement—either only of the failing module's functionality, or of the entire coalition's functionality, depending on the role of the failing module and the ease of replacing it. In many cases a user would need to confirm the action to be taken.

The same basic procedures would also apply when requirements change, that is, when modified or entirely different skills are requested. Either the coalition is able to provide it by adding or exchanging some modules, or a new coalition will be formed using the modules that are available, as illustrated in Fig. 9.

Networked buffering leads to a responsive, changeable system that is error-tolerant and robust against disturbances of many types. As an example, consider a scenario where an assembly system needs to cope with changing requirements. The assembly of a product may usually require a rivet, whereas a variant of the product requires a screw, as illustrated in Fig. 10. The product agent may therefore ask for a different process, which the robot setting the rivet may not be able to provide immediately. The robot (or the coalition of modules which compose the robot) may then check if another robot in the shop-floor layout is able to insert a screw, and if it is available to take over this task. Alternatively, the original robot may ask the user to replace the riveting tool by a helicoidal top-axis, and thus transform itself according to the requirements of the task at hand. More generally, having structural diversity amongst functionally similar agents provides greater flexibility in how a function is achieved and consequently a better chance of finding a way of satisfying a task requirement.

Another example of networked buffering is when a module fails; say a gripper becomes blocked, and resetting it does not resolve the problem. The agents must quickly find an alternative way of executing the task at question to avoid system down-time. An immediate solution may be provided by peers which, although also busy with other

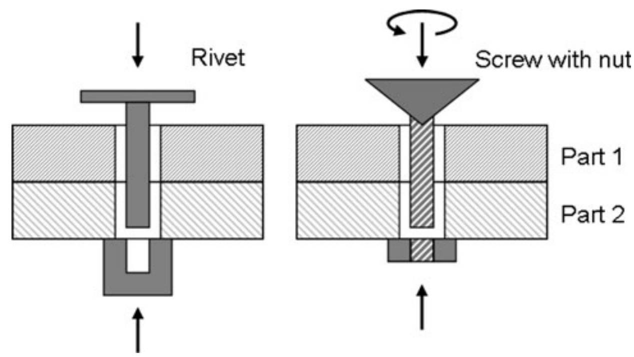

Fig. 10 Joining two parts by a rivet or a screw; parts 1 and 2 are not modified, but the process of joining them is changed, and thus the tools required by the process are different

tasks, have the required skills to temporarily take over the task in question. In parallel, the blocked gripper—or in case it is not responding any more, one of its coalition partners—will alert the user, who will take further actions. The failing gripper may, for instance, be replaced by a similar one, which will quickly be integrated into the existing agent coalition and take up its functionality.

4 Discussion: practical challenges in encouraging evolvability

Degeneracy has many positive effects; however, there may also be challenges to overcome before the benefits from these concepts can be realised within actual engineered systems. For instance, degeneracy increases a system's complexity, making global /centralised decision making more difficult. Because there are many components available which can achieve the same operation but in a potentially different way, it becomes more difficult for a central decision making authority to decide which component should execute which function at which moment. However, in multi-agent simulations it was found that distributed decision making with incomplete information can generate near optimal system performance, which indicates that there are many unique sequences of actions that can generate a beneficial adaptive response at the system level. The realisation of such properties in practice would however need to be evaluated in the context of each specific application domain (Whitacre et al. 2011).

If degeneracy is beneficial to engineered systems facing uncertain future conditions, then it would be important to consider how we might encourage such properties to arise. If we are starting with systems that were designed with an emphasis on reductionist principles and have an architecture that follows a decomposable hierarchy, it would not necessarily be obvious how one might proceed to transform such a system into an architecture with multi-functional agents and efficient buffering networks. In keeping with the evolutionary paradigm, it would also be important that each step taken in modifying the system can be incremental if needed and that each intermediate form constitutes a viable and competitive system. One plausible heuristic approach would be to start by focusing on individual components / devices / robots that are infrequently used and to consider how the roles of these components could be expanded, either by applying existing skills to new tasks or through small redesign to enable the fulfillment of related tasks. The general emphasis would be shifted from one where each component has a single task to one where component utility is defined by the ability to successfully take on any tasks possible, when and where they are needed. By looking at how agents can be better integrated with the system to satisfy its needs, degeneracy and networked buffering should naturally arise without explicit planning as was observed in the simulation studies mentioned earlier.

While degeneracy is easily achieved in biology (e.g. through gene duplication and divergence), the diversity of degenerate systems could present a cost barrier to the implementation of these ideas. On the other hand, trends towards modularity and mass customisation suggest that requirements of multi-functionality and degeneracy would not need to necessarily be costly and might already be straightforward to implement within several industries producing individual goods such as customised watches, mobile phones with many variants, personalised medicine, or custom-made furniture.

The cost of manufacturing systems can be broken down into two main parts: the cost of purchasing equipment, and the cost of its maintenance including reconfigurations. It is rather difficult to draw a precise comparison between a more traditional system and one with degeneracy because no one will be willing to build both systems. All that R&D scientists can usually do is compare system (reconfiguration) performances over hours or days and derive conclusions accordingly. However, such scenarios are unable to fully reveal the longer-term advantages of evolvable manufacturing systems. What can be concluded from limited experimentation and accumulated industry experience is that purchasing a set of smaller standard modules and combing them to build various systems according to upcoming needs is often considerably cheaper than acquiring custom-made specialised systems which are optimised to produce a specific set of products but are useless otherwise. It is also generally cheaper to perform maintenance on small interchangeable modules than to perform maintenance on a coarse-grained specialised system that needs to be taken off-line for the procedure. Finally, the cost of reconfiguring a system which is specifically made for seamless reconfiguration will generally be cheaper than re-engineering a custom-made system. Although such issues must be explored in greater detail and validated within specific manufacturing applications, we suspect that there is a growing number of manufacturing domains where the implementation of degeneracy principles should well be worth it.

Future research will explore design strategies for systematically introducing degeneracy into a system based on the types of localised and incomplete knowledge that one might expect to reasonably measure within an actual manufacturing system.

5 Concluding remarks

Accessing novelty In both biology and engineering, the discovery of an improved component design necessitates the exploration of new design variants. We have provides arguments based on genotype:phenotype mappings and conditional neutrality in fitness landscapes to explain how degeneracy enhances access to design options. In addition, degeneracy also enhances a system's access to design novelty because functionally redundant elements retain unique structural characteristics and thus have distinct options for how they can be modified.

Transforming novelty into innovation The availability of distinct design options is an important prerequisite for innovation, however new opportunities often come with new challenges. To transform a local novelty into a useful innovation, a system must be flexible (e.g. structurally, behaviorally) to accommodate and use a newly designed device effectively. For instance, design changes sometimes require new specifications for interaction, communication, operation, etc. However, a system must accommodate these new requirements without the loss of other important capabilities and without sacrificing the performance of other core system processes. In other words, the propensity to innovate is enhanced in systems that are robust in their core functions yet flexible in how those functions are carried out (Kirschner and Gerhart 1998).

Facilitating unexpected opportunities Because design novelty is not predictable, the flexibility needed to exploit design novelty cannot be pre-specified based on the anticipation of future design changes, either. To support innovation in systems engineering, it appears that this robust yet flexible behaviour would need to be a property that is pervasive throughout the system. Yet within the wider context of a system's development—where each incremental design change involves a boundedly rational and ultimately myopic decision—it also seems that this flexibility must be an emergent property that can readily emerge without foresight. These challenges are generic to system evolvability and are equally relevant in understanding the evolution of system behavior towards changing external requirements or the development of robust responses to unexpected failures.

This article has described how a common biological property known as degeneracy can lead to the emergence of a highly flexible and highly adaptive system. The principles are conceptually simple and could prove broadly relevant to the behavior of biological, social, and technological systems. Here we have outlined how these principles can be directly translated into an assembly system with the result being a highly adaptive and agile manufacturing system. The realisation of degeneracy may come with the investment cost of numerous robotic modules, however, the gained agility and the ability to let the system co-evolve with requirements is expected to compensate for this, and will make the principles discussed in this article economically sensible in the near future.

References

Al-Safi Y, Vyatkin V (2010) Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing. Robot Comput Integr Manuf 26(4):381–391

Ashby W (1956) An introduction to cybernetics. Chapman & Hall, London

Banzhaf W (1994) Genotype-phenotype-mapping and neutral variation-a case study in genetic programming. In: Parallel problem solving from nature III, vol 866. Springer, Berlin, pp 322–332

Barata J (2005) Coalition based approach for shopfloor agility. Edições Orion, Amadora

Barata J, Camarinha-Matos L, Onori M (2005) A multiagent based control approach for evolvable assembly systems. In: 3rd IEEE international conference on industrial informatics (INDIN). Perth, Australia, pp 478–483

Berry G, Boudol G (1998) The chemical abstract machine. Theoret Comput Sci 96(1):217–248

Brueckner S (2000) Return from the ant—synthetic ecosystems for manufacturing control. PhD thesis, Institute of Computer Science. Humboldt-University, Berlin, Germany

Bussmann S, Schild K (2000) Self-organizing manufacturing control: an industrial application of agent technology. In: 4th IEEE international conference on multi-agent systems, Boston, MA, USA, pp 87–94

Buzacott JA, Shanthikumar JG (1980) Models for understanding flexible manufacturing systems. AIIE Trans 12(4):339–350

Camarinha-Matos LM, Afsarmanesh H (2004) A multi-agent based infrastructure to support virtual communities in elderly care. Int J Netw Virtual Organ 2(3):246–266

Choi T, Dooley K, Rungtusanatham M (2001) Supply networks and complex adaptive systems: control versus emergence. Oper Manag 19:351–366

Ciliberti S, Martin OC, Wagner A (2007) Innovation and robustness in complex regulatory gene networks. Proc Natl Acad Sci USA 104(34):13591–13596

Edelman G, Gally J (2001) Degeneracy and complexity in biological systems. Proc Natl Acad Sci USA 98(24):13763–13768

ElMaraghy H (2006) Flexible and reconfigurable manufacturing systems paradigms. Int J Flex Manuf Syst 17(4):261–276

ElMaraghy H, AlGeddawy T, Azab A (2008) Modelling evolution in manufacturing: a biological analogy. CIRP Ann Manuf Technol 57:467–472

Frei R (2010) Self-organisation in evolvable assembly systems. PhD thesis, Department of Electrical Engineering, Faculty of Science and Technology, Universidade Nova de Lisboa, Portugal

Frei R, Barata J (2010) Distributed systems—from natural to engineered: three phases of inspiration by nature. Int J Bio Inspired Comput 2(3/4):258–270

Frei R, Barata J, Di Marzo Serugendo G (2007) A complexity theory approach to evolvable production systems. In: Sapaty P, Filipe J (eds) International Workshop on Multi-Agent Robotic Systems (MARS), Angers, France. INSTICC Press, pp 44–53

Frei R, Di Marzo Serugendo G (2011a) Advances in complexity engineering. Int J Bio Inspired Comput 3(4):199–212

Frei R, Di Marzo Serugendo G (2011b) Concepts in complexity engineering. Int J Bio Inspired Comput 3(2):123–139

Frei R, Di Marzo Serugendo G (2011c) Self-organising assembly systems. IEEE Trans Syst Man Cybern C Appl Rev 41(6): 885–897

Frei R, Di Marzo Serugendo G, Barata J (2008) Designing self-organization for evolvable assembly systems. In: IEEE international conference on self-adaptive and self-organizing systems (SASO), Venice, Italy, pp 97–106

Frei R, Di Marzo Serugendo G, Serbanuta T (2010) Ambient intelligence in self-organising assembly systems using the chemical reaction model. J Ambient Intell Humaniz Comput 1(3): 163–184

Gaugel T, Bengel M, Malthan D (2004) Building a mini-assembly system from a technology construction kit. Assem Autom 24(1):43–48

Gowdy J, Rizzi A (1999) Programming in the architecture for agile assembly. In: IEEE international conference on robotics and automation (ICRA), vol 4, pp 3103–3108

Hanisch C, Munz G (2008) Evolvability and the intangibles. Assem Autom 28(3):194–199

Hollis R, Rizzi A, Brown H, Quaid A, Butler Z (2003) Toward a second-generation minifactory for precision assembly. In: International advances robotics program workshop on microrobots, micromachines and microsystems, Moscow, Russia

Jin Y, Gruna R, Paenke I, Sendhoff B (2009) Evolutionary multi-objective optimization of robustness and innovation in redundant genetic representations. In: IEEE symposium on computational intelligence in multi-criteria decision making, Nashville, USA, pp 38–42

Jin Y, Trommler J (2010) A fitness-independent evolvability measure for evolutionary developmental systems. In: IEEE symposium on computational intelligence in bioinformatics and computational biology (CIBCB), Montreal, Canada, pp 1–8

Katz R (2007) Design principles for reconfigurable machines. Int J Adv Manuf Technol 34:430–439

Kauffmann S (1993) The origins of order: self-organization and selection in evolution. Oxford University Press, Oxford, United Kingdom

Kaula R (1998) A modular approach toward flexible manufacturing. Integr Manuf Syst 9(2):77–86

Kephart J, Chess D (2003) The vision of autonomic computing. IEEE Comput 36(1):41–50

Kirschner M, Gerhart J (1998) Evolvability. Proc Natl Acad Sci USA 95(15):8420–8427

Knowles JD, Watson RA (2003) On the utility of redundant encodings in mutation-based evolutionary search. LNCS, pp 88–98

Koestler A (1967) The ghost in the machine. Hutchinson, London

Koren Y, Heisel U, Jovane F, Moriwaki T, Pritchow G, Ulsoy A, Van Brussel H (1999) Reconfigurable manufacturing systems. CIRP Ann Manuf Technol 48(2):6–12

Kume S, Rizzi A (2001) A high-performance network infrastructure and protocols for distributed automation. In: IEEE international conference on robotics and automation (ICRA), vol 3, Seoul, Korea, pp 3121–3126

Kurakin A (2009) Scale-free flow of life: on the biology, economics, and physics of the cell. Theoret Biol Med Modell 6(6). doi:10.1186/1742-4682-6-6

Leitão P (2004) An agile and adaptive holonic architecture for manufacturing control. PhD thesis, Department of Electrical Engineering, Polytechnic Institute of Bragança, Portugal

Leitão P (2008) A bio-inspired solution for manufacturing control systems. In: IFIP international federation for information processing, vol 266, Springer, Boston, pp 303–314

Leitao P, Restivo F (2008) Implementation of a holonic control system in a flexible manufacturing system. Trans Syst Man Cybern C Appl Rev 38(5):699–709

Marik, V, Vyatkin, V, Colombo, A (eds) (2007) Holonic and multi-agent systems for manufacturing. Springer, Heidelberg

Mehrabi M, Ulsoy A, Koren Y (2000) Reconfigurable manufacturing systems: key to future manufacturing. J Intell Manuf 11:403–419

Niemeyer C (2006) Pick and place in a minifactory environment. PhD thesis, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland

Onori M (2002) Evolvable assembly systems—a new paradigm? In: 33rd international symposium on robotics (ISR), pp 617–621, Stockholm, Sweden

Onori M, Groendahl P (1998) MARK III, a new approach to high-variant, medium-volume flexible automatic assembly cells. Robotica, Special Issue 16(3):357–368

Puviani M, Di Marzo Serugendo G, Frei R, Cabri G (2011) A method fragments approach to methodologies for engineering self-organising systems. ACM Trans Auton Adapt Syst (TAAS)

Rizzi A, Gowdy J, Hollis R (1997) Agile assembly architecture: an agent based approach to modular precision assembly systems. In: International conference on robotics and automation (ICRA). Albuquerque, New Mexico, pp 1511–1516

Rothlauf F (2006) Representations for genetic and evolutionary algorithms. 2nd edn. Springer, Berlin

Rothlauf F, Goldberg DE (2003) Redundant representations in evolutionary computation. Evol Comput 11(4):381–415

Semere D, Onori M, Maffei A, Adamietz R (2008) Evolvable assembly systems: coping with variations through evolution. Assem Autom 28(2):126–133

Shen W, Hao Q, Yoon HJ, Norrie DH (2006) Applications of agent-based systems in intelligent manufacturing: an updated review. Adv Eng Inf 20:415–431

Ueda K (2006) Emergent synthesis approaches to biological manufacturing systems. In: 3rd International CIRP conference on digital enterprise technology (DET), Keynote paper, Setubal, Portugal

Ulieru M, Unland R (2004) Emergent e-logistics infrastructure for timely emergency response management. In: Di Marzo Serugendo G, Karageorgos A, Rana O, Zambonelli F (eds) Engineering self-organizing systems: nature inspired approaches to software engineering, vol 2977 of LNAI, Springer, Berlin, pp 139–156

Valckenaers H, Van Brussel P (2005) Holonic manufacturing execution systems. CIRP Ann Manuf Technol 54(1):427–432

Vallée M, Kaindl H, Merdan M, Lepuschitz W, Arnautovic E, Vrba P (2009) An automation agent architecture with a reflective world model in manufacturing systems. In: IEEE international conference on systems, man and cybernetics (SMC). IEEE Press, pp 305–310

Vrba P (2003) MAST: manufacturing agent simulation tool. In: IEEE international conference on emerging technologies and factory automation (ETFA). Lisbon, Portugal, pp 282–287

Whitacre J (2010a) Degeneracy: a link between evolvability, robustness and complexity in biological systems. Theoret Biol Med Modell 7(6)

Whitacre JM (2010b) Evolution-inspired approaches for engineering emergent robustness in an uncertain dynamic world. In: Conference on artificial life XII. Odense, Denmark, pp 559–560

Whitacre JM (2010c) Genetic and environment-induced innovation: complementary pathways to adaptive change that are facilitated by degeneracy in multi-agent systems. In: Conference on artificial life XII. Odense, Denmark, pp 431–432

Whitacre J (2011) Genetic and environment-induced pathways to innovation: on the possibility of a universal relationship between robustness and adaptation in complex biological systems. Evol Ecol 25:(5):965–975

Whitacre JM, Bender A (2009) Degenerate neutrality creates evolvable fitness landscapes. In: WorldComp-2009, Las Vegas, Nevada

Whitacre J, Bender A (2010a) Degeneracy: a design principle for achieving robustness and evolvability. J Theoret Biol 263(1): 143–153

Whitacre J, Bender A (2010b) Networked buffering: a basic mechanism for distributed robustness in complex adaptive systems. Theoret Biol Med Modell 7(20). doi:10.1186/1742-4682-7-20

Whitacre JM, Rohlfshagen P, Yao X, Bender A (2010) The role of degenerate robustness in the evolvability of multi-agent systems in dynamic environments. In: 11th International conference on parallel problem solving from nature (PPSN). Krakow, Poland, pp 284–293

Whitacre JM, Rohlfshagen P, Bender A, Yao X (2011) Evolutionary mechanics: new engineering principles for the emergence of flexibility in a dynamic and uncertain world. Nat Comput. doi: 10.1007/s11047-011-9294-5

Wuertz R (ed) (2008) Organic computing. Understanding complex systems. Springer, Berlin

Yu T, Miller JF (2001) Neutrality and the evolvability of boolean function landscape. In: Proceedings of the 4th European conference on genetic programming. Springer, London, pp 204–217