15th Triennial World Congress, Barcelona, Spain

ON THE DESIGN OF COMPLEX EMERGENT SYSTEMS

Paul Valckenaers, Hendrik Van Brussel, Olaf Bochmann, Hadeli, Martin Kollingbaum

K.U.Leuven, Department of Mechanical Engineering,
Celestijnenlaan 300 B, B-3001 Belgium
E-mail: Paul.Valckenaers@mech.kuleuven.ac.be

Abstract: Many useful manmade systems in this world are extremely complex; a typical example is a large infrastructure. No design team ever invents these artefacts because they are too complex. The artefacts are made by combining existing elements (legacy) and by building new subsystems without explicit and comprehensive up-front coordination. To a large extent, these complex systems emerge and evolve. Experience shows that designers often fail to develop artefacts that, when combined, facilitate the emergence of effective and efficient emerging systems. This paper formally elaborates the mechanism behind this phenomenon, and proposes design principles for the design of emergent systems' components. These principles are discussed and illustrated. *Copyright © 2002 IFAC.*

Keywords: Complex systems, Constraints, Co-ordination, Design systems, Integration, Manufacturing systems.

1. INTRODUCTION

Both design teams and individual designers face limitations concerning the complexity of the artefacts that they may develop. When building complex systems, designers develop subsystems that are combined into a larger system. The resulting system exhibits an emergent behaviour that was never explicitly planned or conceived by these designers simply because its complexity exceeds their mental capabilities.

Many industrial design teams, designing for instance automobiles, remain largely in control of what their artefacts will be. In contrast, many of the most valuable artefacts in a modern human society, especially infrastructures, simply are too complex to be conceived explicitly by humans. They emerge by the combination and integration of simpler systems, which form their constituents. Unfortunately, the resulting emergent behaviour is too often characterised by poor performance, missed opportunities and the inability to serve smaller user communities.

This paper discusses the fundamental nature of the above issue. What causes the difficulties occurring when smaller systems are combined and integrated into a larger and more complex system? What can be done to remedy the integration/emergence problems, which are observed in reality?

This research builds on the ideas of Herbert Simon (1990) as well as insights discussed in (Waldrop, 1992). Waldrop gives a readable and concise overview of relevant developments in the domain of complex systems.

Simon emphasises the issue of the fundamental difference between analytical/observing sciences and synthesising/engineering/designing/creating sciences. Every artefact design is, to a significant extent, arbitrary; there are numerous ways in which a design problem can be solved correctly. In contrast, the different manners in which the laws of physics can be described only differ in very superficial ways (i.e. the symbols that are used). In making this observation, Simon touches the core issue: a scientific understanding of synthesis/design is lacking.

Furthermore, this paper complements more classical work on design (Nam Suh, 1988) aimed at design tasks where the design team remains largely in control and typically designs for a short time horizon. Such work typically builds on the decomposition of functional requirements and top-down design of solutions. In contrast, this paper aims at the design of more long-lived artefacts that are part of an emerging overall system. Such artefacts are typically considered as part of the given technology base in the shorter-term design situation.

The paper starts with a formal model for problems and their solutions. Next, it models the integration issue and the problem of facilitating a desirable emergent behaviour achieving good performance, effectiveness and efficiency. Ensuing design principles are presented and their application is shortly illustrated.

2. PROBLEMS AND SOLUTIONS[1]

This section formally addresses what constitutes a problem and its solution(s). Note that the research, on which this paper reports, addresses coordination and control of manufacturing systems as its main application domain.

1.1 Problems

A (basic) problem is defined as follows:

> A problem P is a constraint on the state space $\mathbf{U}$ of the universe U, defining a set $\mathbf{P} \subseteq \mathbf{U}$.

When U is the world in which we live, $\mathbf{U}$ is an infinite state space. This universe is subject to physical laws (constraints), which limit the number of states that are reachable. Every state $\mathbf{u} \in \mathbf{U}$ has a time coordinate $\mathbf{t_u} \in \Re$. The laws imply that there is exactly one reachable state $\mathbf{u}$ for every $\mathbf{t_u} \leq \mathbf{t_{now}}$.

The universe U describes a trajectory through its state space as time progresses. This trajectory is defined for states up to $\mathbf{t_{now}}$. It consists of the states in which the universe has been in the past. The future trajectory is only partially defined.

This future trajectory is constrained by physical laws, possibly including stochastic aspects, and is affected by the actions of the human and other entities in this world. These actions affect the choice of the successor states of the current state corresponding to $\mathbf{t_{now}}$. Normally, any significant impact on the trajectory requires sustained action during a substantial amount of time. A problem P is solvable by an agent (human or otherwise) if s/he is able to make this trajectory stay within the given subset $\mathbf{P}$.

Remark 1. In our world, there exist two major types of problems. First, there are the *one-shot* problems, which require the state of the universe to comply once with a constraint at some point in time. The problem specification does not care about the states before and after. An example is to deliver in time some quantity of goods with sufficient quality. Such problems often are agreements between humans to coordinate their interactions.

The second type of problems consists of *going concerns*. Going concerns require that the trajectory of the universe complies with requirements that span a complete time window, typically starting from $\mathbf{t_{now}}$. For instance, a coordination and control system must keep its underlying system in a safe state (no casualties…). Note that there exist many problem-solving technologies that cannot handle going concerns (e.g. database query engines, most optimisation software).

Inherently, most real-life problems are going concerns, where one-shot problems often are artificial problems. Accordingly, coordination and control technology normally addresses going concerns, possibly using one-shot problem solvers as subsystems.

Remark 2. The above defines a basic problem as a set of constraints with which a solution must comply. In reality, there also exist optimisation problems. This can be modelled as a set of tuples, where each tuple consist of a basic problem and a valuation, where the overall problem is to optimise this valuation. This valuation defines the value/benefit/… of solutions to the corresponding basic problem. Such a valuation can be single-valued and fully ordered, but it can also reflect multi-criteria optimisation problems… Since the remainder of the discussion does not require this extension, it is not elaborated further.

Remark 3. The above definition could be applied, in principle, to any kind of universe that can be modelled with a state space. However, this document only considers the macroscopic world in which we live.

1.1 Solutions

A solution to a basic problem is defined as follows:

> A solution S of a problem P is a constraint on the state space $\mathbf{U}$ of the universe U defining a set $\mathbf{S}$,

---

[1] Disclaimer: This paragraph models the world in which we live for the purpose of the communicating a view on problem solving in the present context only. It makes no claim to correctness and completeness in a philosophical sense or within the physical sciences.

where $\mathbf{S} \subseteq \mathbf{P} \subseteq \mathbf{U}$ and
$\forall\, \mathbf{t} \in \Re, \exists\, \mathbf{s} \in \mathbf{S}: \mathbf{t} = \text{timeCoordinateOf}(\mathbf{s})$.

Agents (human or otherwise, intentionally or unintentionally) solve a given problem P when they confine, through their actions, the trajectory of the universe U to the corresponding subset $\mathbf{P}$. Therefore, their actions — combined with the laws of the universe — correspond to constraining the state of the universe to a subset $\mathbf{S}$ of $\mathbf{P}$. $\mathbf{S}$ cannot be empty; it will always have one state for every time coordinate. If S fails to comply with this condition, the agents failed to solve the problem.

Typically, $\mathbf{S}$ and $\mathbf{P}$ will differ significantly concerning the states with a time coordinate that is smaller than, equal to or marginally larger than $\mathbf{t}_{\mathbf{now}}$. The problem P only cares about what is needed/useful/… Therefore, it allows as many states as possible as long as the choice amongst them does not matter — note that this discussion only considers intrinsic aspects and makes abstraction of issues concerning the explicit specification of constraints.

In contrast, the solution S is embedded in the universe, which allows only a single state for every time coordinate in the past (including the present) and imposes severe limitations on what states can be reached in the immediate future from the current state. In other words, $\mathbf{S}$ will be significantly smaller than $\mathbf{P}$, especially concerning states close to the present time and older. Therefore, problem-solving agents have to make choices whenever deadlines approach.

In real life, a problem solving activity consists of a sequence of actions over time. Using the above definition, such sequence of actions corresponds to a sequence of solutions $S_1, S_2, \ldots S_{end}$ that solve P, where $S_{end} \subset \ldots \subset S_2 \subset S_1 \subset P$. This reflects that the agents make more and more choices as time progresses in order to solve the problem and comply with the laws of the universe.

Moreover, any solution for a problem that constrains future states needs to extend itself into this future. In simple cases and for short periods, the laws of the universe may be sufficient. However, some incarnation of an autonomous system will be necessary to ensure a full and final solution in most cases. In real life, human beings play an essential part in providing such autonomous behaviour.

The issue of autonomy is however not the most relevant one for the purpose of this manuscript. The introduction of constraints by the solution is the key issue.

An example of the introduction of constraints is the design of a railway system to solve transportation problems. When the actual instance of usage approaches, the designers have to make more and more choices. For instance, they must select a specific value for the space in between the rails. In fact, the whole problem solving activity can be seen as a sequence of design choices, starting from the selection of a rail-based system over other possibilities.

3. EMERGENT SOLUTIONS

When solving real-life problems, agents — human or otherwise — start from existing subsolutions, technology, and infrastructure. In addition, the individual agents provide parts of the overall solutions. These parts, existing or new, are brought together, integrated as far as possible and an overall solution emerges. In this situation, the individual agents face a high level of uncertainty about the other parts with which their part needs to cooperate.

Typically, the agents contribute to the solution of many problems over time (e.g. a section of a transportation infrastructure is used in solving numerous individual transportation problems). Likewise, the agents' contribution is used to solve problems unknown at the time it is created, and it must be combined with contributions from other agents. Many of these other contributions are developed independently such that the individual agent has limited opportunities to coordinate its contribution with the others. Some of these other contributions only emerge after the creation of the contribution of such individual agent.

Formally, agent **x** solves problem P through solution S_p. The other agents solve problem Q through solution S_q. This results in the state sets $\mathbf{S_p} \subseteq \mathbf{P} \subseteq \mathbf{U}$ and $\mathbf{S_q} \subseteq \mathbf{Q} \subseteq \mathbf{U}$. For instance, agent **x** has constructed the railway system in France to answer the need for transportation. The other agents implemented similar railway systems in the remainder of Europe. These systems are defined as to include the human organisation that operates, maintains and adapts/expands the hardware therein.

The ambition of research on emergent systems design is to solve large and complex problems by solutions that emerge when sub-systems/sub-solutions are combined. In the example, transportation all over Europe is to be solved by integration of the national railway systems: solve problem T, where $\mathbf{T} \subseteq \mathbf{P}$ and $\mathbf{T} \subseteq \mathbf{Q}$. Indeed, society is unable to duplicate the effort of developing their national railway systems to provide an international one. Instead, it must reuse the existing systems, including their ability to adapt, among others, to create the international connections amongst the national systems and obtain a system that transports goods and passengers across the borders in Europe.

The main problem with the creation of such solutions is that the integration fails to deliver good performance. In the example, it is relatively easy to provide international transport at a much-reduced level of service; goods and passengers need to change trains at almost every border. Higher levels of service require significant efforts: locomotives that support multiple power supply systems, railway equipment supporting variable width of the railway tracks... The sub-solutions, which were developed independently, have made mutually incompatible design decisions and these decisions have accumulated significant inertia (i.e. it has become costly to undo these decisions). Formally:

$$\exists\, \mathbf{t} \in \Re,\ \forall\, \mathbf{s} \in \mathbf{S_p} \cap \mathbf{S_q} \cap \mathbf{T}: \quad \mathbf{t} \neq \text{timeCoordinateOf}(\mathbf{s})$$

In practice, a solution for T cannot readily reuse the (partial) solutions offered by the individual agents without undoing a lot of design choices. Typically, society only receives a reduced level of service (i.e. solutions to easier problems), and will only gradually outgrow the old designs when technology progresses sufficiently to introduce a new solution from scratch (e.g. high-speed trains).

The key issue is the introduction of constraints, by the agents, that are absent in the corresponding problem and that may cause future integration problems. More precisely, it is the accumulated inertia — i.e. the effort needed to undo such harmful design decision — that constitutes the problem. This issue is mostly relevant for the design of infrastructure elements, which have to last a long time and have impact on future service levels.

4. DESIGN PRINCIPLES

From the above, it becomes clear that the designing of subsystems for emergent solutions imposes its own requirements on a design activity. In particular, the problem-solving agent must design a solution S_p capable of surviving in an uncertain environment concerning its future.

Formally, such uncertain environment corresponds to $\wp$, a set of subsets of the state space $\mathbf{U}$. The actual future will offer an intersection of members of $\wp$ as the space that is available for S_p to contribute its part to the overall solution. Some members of $\wp$ correspond to the constraints imposed by the future problems for which solution S_p may be part of the overall emergent solution. Alternatively, members of $\wp$ correspond to the constraints imposed by other candidate subsolutions that may contribute to solving the bigger problem at hand. A designer of solution S_p does not know which members of $\wp$ will be present, but must avoid conflicts with any constraints that might be present.

In this context, design decisions can introduce two types of constraints: stable and unstable. Stable constraints will be present in all conceivable future situations within the scope of S_p. For instance, an agent may assume that power supply will be 240V/50Hz or 130V/60Hz when designing an electrical appliance. Formally, no member of $\wp$ will be in conflict with the stable constraint.

In contrast, unstable constraints represent conflicts with some members of $\wp$. Design decisions that introduce unstable constraints reduce the future capability of the solution. For instance, a software designer may assume that two digits suffice to represent the year in date information. Likewise, the designer of a computer operating system may decide to limit application memory to 640 kilobytes. Also, the designer of a rocket inertial navigation program may choose to limit the range of the acceleration supported to the limits of the current rocket, causing the crash of the first rocket of the next generation when the constraint remained undetected.

Based on the above, a number of design principles emerge. Note that these principles apply when designing lasting artefacts, like infrastructures, and not for the short-time solutions for the immediate future.

1. *Problem solvers must avoid introducing potentially harmful constraints.*

This guideline is twofold. First, it advises against the introduction of (any) constraints. As shown above, this cannot be avoided for the immediate future. However, concerning problem solving in a slightly more distant future, (autonomous) systems implementing low and late commitment strategies avoid the introduction of constraints. This is a relatively well-known fact, and it is widely adhered to. Unfortunately, costs and complexity of such solutions seriously restrict the applicability of this design principle. For instance, building locomotives and railway wagons for variable-width tracks is prohibitively expensive/complex.

Second, the guideline encourages introducing constraints unlikely to cause future conflicts; these are called stable constraints or decisions in this manuscript. In short, the introduction of stable constraints simply reflects the fact that the constraint is already present in the environment; a stable design decision preserves and reflects the scope of the problem domain.

2. *Problem solvers must avoid/reduce the inertia build-up for potentially harmful constraints.*

Again, this guideline is twofold. First, designers are encouraged to implement low-inertia versions of their solutions. This is a well-known principle; specific technologies have been developed to support this: CAD, CAE, …

Second, designers must avoid reinforcing unstable constraints introduced by earlier unstable design decisions. Designers cannot avoid introducing unstable constraints, e.g. selecting a width for the railway track, when the time for implementing the solution approaches. However, the fact that this width need to be fixed before the first locomotive can be build does not imply that this width needs to be fixed in other situations, e.g. being hard-coded in a piece of embedded software, where the cost of keeping your options open is negligible. Whereas stable constraints can be reinforced without problems — the environment or general scope of the problem domain already imposes such constraints — an earlier unstable constraint gains inertia every time another design decision introduces it again. For instance, it is a well-known problem that highly automated production facilities are hard to adapt because the control systems (= software) reinforce the existing way of operating and require significant maintenance for every change in the system. Ironically, the technology that is seen to be extremely flexible causes the rigidity (because of its poor design).

In summary, the novel principles for the designers are: (1) designers must prefer stable design decisions and (2) earlier unstable design decisions are no justification for later decisions imposing the same constraint(s). The first design principle avoids the introduction of new constraints. The second avoids the build-up of inertia for unstable constraints that were introduced earlier; every unstable design decision must be justifiable itself. Earlier unstable design decisions only affect which later unstable design decisions are taken (i.e. compatible with these earlier ones), not whether a later unstable decision will be taken at all. Next to these two items, some better-known principles — low and late commitment, autonomous systems, and low inertia implementations — remain valid. Note that low commitment includes the ability to fall back on interim designs at stages before unstable constraints are introduced.

5. APPLICATION OF THE DESIGN PRINCIPLES

The application of the above design principles differs significantly across the existing design areas. For instance, designers of macroscopic mechanical structures typically incur severe cost penalties when they postpone the introduction of a design choice. For instance, in building construction the distance between the supports, which carry the load of the roof, has a major impact on the cost of a building. And, a building design that postpones how the building will be used benefits from having large spaces inside without those obstructing supports. Likewise, the cost of the night train between Paris and Madrid, capable of running over variable-width tracks, is prohibitively high for more widespread application.

As mentioned above, the research focuses on coordination and control of manufacturing systems. This area is better suited for application of the above design principles. More precisely, the underlying systems, which are being controlled, constitute a rich source of stable constraints for the designers of their coordination and control systems.

For a coordination and control system, the underlying system is given. Their designers are not responsible for the performance that emerges when the components and subsystems in this underlying system are combined and integrated. If the underlying system is highly sub-optimal, like the railway systems with different track widths, the coordination and control system must make the best possible use of the situation. It must not try to remedy this situation. It takes whatever emerges as it is, and provided appropriate coordination and control.

Actually, it is desirable that the coordination and control elements make minimal assumptions about the underlying system as a whole since this makes them suitable components in an overall system that emerges by their combination and integration. Indeed, they will not cause problems by imposing constraints with which the underlying hardware must comply.

The key point is that the underlying system is part of reality, and in reality everything is consistent, if not necessarily as desired. As a consequence, coordination and control system elements that reflect parts of the underlying system, avoiding potentially harmful design choices beyond this mirroring of some part of the underlying reality, will not cause problems when they are integrated and combined into an overall coordination and control system.

The above implies the research question whether and how it is possible to design a coordination and control mainly consisting of components that reflect parts of the underlying reality. Moreover, the development and maintenance costs for these components must be recoverable by their user market; these components must be employed in sufficient quantity for sufficient time, in function of their complexity.

Research on multi-agent coordination and control in manufacturing reveals this to be possible at least in a research context. In such control systems, resource agents reflect manufacturing resources like conveyor belts and machining stations (Valckenaers, 2001). Order agents manage the work in progress, each reflecting a product instance being made. Product agents reflect the recipes of the products, informing order agents what their processing options are and providing the resource agents with technical data. The agents interact to make relevant information available at the proper places to guide coordination and control decisions.

Consider, as an example, the following basic function in a coordination and control system: when a product instance must make a routing decision at a crossing in the manufacturing plant, the corresponding order agent needs to discover which routing options support the required processing steps.

The challenge is to expose the order agent and the resource agents neither to the complexity of the plant topology, nor to the status/availability of the processing equipment. A possible solution goes as follows:

1. Resource agents at the exits of the factory regularly create mobile agents that travel upstream, collecting information about the processing units on their way. At every routing point, these agents drop the information, which they collected so far.

2. Resource agents of processing units inform the passing mobile agents about their processing capabilities. These resource agents need no knowledge about the plant topology or about their potential visitors.

3. Resource agents corresponding to transportation resources are capable of routing the mobile agents from their exits to their entries. When a resource has multiple entries, the mobile agents clone themselves accordingly. Note that the mobile agents discover the routing topology as they go. They do not need/use explicit information on the plant topology. Likewise, they discover the status of the routing devices, and avoid propagating themselves over links that are out-of-order. Conversely, the resource agents only know their own resource and their immediate neighbours.

4. The order agents consult the information attached to the resource agents of resources that incorporate a routing choice to detect which options are offering the required processing step. The choice amongst the valid options is based on mechanisms that are outside the scope of this paper.

5. The information attached to the routing devices is discarded after a finite amount of time. This mechanism ensures that stale information is forgotten within a short delay. The agents corresponding to the exits of the factory create the mobile agents at a sufficiently high frequency such that the required information is continuously available and new opportunities are discovered rapidly.

Mechanisms, similar to the above example, to balance loads, create batches, avoid deadlocks… are needed to implement a comprehensive coordination and control system. Their implementation includes technical details like hub limits to avoid infinite cycling of these mobile agents. Their discussion is outside the scope of this paper. Note however that the core of such a coordination and control system can be developed from agents reflecting limited parts of the underlying system. The agents can be combined without major conflicts. In other words, the individual agents have very limited exposure to the characteristics of the overall system. The agents do their job regardless of what system they are part of; instead, the agents are experts at the part of reality that they reflect (e.g. a conveyor belt).

6. CONCLUSIONS

The really complex human-made systems in this world emerge when their subsystems are combined into the overall system. No designer or design team is able to conceive what these complex systems will be, if only because the components are designed before the utility of the overall system is discovered (national railway systems in Europe were designed before anybody considered a European railway system).

The performance of such emerging systems often is very poor. This manuscript analyses why this happens and derives design guidelines for developing elements of emergent systems that are more likely to succeed, if only because designers become more aware when they take risky decisions.

There are limitations on how suitable elements can be made for incorporation in an emergent design. These limitations differ across application areas. Coordination and control systems are highly suitable because the underlying system provides a rich source of stable constraints, which enable designers to take safe decisions first, and limit the potentially harmful choices to the final phases of the development, such that they are easily undone. This succinctly illustrated by a part of a multi-agent coordination and control system design.

REFERENCES

Simon, H. A. (1990). *The sciences of the artificial,* 2nd ed., MIT Press, Cambridge Mass.

Waldrop, M. (1992). *COMPLEXITY, the emerging science at the edge of order and chaos,* VIKING, Penguin group, London.

Suh, Nam P. (19988). *Basic concepts for produce-ability*, CIRP Annals, Vol. 37/2/1988.

Valckenaers, P., Van Brussel, H., Kollingbaum, M., Bochmann, O. (2001). *Multi-agent coordination and control using stigmergy applied to manufacturing control*, Lecture notes in Artificial Intelligence, **2086**, 317-334, Springer-Verlag.