Management Studies, Sep.-Oct. 2019, Vol. 7, No. 5, 379-413
doi: 10.17265/2328-2185/2019.05.001

Holonic Management Tree Technique for Performance Improvement Over Self-Similar System Structures*

Andrea Bonci, Sauro Longhi, Massimiliano Pirani
Università Politecnica delle Marche, Ancona, Italy

This paper presents the detailed definition of technique for the automation of the management of complex business systems, whether they are constituted of services and productions. The technique has been referred here for the first time as Holonic Management Tree (HMT) and has been consistently contextualized into undergoing research on technological frameworks for the management cybernetics, in which the concept of viable system represents a fundamental stance. HMT is based on the use of recursive formulas over self-similar holonic structures for the attainment of continuous performance improvement in a complex and continuously evolving process. The problem is associated to a recursive tree of self-similar structures of which in this paper we discuss the interpretation with respect to knowledge modeling domain. The basic expressions for the computation of the 2nd degree trees are provided and explained in detail by means of an example in lean management context. Moreover, this work presents and discusses the expressions that handle the implementation of the n-th degree case as a recurrent abstraction of the basic and simple 2nd degree computation.

Keywords: performance improvement, overall throughput effectiveness, management cybernetics, complexity, holonic systems

Introduction

The management of the processes in industrial context is increasingly focused on the complexity of the interaction of the actors involved in the game. In the era of the Industry 4.0, the management involves holistically all the levels of decision making, from the strategies of the enterprise within social and economical domain down to the shop floor machineries. A recent reference architecture, the Reference Architecture Model Industrie (RAMI) 4.0 (VDI/VDE Society, 2015), has been proposed as a 3D reference grid for the development of the standards on technologies and processes, along with the ontological unification of the semantics between the several identified layers of the cyber-physical system (CPS) models (Nagorny, Scholze, Ruhl, & Colombo, 2018). In this framework, still the dominant trend is a functional reductionist approach, where complexity is

* This work has been partially funded by the European H2020 ENCORE project. The ENCORE project receives funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 820434.

Andrea Bonci, Ph.D., assistant professor, Department of Information Engineering (DII), Università Politecnica delle Marche, Ancona, Italy.
Sauro Longhi, Ph.D., full professor, Department of Information Engineering (DII), Università Politecnica delle Marche, Ancona, Italy.
Massimiliano Pirani, Ph.D., Department of Information Engineering (DII), Università Politecnica delle Marche, Ancona, Italy.
Correspondence concerning this article should be addressed to Andrea Bonci and Massimiliano Pirani, Department of Information Engineering, via Brecce Bianche 12, 60131, Ancona, Italy.

broken down into pieces to be better put under modeling and then control. Unfortunately, it turns out that the number of the constituting elements in the models tends to be very high and their interaction in time causes the facing of overly complex environments about which making good decisions at real time is very challenging. Cadsby (2014) analyses this kind of modern situations in which our thinking models fail to keep the pace with the complexity of the artificial and technological world we have created. He refers to this action environment as the complex *World #2*, in which cause and effect are apparently not as closely as expected, and in which our basic intuitions are not well adapted. Cadsby observes that for this world we have not developed reliable intuitions about how complex things work, hence we oversimplify with interpretive models that are too basic, and we are too confident in our preliminary conclusions. Though Cadsby posed the analysis correctly, still the synthesis of the solutions for complex worlds are far from being reachable. A new generation of technologies in automation could be the picklock for accessing unexplored possibilities for the tractability of the current overwhelming problems in the management and control of modern systems.

Artificial intelligence (AI), one of the weapons claimed by the Industry 4.0 through the CPS paradigm, is not mature enough for guaranteeing a perfect tool to human managers in their effective decisions at every level of the enterprise (Pearl & Mackenzie, 2018). Nonetheless, Pearl, the father of prominent, widespread and powerful tools for AI in the last decades, is confident in proposing his new causal inference framework towards the achievements that Strong AI promised since its inception. The relevant goal is to reach the long-time expected results in the effective harnessing of causality inference in complex systems. Current AI technology still cannot keep the pace of human abilities, regardless of the deluge of data and the many analysis techniques we have at our disposal today. Had the Pearl's dream realized soon, we would be able to obtain machines capable of human-like introspection and intuition, but the persisting risk is the eventual attainment of machines that still fail as the humans do in the Cadsby's perspective. This is indeed one of the inherent bounds to Strong AI when it has to cope with systems thinking like in the influential vision of Senge (2014). Systems thinking is a framework for seeing wholes, for seeing interrelations rather than things, for seeing patterns of change rather than static snapshots. Actually, the complex systems of the reality in which we are acting feature many nested levels of emerging behaviours, each of which involves different emergent domains and associated languages (Albertazzi, 2010; Poli & Obrst, 2010). The knowledge of the emergent levels seems to go even beyond the universe of discourse of human beings (Maciag, 2018) and therefore needs embodiment of some level of autonomous intelligence at many levels of granularity, as it apparently happens for fish schools or swarms in nature.

What seems desirable, in the near future, is a framework in which automation and AI are mixed in a convenient recipe to provide a step change in the power of information processing (Burgin, 2006), automation, and robotics technologies to reach the flexibility of humans in a perfect effective and safe collaboration. Nonetheless, humans must remain in complete control of the brute-force computations and of the actuating machines, leaving the possibility of a swift and purposeful exchange of roles. This is the actual trend in new approaches and standards in Industry 4.0 that introduced the CPS paradigm to allow for autonomic automation systems (made of self-* capable machines, where * stands for configuring, adjusting, optimizing, organizing, programming, and so on). Autonomic systems should pervade the reality of a production while collaborating at many levels with the humans. The Trusted Autonomy (TA) field of research focuses on understanding and designing the interaction space between two entities, each of which exhibits a level of autonomy. These entities can be humans, machines, or a mix of the two (Abbass, Petraki, Merrick, Harvey, & Barlow, 2016).

It is in the previous far-reaching scenario that the new means of management should be developed. The new, possibly automated, means should address widely adopted practices like Kaizen, just-in-time, total productive maintenance (TPM), and Value Stream Mapping (VSM). The common difficulty in all of the mentioned methods is their implementation. It concerns multidimensional aspects, along with the need of "soft practices" that involve people, their relations, and even psychology, to arrive at unavoidable implications in the complexity of environmental issues (Antomarioni, Bevilacqua, & Ciarapica, 2018).

The present work presents the kernel and the seeds of a methodology that goes in this direction. It provides primarily a ready-to-use tool that can allow management, at many levels of the company pyramid, to be applied as an automation. Here it is proposed a framework for the automation of management and decision making. Stemming on previous work form the authors, this paper corrects, refines, extends, and completes the discussion and the examples of the methodology that has been primarily conceived for distributed autonomous artificial entities, yet simple and handy enough to be used by a human with a spreadsheet or paper and pen as well. Moreover, we present here a first account and discussion on the potentiality of the methodology under the perspective of the automation of the management. Indeed, the main aim of the technique resides firstly in its simplicity of the structure of the computation. Secondly, the methodology allows a plug-in capability into a cognitive context of knowledge modeling and reasoning about complex systems of systems.

The method relies on the definition of four basic system structures as primitives that appear recursively at different levels of a tree-shaped hierarchy of a system of systems (SoS), which usually represents a convenient modeling of nested productive goals that has to undergo control processes. The control structure can be associated to a work breakdown structure (WBS) of the process through hierarchical and incremental decomposition into purposeful production phases, along with increasing granular details on the work performed by the productive unit tasks. The method allows leaving the hierarchy open at the top and open at the bottom with no granularity specified a-priori. To each element of the structure, and of the tree, a set of performance indicators is recursively associated to the structures that map continuously the state of task goal achievements. Based on the outcome of these performance indicators, the manager is guided towards a continuous and monotonic improvement of the whole process by means of single and minimal subsequent actions. The computation mechanism will suggest timely the amount of possible improvement of the processes that are detected as bottlenecks. Bottlenecks are here defined as of being system parts where there is an opportunity to provide improvement to the whole process.

Through a set of suggested possible actions and the possibility to simulate the improvements in nature and quantity, the manager is capable to decide the next step towards improvement with respect to a strategy or policy. The tool is an evolving tool as it is based on the quality of the modeling of the actual process under control. If a suggested action does not provide the foreseen result on the system, the departure from expected efficacy is considered a signal for the manager herself to improve her model or knowledge of the reality. The improvement of the model is improved piece-wise, gradually, only on the parts that contingently deserve an upgrade, not the whole system, in a best effort and minimal way fashion.

In Section 2 we recall the basics of the methodology, developed in detail for structures of the trees of the second degree. In Section 3 the methodology is further discussed with respect to the related work in the field and the ongoing research. In Section 4, a generalization of the methodology is provided for trees of n-th degree. Section 5 is dedicated to examples of the computation. Section 6 details an outlook on future work, and Section 7 is dedicated to the conclusion of the work.

Basics on the Performance Improvement Methodology

In this section, the performance improvement technique is introduced with an introductory example. For a more consistent and detailed introduction to the methodology and its historical developments, from the initial developments industrial automation field, reader may refer to Pirani (2016), Bonci (2017a), and Bonci (2018a). In the following, some recalls about the technique and a simple example are discussed within the context of management problems. Beyond the basics recall, in this work new formulations are provided along with corrections and simplification of the methodology with respect to earlier works. It is done by leveraging the hindsight of the new developments and generalization of the technique to the n-th degree (the definition of degree in the following), as firstly treated in Bonci (2018b).

The main problem at hand is about how to provide the management structure with a tool and a well-determinate computation structure that helps to automate the decisions and the actions towards a monotonic improvement of the indicators of performance. A process is recursively decomposed into a hierarchical breakdown structure that follows the holonic principles (Koestler, 1970). Each task in the hierarchy, following the holonic paradigm, is internally decomposed in a sub-tree of children tasks, whilst being at the same time a child (a part) of some other parent task. A task is associated with a goal and indicators that quantify the status of its achievement. This association, made through semantic attributions and interpretation of the reality, is the key of the methodology. The complexities of the reality are encapsulated into a (viable) abstract model of the task. Improving the task means often to act in a complex way. For example, the same improvement process can reproduce a complex cycle of improvement of the kind of "Current state map"—"Kaizen action"—"Future state map" (Antomarioni et al., 2018). The bigger the granularity of the task being improved, the more complex the actions. Nevertheless, tasks can be decomposed indefinitely into sub-tasks, which control actions will tend to simplicity and be eventually prone to automation.

Unfortunately, once a viable interpretation of a process into a hierarchical tree of tasks is obtained, we will be in front of a multivariate optimization problem that is typically computationally hard. Moreover, the contexts of the environment usually change and evolve too rapidly with respect to the possibility to obtain on time a solution; solutions coming from optimization are the best possible by definition, but they may never arrive on time. The proposed technique gives some relief to this problem by abstaining from optimization ambitions. It aims to provide an automation that guarantees a minimal sequence of well-controlled single-variable actions that lead to the improvement of the performance of the overall system.

The method consists in extending the semantic attribution of a triple of key performance indicators, typically adopted in manufacturing, that constitute a standard definition of the OEE (overall equipment effectiveness) (Muthiah, 2007):

$$OEE = Aeff \times Peff \times Qeff \quad (1)$$

where: *Aeff* = "availability efficiency", captures the deleterious effects due to breakdowns, setups, and adjustments; *Peff* = "performance efficiency", captures productivity loss due to reduced speed, idling, and minor stoppages; *Qeff* = "quality efficiency", captures loss due to defects, and rework.

If we abstract the concept of *equipment* used in (1) and Muthiah and Huang (2007) to "the coordinated set of resources of a production unit", and if the production is interpreted as the process of achievement of a task with associated goal, the *OEE* becomes a flexible indicator.

OEE is in general made up of a speed factor (*Peff*), an availability or capability factor (*Aeff*), and a quality factor (*Qeff*). This expression of the indicator is prone to a simple recursive formulation of performance indicators like the *OTE* (overall throughput effectiveness). *OTE* measures the performance of a whole system made up of a flow or a composition of sub-tasks. The *OTE* performance metric can be recursively computed from the *OEE* of the production units by using the expression in Figure 1.

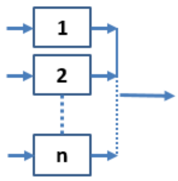
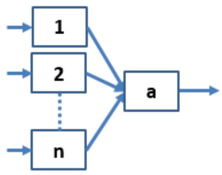
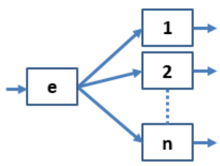
System type		OTE of system	Qeff of system
Series		$\min\left\{\min\limits_{i=1,\dots,n-1}\left\{O_i \cdot \prod\limits_{j=i+1}^{n} Q_i\right\}, O_n\right\}$	$\prod\limits_{i=1}^{n} Q_i$
Parallel		$\dfrac{\sum\limits_{i=1}^{n} O_i}{n}$	$\dfrac{\sum\limits_{i=1}^{n} Q_i}{n}$
Assembly		$\min\left\{\min\limits_{i=1,\dots,n}\left\{\dfrac{O_i \cdot Q_a}{k_{a,i}}\right\}, O_a\right\}$	$\dfrac{\sum\limits_{i=1}^{n} k_i Q_i}{\sum\limits_{i=1}^{n} k_i} Q_a$
Expansion		$\sum\limits_{i=1}^{n} \min\left\{O_e \cdot k_i \cdot Q_i, O_i\right\}$	$\dfrac{\sum\limits_{i=1}^{n} k_i Q_i}{\sum\limits_{i=1}^{n} k_i}$

Figure 1. Formulas for the computation of OTE and Qeff with respect to the four basic system structures.

The notation of Figure 1 assigns *Q* as *Qeff*, and *O* as either *OTE* or *OEE*, depending on the position of the unit in the tree. The *OTE* of a parent node depends recursively on the following variables of the children systems: O_i, the OTE of the *i*-th children; Q_i, the quality factor of the *i*-th children ($i = 1, \dots, n$, where *n* is the degree of the tree). O_a and Q_a, are respectively the *OTE* and quality of the head node in an *assembly*. O_e and Q_e, are respectively the *OTE* and quality of the head system in an *expansion*. k_i, should appear as weighting factors on the edges of the structures in the *assembly* and *expansion* systems. The *O* quantity is evaluated with (1) as an *OEE* only on the leaves of the system's tree. The *O* of other upper nodes, i.e., *OTE*, is recursively obtained at every tree level with the formulas of Figure 1. The formulas of Figure 1 are a simplified, though extended in meaning and in their recursive expression, version of the original formulation of Muthiah and Huang (2007), with the assumption that some factors can be reinterpreted (Bonci, 2017a). In addition, it has to be noted that the word "system" mentioned in the context of Figure 1 denotes only one holonic node of the overall system's tree that can extend itself open at the top and at the bottom, according to the vision of Abbot (2006). This holonic node can assume one of the four fundamental structures: *series*, *parallel*, *assembly*, and *expansion*. The limited choice among the set of four structures provides self-similarity to the overall system model. The use of recursive approaches and self-similarity is the key to the programming of systems of systems with high granularity of autonomous components (Calabrese, Amato, Di Lecce, & Piuri, 2010). With a tree-based representation of the holarchy (a hierarchy of the holons) of the system, a recursive computing for the improvement of the effectiveness of productive components and of the overall system can be performed.

Basing on the previous arguments, the methodology here explained will be deemed henceforth as Holonic Management Tree (HMT). In this paper, we will focus only on the core element of the methodology that covers only one level of the tree at time.

For a node system in the tree, there are usually more actions available towards improvement. They can be associated to a definition of bottleneck in which the bottleneck is an opportunity for the improvement. Due to the *min* operator in the formulas of Figure 1, only the parts that are sifted by the *min* operator provide sensible places where to focus attention for the improvements. If we act on other parts with different variables, the costs and the efforts are not justified. In general, it is not so easy to relate performance measurements directly to the detection and the identification of bottlenecks. Mostly because the definition of bottleneck, and then of the critical path in a workflow, might not be unique and straightforward (Wang, Chen, Zhang, & Huang, 2016). Nonetheless, the recursive definition and the simple formulation of the OTE here adopted reliefs also this problem, lest the granularity and so the accuracy of the models are fit to the management problem.

In Figure 2, a flow chart is provided to express the performance improvement process, of which an implementation detail on distributed automations can be found in Bonci (2018a).

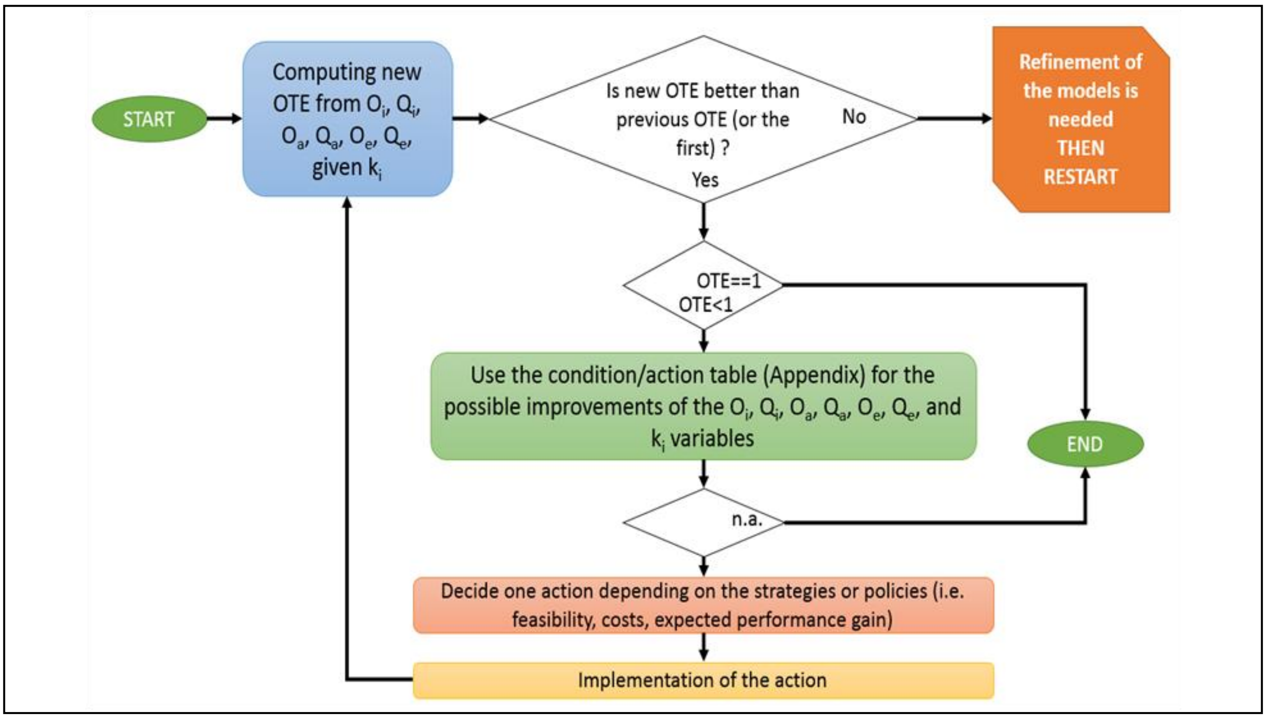


Figure 2. Conceptual flowchart of the performance improvement workflow for a node system.

The process and the flow of Figure 2 is performed top-down and bottom-up for every node of the systems' tree: Top-down processing achieves the recursive analysis of the system's bottlenecks; bottom-up is the effects of the physical actions that can be implemented only on the leaves of the tree, but which amount and effect on the whole tree can be previewed. The process starts with the computation of the OTEs and the sensing of the OEEs as provided by (1) and the formulas of Figure 1. Then, depending on the structure at hand, the condition/action tables of Appendix can be used. First, it is established in which status of the computation we are with the leftmost column of the tables. Then, a set of maximum (theoretical) improvements of specific variables is provided. The rightmost columns of the tables in Appendix provide a measure of the maximum expected gain of the *OTE* that is going to be obtained with a specific action. Note, that also the actions with 0 gain might be useful at a subsequent steps, as they let the system to escape from a possible stagnant status. Having this information, the next action can be planned with respect to some policy or contingency. For example, if actions on O_1 and Q_1 are suggested, maybe that Q_1 in practice (or at the very moment) is not a viable or not a practicable simple solution (indeed usually, the quality of productions is the most difficult variable to tweak quickly in real cases). After the choice of the next action, and having implemented it, the whole system is re-assessed again. If the *OTE* reaches 1, nothing is left to do. If the new *OTE* is worse than the previous, something in the model is wrong and must be refined before restarting the process. Else, if we obtained any improvement we can continue iterating.

Among all the quantities, the k_i needs a special treatment. They are parameters that are under direct control of the parent controller of the structure (assembly or expansion). They can be used to control the performance of the structure beyond the limitation in performance of the units by a weighing operation. They perform usually a balance, rather than a change of the system. However, the extent and the meaning of the actions on k_i depend on the specific interpretation.

In the following, an introductory example is provided to better understand the detail of the whole procedure. We will instantiate an example of management at a high level of decision, where the automation of the implementation of the actions is impossible at the present day and at the state of the art of artificial intelligence. It means that in this case, the model will only be able to suggest the right direction in which the manager should focus her decisions, and take into account her complex capabilities in the implementation of them.

This example will provide a link to management cybernetics and to lean management frameworks. Steinhaeusser, Elezi, Tommelein, and Lindemann (2015) show that management cybernetics can help sharpen the understanding of the implementing of lean thinking in an industrial context. Management cybernetics may also help identify problems where the implementation of lean thinking does not live up to the desired results (Steinhaeusser et al., 2015). Under this premises, the example system here displayed represents a toy problem just to highlight the principles of the methodology in detail. Nonetheless, it is contextualized in a work of reference in a study about the effects of the adoption of lean practices on operational effectiveness and overall company performance management (Bevilacqua, Ciarapica, & De Sanctis, 2017).

In Figure 3, a simple *assembly* structure model maps the goals of operational responsiveness of a productive company. As studied in Bevilacqua et al. (2017), operational responsiveness is contributed from the capability of achieving some product mix variety and product innovation at the same time. A combination of the two is "assembled" into a goal of overall operational responsiveness. The third cell in the assembly structure, deemed *head* cell, assembles the two heterogeneous effects into a global outcome. In this view, usually the head cell is given the interpretation of "the capability of the assembly to capture and assemble the incoming effects". Moreover, the parameters k_1 and k_2, can represent the weighted amount of the first component with respect to the second into the assembly. While the OEE of a cell measures the achievement of its production objective, a k parameter determines "how much" of that production objective should be taken. The Table 1 reports the interpretation of the cells used in the example. Note that this table is the link between the performance indicators and their physical meaning. The complexity of the actions and of the interpretations reflects with a simple model of the reality, which might be adequate only temporarily.

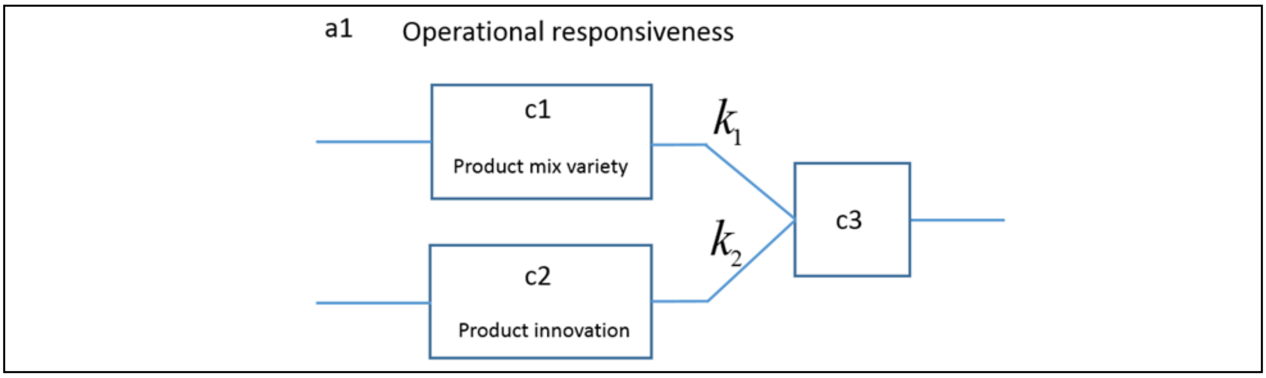


Figure 3. Assembly structure of the 2nd degree of the toy example.

Table 1

Interpretation of the Elements in the System of Example

Element id	Description	Interpretation of the goal	Interpretation of the action
a1	Operational responsiveness	OTE of a1 to rate the current capability of responding immediately to events, changing conditions, and customer actions with a minimum of extra steps or mistakes, in order to make the goal business done quickly and effectively.	n.a. (actions at this level has to depend only from action of children systems).
c1	Product mix variety	OTE of c1 (O_1) to rate the achievement of a goal in obtaining a desired (or requested) change in product mix variety. Depends on Aeff, Peff, and Qeff (Q_1). Aeff, is the availability to respond to a change of product mix. Peff is the speed in achieving the change of product mix. Qeff is the success in targeting the desired change of product mix without errors or misinterpretations.	Acting on OTE (O_1) means raising Aeff, Peff, or Qeff separately by increasing them. Aeff: improve available calendar days of resources that can perform the change. Peff: improve the tools or the knowledge that allow resources to address the request of speed up. Qeff (Q_1): act on incentives, tools, training, and turnover of resources to improve quality.
c2	Product innovation	OTE of c2 (O_2) to rate the achievement of a goal in obtaining a programmed innovation of the product. Depends on Aeff, Peff, and Qeff (Q_2). Aeff, is the availability to allocate efforts for the innovation of the product. Peff, is the speed in achieving the innovation. Qeff, is the success in targeting the innovation as planned with respect to some quality measure.	Acting on OTE (O_2) means raising Aeff, Peff, or Qeff separately by increasing them. Aeff: improve available calendar days of resources that can achieve the innovation. Peff: improve the tools or the knowledge that allow resources to address a fast innovation. Qeff (Q_2): act on incentives, tools, training, and turnover of resources to improve quality.
c3	Head cell of the assembly	OTE of c3 (Oa) to rate the achievement of assembling the outcomes of c1 and c2 into the objectives of a2, and to respond quickly to changes of k_1 and k_2. For example, the sales department, and the technicians that factually adopt the changes in product mix and the innovations towards the customers. Depends on Aeff, Peff, and Qeff (Q_a). Aeff, is the availability of the resources in the sales dept, or technicians to adopt the changes. Peff, is the speed in the adoption of changes. Qeff (Q_a), is the quality of transmitting effectively the complete value of the changes to the customers.	Acting on OTE (Oa) means raising Aeff, Peff, or Qeff separately by increasing them. Aeff: improve availability of resources that can reorganize the sales or the technician squads. Peff: improve the tools or the knowledge that allow resources to address a reorganization of the sales or technician squads. Qeff (Q_a): act on incentives, tools, training, and turnover of resources to improve quality of sales dept, or technician squads.

Table 1 to be continued

k_1	Weighing factor of c1	Relative amount of contribution of product mix variety.	Change relative amount of contribution of product mix variety.
k_2	Weighing factor of c2	Relative amount of contribution of product innovation.	Change relative amount of contribution of product mix variety.

Before proceeding with the example, we have to state that all the variables take a value comprised between 0 and 1. In particular, assumption is made that $k_1 + k_2 = 1$. The mapping that connects mathematically an indicator with a physical can have any operator closed or open form, but with codomain (0, 1]. Nevertheless, in nature most of the phenomena follow a logistic curve shape with saturations at the extremes of the codomain. Hence, a sigmoidal function choice is very common choice in most of the cases. A discussion and example of this argument has been already treated for a robotic cell use case in (Indri, 2018) in quite a detail. For the purpose of our treatise, the details of the mapping between the physical realm and the computational structure are left uncovered and beyond the scope of the paper.

We start with an arbitrary, but always possible, assignment of the initial indicators that puts the models in a central and balanced situation for the system:

$$[O_a, O_1, O_2, k_1, k_2, Q_a, Q_1, Q_2] = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5] \quad (2)$$

This is a convenient choice to let numerical problems be normalized, and to leave the maximum sensitivity available for the improvements.

By the use of the formulas for the OTE of an assembly system, the OTE of *a1* is initially equal to 0.5. At this point, by computing the ratios O_1 / k_1, O_2 / k_2, and O_a / Q_a, we find ourselves to be in the condition *a13* of Table A3 in Appendix. Under this condition, we have only the following choices for proceeding towards improvements:

$$[\Delta O_a, \Delta O_1, \Delta O_2, \Delta Q_a] = [0.5, 0.5, 0.5, 0.5] \quad (3)$$

The increments suggested in (3) are the maximum allowed ones. To continue, we suppose that we can afford (or figure out a way) to increase the effectiveness of the product mix variety to let $\Delta O_1 = +0.25$. This increment can be obtained in turn by acting separately or not on Aeff, and Peff, like expressed in the Table 1 for the c1 system. At this point the overall OTE is computed against the following new vector of variables:

$$[O_a, O_1, O_2, k_1, k_2, Q_a, Q_1, Q_2] = [0.5, 0.75, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5] \quad (4)$$

Unfortunately the new OTE is still 0.5. Yet we should have noted before that condition *a13* does not provide for an immediate gain (rightmost column of the Table A3 in Appendix). In cases like this, the modifications of the variables allow only to escape from a condition to arrive at some possibly other options of immediate OTE improvement. Indeed, with the former choice we eventually land on condition *a9*. The new opportunities we have are:

$$[\Delta O_2, \Delta k_1, \Delta O_a, \Delta Q_a] = [0.25, 0.1, 0.25, 0.1667] \quad (5)$$

At this point, a manager decides to act on the maximum available margin on k_1 to obtain the following new variable vector:

$$[O_a, O_1, O_2, k_1, k_2, Q_a, Q_1, Q_2] = [0.5, 0.75, 0.5, 0.6, 0.4, 0.5, 0.5, 0.5] \quad (6)$$

OTE of system is still 0.5, but she knew from the table. Nevertheless, she escaped from condition *a9* to land on *a11*. Note that an increment of k_1 forced a corresponding decrement of k_2.

At this new point, being tired of no direct improvements, the only variable that would admit some immediate effect in condition *a11* is $\Delta O_a = +0.125$, producing the following vector:

$$[O_a, O_1, O_2, k_1, k_2, Q_a, Q_1, Q_2] = [0.6, 0.75, 0.5, 0.6, 0.4, 0.5, 0.5, 0.5] \quad (7)$$

With the variables in (7) we can achieve *OTE* = 0.6 for the whole system, an effective improvement. The process can continue iterating in this way, ideally to arrive at *OTE* = 1.

The question that may correctly arise at this very moment is: Have the actual system improved of an amount of 0.1 (20% more effectiveness) after the implementation of the actions? This is a central theme. The new performance obtained depends both on how adequate the model is and on how well the enforcement of the actions is made. If we do not observe actual improvements (or worse a diminishing performance), this means that our model is wrong. This means also that the knowledge about the physical underlying environment is inaccurate. The good news is that we have now a hint on which part of the modeling we have to improve first. By following the example, if acting on O_a did not meet the expectations, it means that the model (interpretation) of c3 must be refined, and we can focus there before redoing the process. This mechanism pushes both the model and the system to improvements at least a minimum viable state, in line with the renowned “simple but not simpler” motto and the well-known framework on viability of systems due to Beer (1981).

It must be noted that the possible regrets from a manager for having missed some other better but undetected opportunities of performance improvement can be well balanced from the possibility of having a safe and prompt direction for an improvement in a very complex context like this. The manager has always an opportunity that avoids stagnating and being hapless in making new decisions that would have involved deep and hard multivariate optimizations. This is particularly valuable in situations in which both the modeling and the computational means are very limited due to time, space, technology, and costs. Actually, this methodology was born in a context where very tiny, but numerous, components are to constitute a relevant part of an evolving cyber-physical system (Pirani, 2016; Bonci, 2018a).

Having shown the main characteristics of the methodology and the technique, in the next section some discussion is opened to contextualize it with respect to ongoing linked research in the many fields of concern for management aspects. Next section will provide also references to a more complete integration of the methodology in the context of holonic research in system of systems.

The Use of the HMT Methodology in Management and Related Work

In the previous section, we have started to explore the capabilities and the essentials of the techniques in the HMT (Holonic Management Tree) method. This section is used to discuss the role of this methodology in particular for applications that are of interest to the management processes.

The HMT Understanding of Complexity

There are many context-dependent definitions of complexity. In this paper, we used this term above without appropriately contextualizing. According to Burgin (2016), the theory of computation can provide some definitions that can be useful in operational and efficiency terms. In the most common vision, a complex entity cannot be divided into parts that can provide sufficient information to deterministically or statistically predict the properties of the other parts. Burgin interprets this kind of complexity mostly in terms of efficiency, in relation to the processes that can be applied to a system (Burgin 2016, p. 126): "Complexity of a system R with respect to a process (or a group of processes) P is the quantitative or qualitative characteristic (measure) of resources necessary for (used by) the process P involving R". In the present current decision-making and management context, this interpretation would mostly be useful in the development of methods that detect the onset of some computational difficulties in the modeling and in the control of the continuously evolving environmental reality. Nevertheless, Burgin's definition does not explicitly address the dynamical development of the systems. The evolution of a system, possibly unpredictable or unexpected, is a major characteristic that discriminates between the complexity and the complicatedness status of a situation. Actually, the mere act of measuring of the resources involved in the solution of a computation cannot effectively distinguish a complicated process from a complex one. A complicated problem is a problem that requires big effort to be solved, but the process to the solution is stable and reproducible once reached. In this case, the complexity resided only on the route towards the solution, and it disappears at a certain point. Nonetheless, a complicated process cannot solve a complex problem as it is not stable and continuously changing (Badinelli, Barile, Ng, Polese, Saviano, & Di Nauta, 2012). In theoretical computational terms, it means that the solution to a complex problem requires a never-stopping computation, as in super-recursive computation frameworks that go beyond the Church-Turing thesis (Burgin, 2006). To exemplify, it is rather useless to optimize a problem in complex scenarios. Optimization, itself a complex process, has to come up with a result, which is complicated enough to provide an optimal solution as far as the scenario is immutable (lest some robust optimization methods). As some parameters in the system change, a complex process towards a new complicated solution has to be restarted. If the variables of the controlled system are huge, the optimization problem is rapidly intractable. In addition, a renowned result of Ross Ashby states that a controller process has to feature the same variety of the controlled process or system (Conant & Ross Ashby, 1970), which means that model reduction methods risk being not adequate in many practical cases. Normally, to support decision-making processes in fast-changing contexts characterized by emerging and unexpected interactions, the qualitative approaches are more considered (Badinelli et al., 2012).

It is under these premises that the HMT methodology proposes a simple computation structure that tries to keep the pace of the multivariate and dynamic complex system evolutions, to provide a continuous improvement, although not optimal. Note that the simplicity in HMT is mostly on its recursivity and the self-similarity of the structures. Questionably, for highly deep trees of high degree, the complicatedness appears again if no automation of the HMT programming is obtainable (Bonci, 2018c; Bonci 2018f). Nonetheless, the HMT principle tends to use the as-simple-as-possible tree and starts with it, according to a lazy approach that refines the variety degree only when this is strictly needed, by plugging HMT in the context of a viable system approach (Bonci, 2018c; Badinelli et al., 2012), moreover, to make the level of precision of a control adjustable with each acquisition of new knowledge about the controlled system. Badinelli et al. (2012) prescribe an adaptive, fuzzy-logic controller as a robust modeling tool for agents' epistemology and decision-making. In analogy to the fuzzy controller membership function definition, the complexity encompassed by the interpretation of the HTM structures, as exemplified in section 2 (Table 1), is obtained through abductive processes, leaving deductive and inductive methods as a feedback in the control system (Badinelli et al., 2012).

Seminal Endeavours of the HMT

In the seminal work of the methodology, Pirani (2016) firstly proposed the adoption of this technique in the context of low cost and low size embedded electronics devices. Nowadays, tiny intelligent devices tend to constitute a major part of reference architectures of distributed control in industrial context, where intelligence permeates at different levels by means of holonic representations under the relational model (Bonci, 2016; Bonci, 2018d; Bonci, 2018e). In all of its versions, the proposed approach relies heavily on the simplification obtained by means of the leveraging of the assumption of self-similarity across the numerous components of the production process. Initially, the methodology covered mostly the automation of the shop-floor actors. In a second step, it has been extended to holistically control the improvements of a bigger part of the value chain encompassing the supply management, by means of appropriate interpretation of the OTE tree (Bonci, 2017b). In (Stadnicka, 2017a), the methodology has been proposed as a basis and a tool for the management of workflows in service processes guided by the value stream mapping principles, in association to knowledge bases and artificial intelligence reasoning. In (Stadnicka, 2017b) the technique has been adapted to cover a typical management problem modelled with the use of the Ishikawa diagram. In addition, (Stadnicka, 2017b) showed a first example of seamless integration of human management and CPSs, in a context in which the massive introduction of digital technologies is not fully appropriate or feasible. In particular, it happens in contexts where the great part of the processes requires human flexibility and the unique adaptive capabilities and skills of the human operators. The HMT (still not named at the time) resulted a valid computational infrastructure also in those kinds of contexts.

Other involved sector, in which the human component is prevalent due to the dynamically changing of the process with respect to the variety of the environment and the resources, is the construction and building sector. In this sector, the most advanced infrastructure for the introduction of digitalization and ICT (information and communications technology) is currently connected to the new evolutions of the BIM (building information modeling). The new BIM should become the digital informational hub to monitor and control the building performance in real-time. In (Pirani, 2018), a HMT computation structure has been developed for indoor comfort management in an office room. The structure developed was able to drive both the operation management phase and the medium- and long-term improvement of performances of the buildings in association to its plants. Carbonari (2018) copes with the principles of efficiency and cost-effectiveness in the wider context of facility management. In addition, HMT has been used in a CPS scenario to successfully advice the operation management of buildings, as well as long-term refurbishment processes.

Holonic Methods in Viable Systems

Mella (2014) proposes an interesting simplification of the control for of complex causal systems stemming from the systems thinking approach of Senge (2014). Even though these valuable kinds of simplification and useful tools exist, the current methodologies do not address satisfactorily the integration between humans and the new possibilities of cyber physical systems (CPSs) as a homogeneous whole. The cybernetic theory of organizations encapsulated in the VSM (Beer, 1981; Vahidi, Aliahmadi, & Teimoury, 2018) prescribes that viable systems are recursive and contain viable systems that can be modelled using an invariant cybernetic description within the containment hierarchy. Badinelli et al. (2012) highlight the contribution of the viable systems approach as an interpretative and governance methodology based on systems thinking. Steinhaeusser et al. (2015) describe lean thinking rules from the perspective of management cybernetics and explore the use of management cybernetics through the Beer's VSM, as a theoretical basis to establish how cybernetics hierarchies should be used to address the complexity of management. In addition, Nechansky (2010) notes that a hierarchy becomes a cybernetic necessity, whenever conflicting interests have to be settled or cooperating systems reach their maximum channel capacity for direct communication. Still in Nechansky (2010), the Beer's VSM is credited to acquire importance in particular where changes occur from one-man companies to small companies with more production lines and more employees working in a changing environment. This means that the necessity for a (continuously evolving) hierarchy arises in particular not for static systems, but in dynamically evolving systems.

In Bonci (2018c), an adaptation of the VSM leads to the cyber-physical viable system (CPVS). CPVS is a model that is designed to fill the gap between the new paradigm of CPS and the VSM framework, by establishing a continuum between humans and cybernetics. HMT complies with a lean approach that better links humans and ICT. This will come in form of ubiquitous computing and calm technology (Weiser & Brown, 1997), in which technology is designed to enable people to do what they want, need, and be always in the loop (in cybernetics' sense): disappearing computer; things that think.

The CPVS has the ambition to refresh the impact of Beer's work in industry, as it mostly was deemed impractical or at least difficult to apply to real situations (Steinhaeusser et al., 2015). As a basic tool for the CPVS model, the HMT automation here treated should be propaedeutic to the effective application of Beer's approach to lean thinking.

A key to the effective implementation of the CPVS model is the holonic hierarchical view that constitutes the foundation of the HMT (Bonci, 2018c). A satisfying account on the many forms of holonic structures and interpretations in the management area can be found in Mella (2009). The form of holonic system more interesting for the HMT is the dynamic hierarchical holarchy. The hierarchy is inherently implied by the computation in the form of a three, but it does not represent the only and exhaustive use of the holonic structures.

More often, holonic systems are shaped as a network of roles. It is the dominant interpretation in industry since the successful PROSA architecture (Van Brussel, Wyns, Valckenaers, Bongaerts, & Peeters, 1998). An account on the long experience on the developments of the PROSA architecture and its evolutions has been provided by Valckenaers and Van Brussel (2016). Nonetheless, the most important contribution of Valckenaers and Van Brussel (2016) is the definition and the focus on the D4U (design for unexpected). D4U design supports lazy development. It consists in specifying the minimum viable attributes of an entity and leaves plenty of room for refinements and evolutions. Deeper modeling is performed only when strictly needed, relying on a continuous efficient monitoring across the system-of-systems. The resilience capability (i.e., getting back to normal operations in case of unforeseen disturbances) is fundamental for viability. It will always require a model for what is beyond the "world-of-interest". A coarse model providing information on to whom asking intervention or support (humans or artificial agents) in case something unexpected happens.

This D4U concept makes also a natural link to a supplementary feature that the VSM model of Beer should require for its viability, namely the algedonic loop. HMT can be a key for its implementation. The algedonic loop is an important feature for a system to "survive" to unexpected dangerous events occurring at instantaneous time-scale. The algedonic cut-through communication is a safety device that bypasses most or all the layers of the hierarchical system through an escalation mechanism that avoids detailed analysis. This mechanism comes along with the VSM, but not part of it (Carter, 2016; Steinhaeusser et al., 2015). HMT, seen as a holonic overlay over the system of systems (Bonci, 2018f), can detect efficiently the deviations from system's capability. When some of the actors did something well or something badly, an algedonic alert should be sent to the parent level. If corrective action, adoption of a good technique, or correction of an error, is not handled in a timely manner from that level, the alert is quickly escalated. In the HMT vision, algedonic alerts are alarms and rewards (bad or good performance indications) that escalate through the levels of the HMT recursion, thanks to the sifting nature of the performance bottleneck detection (Bonci, 2018a). This mechanism is linked typically to a timeout trigger, which can be simply implemented, for example, by sensing a fast decreasing of the *Peff* (defined in the previous section) that is linked to a speed of achievement of a goal or task.

HMT Under the Management Cybernetics Approach

Through the use of the HMT methodology and technique here proposed as a fundamental management tool component, the automation of the management and governance at all the levels of the system of systems can be potentially obtained at low cost. The HMT technique can yet be ascribed as another useful tool in the management cybernetics field (Vahidi et al., 2018). Management cybernetics establishes a foundational framework for the harnessing of complex systems of systems of the enterprise (Vahidi et al., 2018; Steinhaeusser et al., 2015). Carter (2016) identifies 68 architecture frameworks, by following the ISO/IEC/IEEE 42010:2011 standard, which are linked to the management and the governance of complex systems based on the management cybernetics concept and the viable system model (VSM) of Beer (1981). The most significant gaps found by Carter in existing enterprise architectures and systems theory is attributed to high variety in existing frameworks and architectures and to the lack of defined minimal critical specifications for a complex system governance architecture framework. In addition, the number of viewpoints and views extends the number of architecture views to 166 under research-driven criteria that necessarily cannot entirely be prescriptive due to each complex system's unique context (Carter, 2016). Definitely, in the complex systems' field the means of control are still too complex and not well specified. Actually, many management techniques and paradigms can show their limits. Many reports state that lean implementations result in large benefits in general. Nevertheless, still a lot of skepticism remains regarding attainable and measurable results and the possibility to apply Lean approach outside high volume manufacturing and stable contexts (Bevilacqua et al., 2017; Steinhaeusser et al., 2015).

The HMT can be a useful tool towards the implementation, validation, and simulation of several practices in the field. Indeed, the switching between several policies and management strategies can be supervised directly during the use of the instrument itself. The accurate interpretation and definition of the key performance indicators that are at the grounds of HMT assume a big role. HMT renders this role independent from the structure and the policies that remain abstract and invariant under pure computational terms. The validity of the computation structure can be validated independently from the specific domain at hand. On the other hand, the HMT tools permits the detection of lack of performance or model accuracy in each of these domains if the representation is made appropriately.

The HMT structure can be useful also to perform some causal probing of the structural equation modeling in a fashion to the casual inference methodology of Pearl, Glymour, and Jewell (2016). For example, to come back to the source of the HMT example of previous section, Bevilacqua et al. (2017) studied that the lean practices implementations can be negatively influenced by product mix variety and innovation, while positively influenced by time effectiveness variables. No direct relationship was found between lean best practices and firm's performances. The inspection of the causes and verification of a mediator variables may involve the verification of hypothesis like "Lack of resources and mainly poor communication and managers' commitment and support seem to be the main obstacles of lean implementation and success" (Bevilacqua et al., 2017). In this cases, the HMT can provide a simple means for testing the hypothesis by detecting at run-time or in simulation the lack of performance of certain sections in the system, and so the actual causal relationships between the relevant behaviours in the system.

In the next section, we will extend the structure and the formulas of the 2nd degree explained in the previous section to the general case of n-th degree.

Extension of the HMT to N-th Degree

The essential machinery of the performance improvement of a structure of 2nd degree has been explained in section 2, by means of the assembly toy example. In this section, an extension to the n-th degree of the technique is presented.

Propaedeutic Definitions and Notations

In order to define formally the degree of a fundamental structure, a reference is made to the table of the formulas for the *OTE* in Figure 1. We formally define the degree of a structure as the index n in the formulas. In other less formal terms, the degree of a fundamental structure is the number of the cells (fundamental units) involved by the structure, not counting the head cells in the *assembly* and *expansion* cases. Moreover, in this section a tree notation will be established to elaborate the recursive nesting of the structures of the HTM. For a system of systems mapped recursively into the four fundamental structures, the degree will be defined as the maximum degree of the fundamental structures in the recursion.

We need to introduce the following notations to express conveniently all the features and the manipulations for the extension at the n-th degree:

$$\overset{(l,g)}{S_X}(n) \quad ; \quad \overset{(l,g)}{O_X}(n) \triangleq O(\overset{(l,g)}{S_X}(n)) \quad ; \quad \overset{(l,g)}{Q_X}(n) \triangleq Q(\overset{(l,g)}{S_X}(n)) \quad ; \quad \overset{(l,g)}{k_{i,X}} \tag{8}$$

S denotes a system's structure, having other attributes and parameters as follows. The variable n is of the system S its degree. The parameter l provides an index for the depth level of the system in the tree associated to the systems-of-systems of fundamental structures. The parameter g is used to denote the "grouping" index. Its meaning will be clearer with delving into the mechanism of the recursive extension of the degrees. It will refer to an aggregation operation of some parts of a system into a virtually equivalent sub-system. The X subscript stands for the structure type and will be assigned to *S, P, A, E* to respectively denote *series*, *parallel*, *assembly*, and *expansion* structure types, and if required from the context, an additional system's identifier. Any structure is composed of cells and a structural relation among them. A cell can be a *composite* or *ground*. A *composite cell* can be decomposed recursively into another structure. A *ground cell* is not decomposable, or simply not decomposed at specific phase to stop the recursion. A *ground cell* is a leaf in the HMT tree. Note that, in the HMT a leaf can be transformed into a sub-tree when the details of the model about the environment start to become not effective. In that case, the ground of the tree moves below. If the structure is trivially a *ground cell*, the X subscript is used to express only a generic identifier of the system, with no particular reference to any structure.

In the expressions (8), the operators *O* and *Q* are the OTE and the recursive quality operators respectively, applied to the system *S* by means of the recursive formulas of Figure 1. The *k* is associated to the edges of the connectors from head to other cells in the *assembly* and *expansion*. For *k* the *X* identifier is optionally used along with the index *i* that identifies the link to *i*-th cell in the structure. Figure 4 shows an example of tree of the 3rd degree (with g = 0 as no grouping is made at this point). In Figure 5, we provide a different perspective of the 3rd degree system of systems. Note that the degree of the tree would have been the 2nd if we had removed the last series expansion at level 3 of the system $S_S^{(2,0)}(3)$. In Figure 5 it is more evident the link between the tree of Figure 4 and the fundamental structures. Indeed the two kinds of relations are of completely different nature. The *structural relation* is between sibling systems on the same tree level. This is the relation that produces the *OTE* through the formulas of Figure 1. The *tree relation* is the tree-based recursion that links *OTE* with *OEE* or the leaves with the root of the tree.

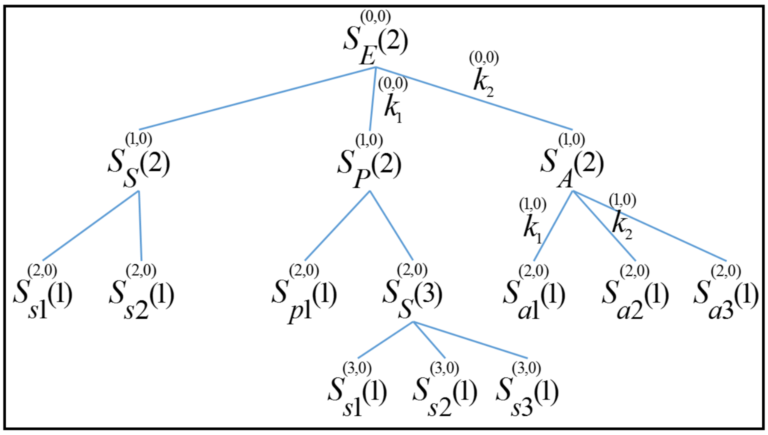


Figure 4. Tree of the system-of-systems. Example of the 3rd degree.

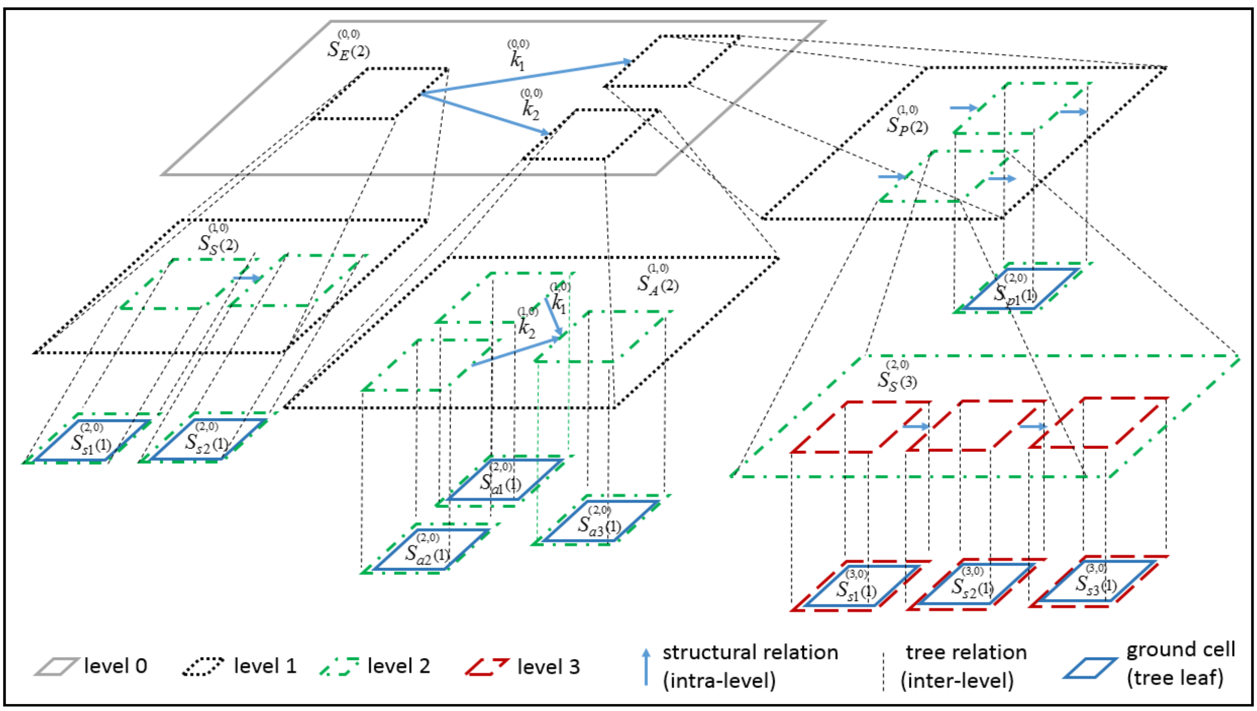


Figure 5. A different perspective of the tree of the system-of-systems of Figure 4.

Note that, in general the *tree relation* is what gives the self-similarity propriety to the system of systems. The *tree relation* is the most invariant and core part in the HMT method. The *structural relation* might be substituted with others, more general or more adapt for different domains. Nonetheless, the *structural relation* must be simple and admit a recursive definition in order to be embedded into a *tree relation* as previously stated. The choice of a particular formulation of the KPIs and goals determines only an instance of HMT. The choice of the OTE/OEE is to be considered as a first and reference instance of HMT, coming from the manufacturing domain.

In order to handle more conveniently the expressions that will lead to the extension of the HMT to the n-th degree, it is necessary to have a tool notation that lets us handle large trees in shorthand. To represent large and deep trees of structures we will rely on the handful Newick format (Olsen, 1990). A few extensions to the original Newick notation will be used to better denote the intra-level relation in the case of *assembly* or *expansion* structures. In this format, the tree is represented by a sequence of printable characters instead of graphs. The best way to explain the notation here used is by the simple following examples. In a first example, we represent the tree of an *expansion* structure in which a machine *A* is feeding production material to three production lines *B*, *C*, and *D* with a proportion of 10, 30, and 60 percent respectively. *A* is the head cell of the expansion. *B*, *C*, and *D* are the other cells of this 3rd degree structure that can be associated to a tree like this in Newick notation: *A(B:0.1,C:0.3,D:0.6)*. If the structure was an assembly, having *A* as head cell, the notation would have been the following: *(B:0.1,C:0.3,D:0.6)A.* If the system was a series, or a parallel, the notation results further simplified as no information is associated to the edges, to head cells, and only the parentheses of a tree node are needed, namely: *(B,C,D).* The parentheses in this notation enclose the sub-tree of a tree node. With considering the possible equivalences in the representations at the levels from 1 down to 3 of the tree, the notation for the system in Figure 4 and Figure 5 results in the following:

$$\begin{aligned}
\overset{(0,0)}{S_E}(2) &\triangleq \overset{(1,0)}{S_S}(2)\left(\overset{(1,0)}{S_P}(2):\overset{(0,0)}{k_1},\overset{(1,0)}{S_A}(2):\overset{(0,0)}{k_2}\right)\\
&\equiv \left(\overset{(2,0)}{S_{s1}}(1),\overset{(2,0)}{S_{s2}}(1)\right)\left(\left(\overset{(2,0)}{S_{p1}}(1),\overset{(2,0)}{S_S}(3)\right):\overset{(0,0)}{k_1},\left(\overset{(2,0)}{S_{a1}}(1):\overset{(1,0)}{k_1},\overset{(2,0)}{S_{a2}}(1):\overset{(1,0)}{k_2}\right)\overset{(2,0)}{S_{a3}}(1):\overset{(0,0)}{k_2}\right)\\
&\equiv \left(\overset{(2,0)}{S_{s1}}(1),\overset{(2,0)}{S_{s2}}(1)\right)\left(\left(\overset{(2,0)}{S_{p1}}(1),\left(\overset{(3,0)}{S_{s1}}(1),\overset{(3,0)}{S_{s2}}(1),\overset{(3,0)}{S_{s3}}(1)\right)\right):\overset{(0,0)}{k_1},\left(\overset{(2,0)}{S_{a1}}(1):\overset{(1,0)}{k_1},\overset{(2,0)}{S_{a2}}(1):\overset{(1,0)}{k_2}\right)\overset{(2,0)}{S_{a3}}(1):\overset{(0,0)}{k_2}\right)
\end{aligned} \tag{9}$$

At this point, we have all the necessary assets to face the problem of coping with HMT trees of n-th degree.

Approach

It has been shown previously that the HMT of the 2nd degree features structures having only two children elements (not counting the head cells). In this case, the analytical derivation of the multivariate conditions that lead to improvement actions is a manageable problem. Eventually, as previously shown, with the use of the tables in Appendix (through Table A1, A2, A3, and A4), a 2nd order problem can be handled also manually in a few finite number of passages, and the OTE formulas are still pretty manageable. For higher degrees, the number of the involved variables raises fast, bringing into play a typical effect of "curse of dimensionality". The OTE formulas of Figure 1 are not a problem themselves, but the decision on the improvement process becomes hard. Even starting from the 3rd degree things get far more complicated immediately and tables similar to the ones in appendix for the 3rd degree are difficult to be obtained in a handy and practical format, though possible in principle.

Our approach will be oriented to reduce an n-th degree problem recursively into a 2nd order problem, in order to achieve a simple automation means based on the easily computable tools of the 2nd degree so far obtained. Fortunately, this approach is possible as equivalence classes can be obtained for the recursive expression of parts of a structure into a sub-structure.

Due to the many possible (mostly arbitrary) interpretations that can be conveyed into a structure, many equivalent virtual trees and associated structures can model a system. Some structure will be more natural or easily applicable than the others, though all of them remain abstract general entities. If we achieve to transform an n-th degree structure into an equivalent tree of nested structures of the 2nd degree, the improvement decision problem is solvable. To this purpose, we will propose simple equivalences that transform an n-th degree structure into an equivalent recursive structure made of 2nd structures. This problem has not unique solution. A structure can be reshaped into many others with appropriate affinity relationships. Nevertheless, to keep some kind of homogeneity and naturality, useful in the conceptual integrity of implementation, we will propose equivalences of only self-similar structures.

Expressions for Performance Improvement of the N-th Degree

With the notations introduced in the previous section, in this section we will present a solution for the *series*, *parallel*, *assembly*, and *expansion* structures of n-th degree. Systems of the 4th degree will be used to let the examples and the expressions render a minimum but sufficient grade of generality.

Series Structure

The transformation of a 4th degree series structure into a recursion of 2nd order degree series is obtained in the following. By making reference to the notation introduced in (8), it is hereafter supposed that the series structure is situated a t the *l*-th level of a HMT, and so the first index (the tree depth level index) is left as a floating parameter in the examples and in the expressions. The role and meaning of the second index, the *group* index will be now clearly expressible with the pictorial example of Figure 6. A value 0 of the group index means a ground situation in which the system is left untreated. Then, the leftmost three systems are collected as children of a 3rd degree series system. The new parent system collecting the children is a new system featuring 1 as *group* index, denoting a first grouping action. In this way, the whole series is expressed with a 2nd degree series of a 3rd degree series plus the 1st degree S_4. With a second grouping action, the 3rd degree system is decomposed in a series of a 2nd degree plus a 1st degree system. Eventually we have obtained that the original 4th degree series is transformed into an equivalent 3-level tree of the 2nd degree made of series structures.

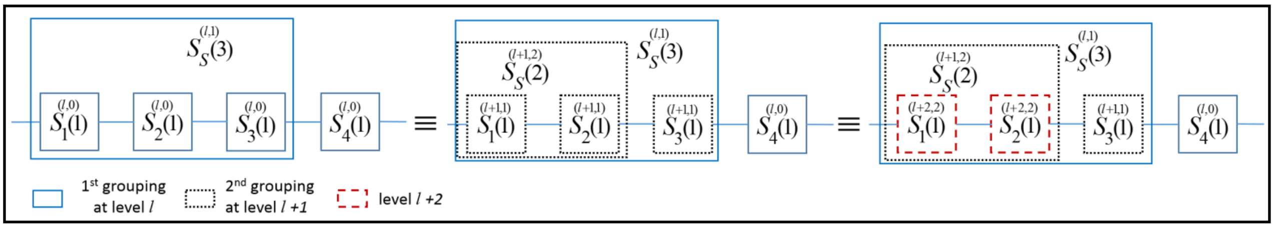


Figure 6. 4th degree equivalences for the series structure.

The grouping and nesting process depicted in Figure 6 can be expressed in the modified Newick notation as follows:

$$\overset{(l,0)}{S_S}(4) \triangleq \left(\overset{(l,0)}{S_1}(1), \overset{(l,0)}{S_2}(1), \overset{(l,0)}{S_3}(1), \overset{(l,0)}{S_4}(1) \right) \equiv \left(\overset{(l,1)}{S_S}(3), \overset{(l,0)}{S_4}(1) \right) \equiv \left(\left(\overset{(l+1,1)}{S_1}(1), \overset{(l+1,1)}{S_2}(1), \overset{(l+1,1)}{S_3}(1) \right), \overset{(l,0)}{S_4}(1) \right)$$
$$\equiv \left(\left(\overset{(l+1,2)}{S_S}(2), \overset{(l+1,1)}{S_3}(1) \right), \overset{(l,0)}{S_4}(1) \right) \equiv \left(\left(\left(\overset{(l+2,2)}{S_1}(1), \overset{(l+2,2)}{S_2}(1) \right), \overset{(l+1,1)}{S_3}(1) \right), \overset{(l,0)}{S_4}(1) \right) \tag{10}$$

The n-th degree expressions of the *series* structures are straightforward if we consider the structure formulas of the *O* and *Q* operator in Figure 1. The conditions that link the transformation of the systems across the grouping and deepening of the tree level in the series case are immediately obtained by mere substitution of the notation of the elements. Figure 1 provides the following expressions:

$$\overset{(l,0)}{Q_S}(n) \triangleq \prod_{i=1}^{n} \overset{(l,0)}{Q_i}(1) \tag{11}$$

$$\overset{(l,0)}{O_S}(n) \triangleq \min \left\{ \min_{i=1,\dots,n-1} \left\{ \overset{(l,0)}{O_i}(1) \prod_{j=i+1}^{n} \overset{(l,0)}{Q_j}(1) \right\}, \overset{(l,0)}{O_n}(1) \right\} \tag{12}$$

By the use of (11) and (12) and the transformations developed in (10), it is easy to obtain the following general relations between the n-th degree *O* and *Q* operators and the corresponding operators for the equivalent 2nd order systems:

$$\overset{(l,0)}{Q_S}(n) = \overset{(l,1)}{Q_S}(n-1)\, \overset{(l,0)}{Q_n}(1) \tag{13}$$

$$\overset{(l+g,g+1)}{Q_S}(m) = \prod_{i=1}^{m} \overset{(l+g+1,g+1)}{Q_i}(1) \tag{14}$$

$$\overset{(l,0)}{O_S}(n) \triangleq \min \left\{ \overset{(l,0)}{O_S}(n-1)\, \overset{(l,0)}{Q_n}(1), \overset{(l,0)}{O_n}(1) \right\} \tag{15}$$

$$\overset{(l+g,g+1)}{O_S}(m) \triangleq \min \left\{ \min_{i=1,\dots,m-1} \left\{ \overset{(l+g+1,g+1)}{O_i}(1) \prod_{j=i+1}^{m} \overset{(l+g+1,g+1)}{Q_j}(1) \right\}, \overset{(l+g+1,g+1)}{O_m}(1) \right\} \tag{16}$$

with $m = 1, ..., n\text{-}1$ and $g = 0, \dots, n\text{-}3$.

By the comparison between (11)-(12) and (13)-(16), the following final substitution expressions are obtained:

$$\overset{(l+g,g)}{Q_m}(1) = \overset{(l,0)}{Q_m}(1) \tag{17}$$

$$\overset{(l+g,g)}{O_m}(1) = \overset{(l,0)}{O_m}(1) \tag{18}$$

with $m = 1, ..., n$ and $g = n\text{-}max(m,2)$.

With (17)-(18) we have the mappings between the original n-th degree system and all the equivalent 2nd degree subsystems in the recursive tree as of (10). With these kinds of relationships, the performance of an n-th degree system is completely determined through the recursive use of 2nd degree expressions. A similar procedure will be used to obtain the relationships for the remaining structures in the following. It is also to be noted that in the series case, the result is trivial. For the other structures, the thing get more complicated, and the actual usefulness of this transformation process will be more evident.

Parallel Structure

An example of the recursive transformation for parallel structure is provided in Figure 7. For the parallel case, still the transformation is rather obvious, and the importance of the renaming and reinterpretations of the systems at level deeper than *l* might still be not so evident. In the following the importance of it will be more evident.

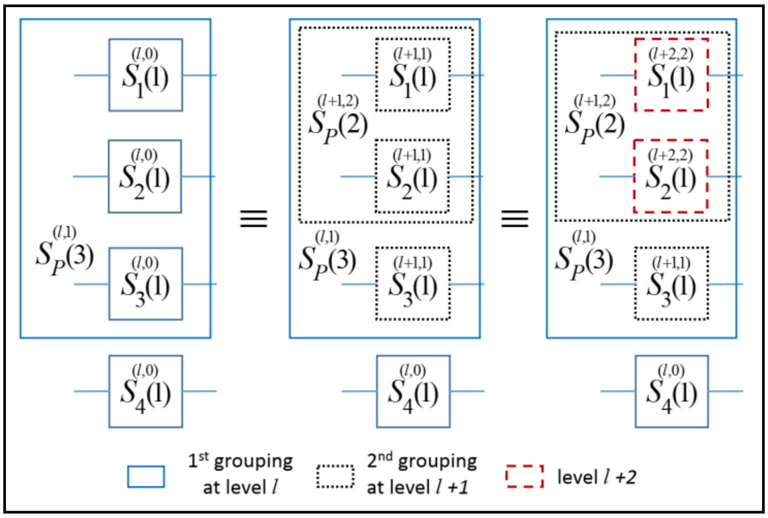


Figure 7. 4th degree equivalences for the parallel structure.

The Newick expressions, very similar to (10) that hold for the trees in Figure 7 are the following:

$$\begin{aligned}\overset{(l,0)}{S_P}(4) &\triangleq \left(\overset{(l,0)}{S_1}(1), \overset{(l,0)}{S_2}(1), \overset{(l,0)}{S_3}(1), \overset{(l,0)}{S_4}(1)\right) \equiv \left(\overset{(l,1)}{S_P}(3), \overset{(l,0)}{S_4}(1)\right) \equiv \left(\left(\overset{(l+1,1)}{S_1}(1), \overset{(l+1,1)}{S_2}(1), \overset{(l+1,1)}{S_3}(1)\right), \overset{(l,0)}{S_4}(1)\right) \\ &\equiv \left(\left(\overset{(l+1,2)}{S_P}(2), \overset{(l+1,1)}{S_3}(1)\right), \overset{(l,0)}{S_4}(1)\right) \equiv \left(\left(\left(\overset{(l+2,2)}{S_1}(1), \overset{(l+2,2)}{S_2}(1)\right), \overset{(l+1,1)}{S_3}(1)\right), \overset{(l,0)}{S_4}(1)\right)\end{aligned} \tag{18}$$

The n-th degree basic expressions of the *O* and *Q* operators for the *parallel* structure, drawn from Figure 1, are:

$$\overset{(l,0)}{Q_P}(n) \triangleq \frac{\left(\sum_{i=1}^{n} \overset{(l,0)}{Q_i}(1)\right)}{n} \tag{19}$$

$$\overset{(l,0)}{O_P}(n) \triangleq \frac{\left(\sum_{i=1}^{n} \overset{(l,0)}{O_i}(1)\right)}{n} \tag{20}$$

From (19) and (20) we can see that in this case O and Q operators have the same expression. For shorthand, we define operator P to be an alias for both O and Q in the following formulas derived from imposing the recursive structure as in (18):

$$\overset{(l,0)}{P_P}(n) = \frac{\overset{(l,1)}{P_P}(n-1) + \overset{(l,0)}{P_n}(1)}{n} \tag{21}$$

$$\overset{(l+g,g+1)}{P_P}(m) = \frac{\overset{(l+g+1,g+2)}{P_P}(m-1) + \overset{(l+g+1,g+1)}{P_m}(1)}{m} \quad ; m = 3,\ldots,n-1;\; g = 0,\ldots,n-m-1 \tag{22}$$

$$\overset{(l+g,g+1)}{P_P}(2) = \frac{\overset{(l+g+1,g+2)}{P_1}(1) + \overset{(l+g+1,g+1)}{P_2}(1)}{2} \quad ; \; g = n-3 \tag{23}$$

By comparing (21)-(23) with (19)-(20) it is natural to assume the following simple relationships between the original n-th degree parallel system parameters and the 2nd degree:

$$\overset{(l+g,g)}{P_m}(1) = \begin{cases} \dfrac{(n-1)!}{(m-1)!}\overset{(l,0)}{P_m}(1) & g = 0,1,\ldots,n-3 \\ (n-1)!\overset{(l,0)}{P_m}(1) & g = n-2 \end{cases} \tag{24}$$

Assembly Structure

The equivalent trees for the *assembly* structure are shown in Figure 8. In this case the transformation procedure is more elaborated than in the series and parallel case. The major difference resides on the adaptation of the k parameters. For every grouping action a new k is created at the parent level. In addition, a new head cell has to be created and added as well to replicate the assembly structure at the lower levels.

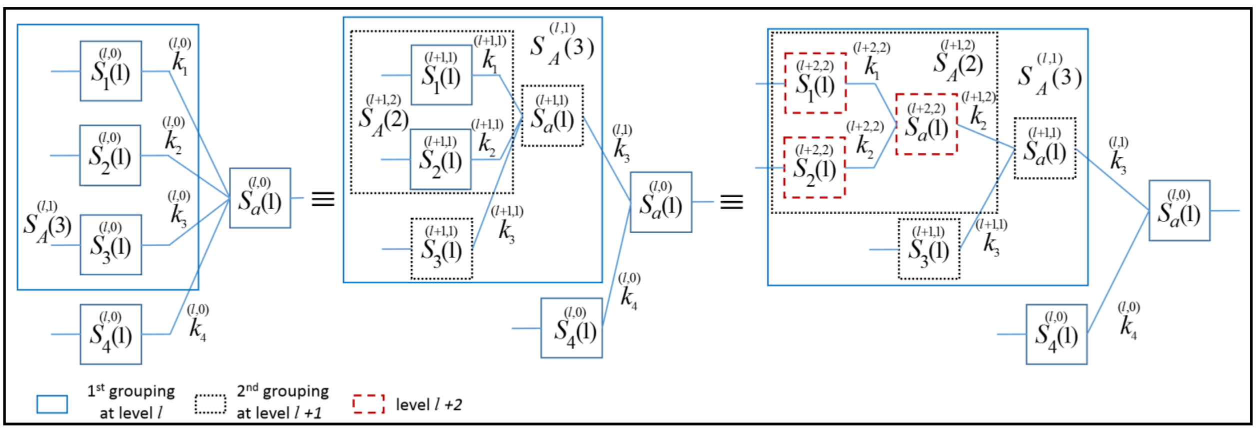


Figure 8. 4th degree equivalences for the assembly structure.

In Newick notation, the trees in Figure 8 can be expressed as follows:

$$\begin{aligned}
\overset{(l,0)}{S_A}(4) &\triangleq \left(\overset{(l,0)}{S_1}(1): \overset{(l,0)}{k_1}, \overset{(l,0)}{S_2}(1): \overset{(l,0)}{k_2}, \overset{(l,0)}{S_3}(1): \overset{(l,0)}{k_3}, \overset{(l,0)}{S_4}(1): \overset{(l,0)}{k_4} \right) \overset{(l,0)}{S_a}(1) \equiv \left(\overset{(l,1)}{S_A}(3): \overset{(l,1)}{k_3}, \overset{(l,0)}{S_4}(1): \overset{(l,0)}{k_4} \right) \overset{(l,0)}{S_a}(1) \\
&\equiv \left(\left(\overset{(l+1,1)}{S_1}(1): \overset{(l+1,1)}{k_1}, \overset{(l+1,1)}{S_2}(1): \overset{(l+1,1)}{k_2}, \overset{(l+1,1)}{S_3}(1): \overset{(l+1,1)}{k_3} \right) \overset{(l+1,1)}{S_a}(1): \overset{(l,1)}{k_3}, \overset{(l,0)}{S_4}(1): \overset{(l,0)}{k_4} \right) \overset{(l,0)}{S_a}(1) \\
&\equiv \left(\left(\overset{(l+1,2)}{S_A}(2): \overset{(l+1,2)}{k_2}, \overset{(l+1,1)}{S_3}(1): \overset{(l+1,1)}{k_3} \right) \overset{(l+1,1)}{S_a}(1): \overset{(l,1)}{k_3}, \overset{(l,0)}{S_4}(1): \overset{(l,0)}{k_4} \right) \overset{(l,0)}{S_a}(1) \\
&\equiv \left(\left(\left(\overset{(l+2,2)}{S_1}(1): \overset{(l+2,2)}{k_1}, \overset{(l+2,2)}{S_2}(1): \overset{(l+2,2)}{k_2} \right) \overset{(l+2,2)}{S_a}(1): \overset{(l+1,2)}{k_2}, \overset{(l+1,1)}{S_3}(1): \overset{(l+1,1)}{k_3} \right) \overset{(l,1)}{S_a}(1): \overset{(l,1)}{k_3}, \overset{(l,0)}{S_4}(1): \overset{(l,0)}{k_4} \right) \overset{(l,0)}{S_a}(1)
\end{aligned} \tag{25}$$

The basic n-th degree formulas in the *assembly* case, from the Figure 1, are:

$$\overset{(l,0)}{Q_A}(n) \triangleq \overset{(l,0)}{Q_a}(1) \sum_{i=1}^{n} \overset{(l,0)}{k_i} \, \overset{(l,0)}{Q_i}(1) \tag{26}$$

$$\overset{(l,0)}{O_A}(n) \triangleq \overset{(l,0)}{Q_a}(1) \min \left\{ \min_{i=1,\ldots,n} \left\{ \frac{\overset{(l,0)}{O_i}(1)}{\overset{(l,0)}{k_i}} \right\}, \frac{\overset{(l,0)}{O_a}(1)}{\overset{(l,0)}{Q_a}(1)} \right\} \tag{27}$$

With a tree similar to (25) we have to impose the following recurrent 2nd degree expressions to let the *O* and *Q* operators be simultaneously satisfied in the equivalences:

$$\overset{(l,0)}{Q_A}(n) = \overset{(l,0)}{Q_a}(1) \left[\overset{(l,1)}{k_{n-1}} \overset{(l,1)}{Q_A}(n-1) + \overset{(l,0)}{k_n} \overset{(l,0)}{Q_n}(1) \right] \tag{28}$$

$$\begin{gathered}
\overset{(l+g,g+1)}{Q_A}(m) = \overset{(l+g+1,g+1)}{Q_a}(1) \left[\overset{(l+g+1,g+2)}{k_{m-1}} \overset{(l+g+1,g+2)}{Q_A}(m-1) + \overset{(l+g+1,g+1)}{k_m} \overset{(l+g+1,g+1)}{Q_m}(1) \right] \\
\textit{with} \quad m = 3,\ldots,n-1;\ g = 0,\ldots,n-m-1
\end{gathered} \tag{29}$$

$$\overset{(l+g,g+1)}{Q_A}(2) = \overset{(l+g+1,g+1)}{Q_a}(1) \left[\overset{(l+g+1,g+1)}{k_1} \overset{(l+g+1,g+1)}{Q_1}(1) + \overset{(l+g+1,g+1)}{k_2} \overset{(l+g+1,g+1)}{Q_2}(1) \right] \quad ;\ \textit{with}\ g = n-3 \tag{30}$$

$$\overset{(l,0)}{O_A}(n) = \overset{(l,0)}{Q_a}(1) \min \left\{ \frac{\overset{(l,1)}{O_A}(n-1)}{\overset{(l,1)}{k_{n-1}}}, \frac{\overset{(l,0)}{O_n}(1)}{\overset{(l,0)}{k_n}}, \frac{\overset{(l,0)}{O_a}(1)}{\overset{(l,0)}{Q_a}(1)} \right\} \tag{31}$$

$$\begin{gathered}
\overset{(l+g,g+1)}{O_A}(m) = \overset{(l+g+1,g+1)}{Q_a}(1) \min \left\{ \frac{\overset{(l+g+1,g+2)}{O_A}(m-1)}{\overset{(l+g+1,g+2)}{k_{m-1}}}, \frac{\overset{(l+g+1,g+1)}{O_m}(1)}{\overset{(l+g+1,g+1)}{k_m}}, \frac{\overset{(l+g+1,g+1)}{O_a}(1)}{\overset{(l+g+1,g+1)}{Q_a}(1)} \right\} \\
\textit{with}\ m = 3,\ldots,n-1;\ g = 0,\ldots,n-m-1
\end{gathered} \tag{32}$$

$$\overset{(l+g,g+1)}{O_A}(2) = \overset{(l+g+1,g+1)}{Q_a}(1) \min\left\{ \frac{\overset{(l+g+1,g+1)}{O_1}(1)}{\overset{(l+g+1,g+1)}{k_1}}, \frac{\overset{(l+g+1,g+1)}{O_2}(1)}{\overset{(l+g+1,g+1)}{k_2}}, \frac{\overset{(l+g+1,g+1)}{O_a}(1)}{\overset{(l+g+1,g+1)}{Q_a}(1)} \right\} \quad ; with\ g = n-3 \tag{33}$$

In the *assembly* case (and then as we will see in the *expansion*) new constraints are under consideration, due to the *k* variable:

$$\sum_{i=1}^{n} \overset{(l,0)}{k_i} = 1 \tag{34}$$

$$\overset{(l+g,g+1)}{k_{m-1}} + \overset{(l+g,g)}{k_m} = 1 \quad ; with\ m = 3,...,n;\ g = n-m \tag{35}$$

$$\overset{(l+g,g)}{k_1} + \overset{(l+g,g)}{k_2} = 1 \quad ; with\ \ g = n-2 \tag{36}$$

By developing (31)-(33), using the *min* operator associativity and comparing with (26)-(27), we obtain an equality of the form:

$$\min\{\alpha_1, \alpha_2, ..., \alpha_n, \beta_a\} = \min\left\{\alpha_1^{'}, \alpha_2^{'}, ..., \alpha_n^{'}, \beta_1^{'}, \beta_2^{'}, ..., \beta_{n-1}^{'}\right\} \tag{37}$$

Sufficient conditions for (37) are:

$$\alpha_i = \alpha_1^{'};\ \ \beta_a = \beta_i^{'} \quad i = 1,...,n \tag{38}$$

From the equalities in (38) we obtain:

$$\overset{(l+g,g)}{O_m}(1) = \frac{\overset{(l,0)}{O_m}(1)}{\overset{(l,0)}{k_m}} \overset{(l+g,g)}{k_m} \frac{\prod_{i=0}^{g-1} \overset{(l+i,i+1)}{k_{n-i-1}}}{\prod_{i=1}^{g} \overset{(l+i,i)}{Q_a}(1)} \quad ; with \quad \begin{array}{l} m = 3,...,n-1 \\ g = n - \max(m,2) \end{array} \tag{39}$$

$$\overset{(l+g,g)}{O_a}(1) = \frac{\overset{(l,0)}{O_a}(1)}{\overset{(l,0)}{Q_a}(1)} \overset{(l+g,g)}{Q_a}(1) \frac{\prod_{i=0}^{g-1} \overset{(l+i,i+1)}{k_{n-i-1}}}{\prod_{i=1}^{g} \overset{(l+i,i)}{Q_a}(1)} \quad ; with \quad g = 1,...,n-2 \tag{40}$$

To determine the remaining expressions of the operators' transformations we use the (28)-(30) in comparison with (26)-(27), (35), and (36). At this point, it must be noted that many possible solutions are available as the equivalences generate an underdetermined problem. Among the possible solutions, a natural one is to assume:

$$\overset{(l+g,g)}{Q_m}(1) = \overset{(l,0)}{Q_m}(1) \quad ; with \quad \begin{array}{l} m = 1,2,...,n-1 \\ g = n - \max(m,2) \end{array} \tag{41}$$

$$\overset{(l+g,g)}{k_m} = \frac{\overset{(l,0)}{k_m}(1)}{\sum_{i=1}^{\max(m,2)} \overset{(l,0)}{k_i}} \quad ; with \quad \begin{matrix} m = 1, 2, ..., n-1 \\ g = n - \max(m, 2) \end{matrix} \tag{42}$$

$$\overset{(l+g,g+1)}{k_m} = \frac{\sum_{i=1}^{m} \overset{(l,0)}{k_i}}{\sum_{i=1}^{m+1} \overset{(l,0)}{k_i}} \quad ; with \quad \begin{matrix} m = 2, 3, ..., n-1 \\ g = n - m - 1 \end{matrix} \tag{43}$$

$$\overset{(l+g,g)}{Q_a}(1) = 1 \quad ; with \quad g = 1, ..., n-2 \tag{44}$$

At this point we have all the necessary relationships between the n-th and the 2nd degrees *assembly* systems. Note also that with the use of (41)-(44), the (39) and (40) can be simplified in more practical expressions as follows:

$$\overset{(l+g,g)}{O_m}(1) = \overset{(l,0)}{O_m}(1) \quad , \quad \overset{(l+g,g)}{O_a}(1) = \frac{\overset{(l,0)}{O_a}(1)}{\overset{(l,0)}{Q_a}} \sum_{i=1}^{n-g} \overset{(l,0)}{k_i} \quad ; with \quad \begin{matrix} m = 1, 2, ..., n-1 \\ g = n - \max(m, 2) \end{matrix} \tag{45}$$

Expansion Structure

An example of 4th degree tree transformation for the *expansion* structure is provided in Figure 9, following the Newick notation:

$$\begin{aligned}
\overset{(l,0)}{S_E}(4) &\triangleq \overset{(l,0)}{S_e}(1)\left(\overset{(l,0)}{S_1}(1) : \overset{(l,0)}{k_1}, \overset{(l,0)}{S_2}(1) : \overset{(l,0)}{k_2}, \overset{(l,0)}{S_3}(1) : \overset{(l,0)}{k_3}, \overset{(l,0)}{S_4}(1) : \overset{(l,0)}{k_4} \right) \\
&\equiv \overset{(l,0)}{S_e}(1)\left(\overset{(l,1)}{S_E}(3) : \overset{(l,1)}{k_3}, \overset{(l,0)}{S_4}(1) : \overset{(l,0)}{k_4} \right) \\
&\equiv \overset{(l,0)}{S_e}(1)\left(\overset{(l+1,1)}{S_e}(1) : \overset{(l,1)}{k_3} \left(\overset{(l+1,1)}{S_1}(1) : \overset{(l+1,1)}{k_1}, \overset{(l+1,1)}{S_2}(1) : \overset{(l+1,1)}{k_2}, \overset{(l+1,1)}{S_3}(1) : \overset{(l+1,1)}{k_3} \right), \overset{(l,0)}{S_4}(1) : \overset{(l,0)}{k_4} \right) \\
&\equiv \overset{(l,0)}{S_e}(1)\left(\overset{(l+1,1)}{S_e}(1) : \overset{(l,1)}{k_3} \left(\overset{(l+1,2)}{S_E}(2) : \overset{(1+1,2)}{k_2}, \overset{(l+1,1)}{S_3}(1) : \overset{(l+1,1)}{k_3} \right), \overset{(l,0)}{S_4}(1) : \overset{(l,0)}{k_4} \right) \\
&\equiv \overset{(l,0)}{S_e}(1)\left(\overset{(l+1,1)}{S_e}(1) : \overset{(l,1)}{k_3} \left(\overset{(l+2,2)}{S_e}(1) : \overset{(l+1,2)}{k_2} \left(\overset{(l+2,2)}{S_1}(1) : \overset{(l+2,2)}{k_1}, \overset{(l+2,2)}{S_2}(1) : \overset{(l+2,2)}{k_2} \right), \overset{(l+1,1)}{S_3}(1) : \overset{(l+1,1)}{k_3} \right), \overset{(l,0)}{S_4}(1) : \overset{(l,0)}{k_4} \right)
\end{aligned} \tag{46}$$

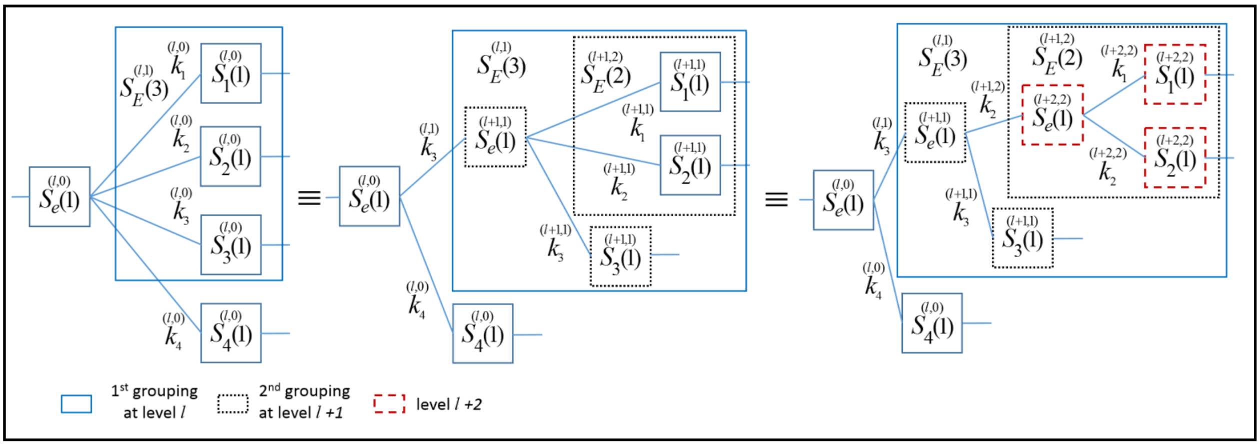


Figure 9. 4th degree equivalences for the expansion structure.

The basic expressions for the *expansion* structure from Figure 1 are:

$$\overset{(l,0)}{Q}_E(n)=\sum_{i=1}^{n}\overset{(l,0)}{k_i}\;\overset{(l,0)}{Q_i}(1) \tag{47}$$

$$\overset{(l,0)}{O}_E(n)=\sum_{i=1}^{n}\min\left\{\overset{(l,0)}{k_i}\;\overset{(l,0)}{Q_i}(1)\,\overset{(l,0)}{O_e}(1),\overset{(l,0)}{O_i}(1)\right\} \tag{48}$$

As usual, due to the self-similarity of a tree similar to (46), the following constraints can be imposed:

$$\overset{(l,0)}{Q}_E(n)=\overset{(l,1)}{k}_{n-1}\,\overset{(l,1)}{Q}_E(n-1)+\overset{(l,0)}{k_n}\;\overset{(l,0)}{Q_n}(1) \tag{49}$$

$$\overset{(l+g,g+1)}{Q}_E(m)=\overset{(l+g+1,g+2)}{k}_{m-1}\;\overset{(l+g+1,g+2)}{Q}_E(m-1)+\overset{(l+g+1,g+1)}{k_m}\;\overset{(l+g+1,g+1)}{Q_m}(1)\quad ;\,with\quad m=3,\dots,n-1;\ g=0,\dots,n-m-1 \tag{50}$$

$$\overset{(l+g,g+1)}{Q}_E(2)=\overset{(l+g+1,g+1)}{k_1}\;\overset{(l+g+1,g+1)}{Q_1}(1)+\overset{(l+g+1,g+1)}{k_2}\;\overset{(l+g+1,g+1)}{Q_2}(1)\quad ;\,with\ g=n-3 \tag{51}$$

$$\overset{(l,0)}{O}_E(n)=\min\left\{\overset{(l,1)}{k}_{n-1}\overset{(l,1)}{Q}_E(n-1)\overset{(l,0)}{O_e}(1),\overset{(l,1)}{O}_E(n-1)\right\}+\min\left\{\overset{(l,0)}{k_n}\;\overset{(l,0)}{Q_n}(1)\overset{(l,0)}{O_e}(1),\overset{(l,0)}{O_n}(1)\right\} \tag{52}$$

$$\begin{aligned}\overset{(l+g,g+1)}{O}_E(m)=&\min\left\{\overset{(l+g+1,g+2)}{k}_{m-1}\;\overset{(l+g+1,g+2)}{Q}_E(m-1)\;\overset{(l+g+1,g+1)}{O_e}(1)\;,\overset{(l+g+1,g+2)}{O}_E(m-1)\right\}\\&+\min\left\{\overset{(l+g+1,g+1)}{k_m}\;\overset{(l+g+1,g+1)}{Q_m}(1)\;\overset{(l+g+1,g+1)}{O_e}(1)\;,\;\overset{(l+g+1,g+1)}{O_m}(1)\right\}\quad ;\,with\ \begin{matrix}m=3,\dots,n-1\\ g=0,\dots,n-m-1\end{matrix}\end{aligned} \tag{53}$$

$$\overset{(l+g,g+1)}{O_E}(2) = \min\left\{ \overset{(l+g+1,g+1)}{k_1} \overset{(l+g+1,g+1)}{Q_1}(1) \overset{(l+g+1,g+1)}{O_e}(1) \ , \ \overset{(l+g+1,g+1)}{O_1}(1) \right\} + \min\left\{ \overset{(l+g+1,g+1)}{k_2} \overset{(l+g+1,g+1)}{Q_2}(1) \overset{(l+g+1,g+1)}{O_e}(1) \ , \ \overset{(l+g+1,g+1)}{O_2}(1) \right\} \quad ; with \quad g = n-3 \tag{54}$$

Using the (34)-(36) into (49)-(51), among many, the most natural constraints to impose are:

$$\overset{(l+g,g)}{Q_m}(1) = \overset{(l,0)}{Q_m}(1)\ , \quad \overset{(l+g,g+1)}{Q_E}(m) = \sum_{i=1}^{m} \frac{\overset{(l,0)}{k_i}}{\sum_{j=1}^{\max(i,2)} \overset{(l,0)}{k_j}} \overset{(l,0)}{Q_i}(1) \quad ; with \quad \begin{matrix} m = 1,2,...,n-1 \\ g = n - \max(m,2) \end{matrix} \tag{55}$$

Next step is to develop (38)-(40) using again (52)-(54), (55), and the following property of the *min* operator repeatedly:

$$\min\{a,b\} + c = \min\{a+c, b+c\} \tag{56}$$

Eventually the following remaining sufficient conditions for the (56) are obtained:

$$\overset{(l+g,g)}{O_e}(1) = \overset{(l,0)}{O_e}(1) \sum_{i=1}^{n-g} \overset{(l,0)}{k_i} \quad ; with \quad g = 1,...,n-2 \tag{57}$$

$$\overset{(l+g,g)}{O_m}(1) = \min\left\{ \overset{(l,0)}{k_m} \overset{(l,0)}{Q_m}(1) \overset{(l,0)}{O_e}(1), \overset{(l,0)}{O_m}(1) \right\} \quad ; with \quad \begin{matrix} m = 1,...,n-1 \\ g = n - \max(m,2) \end{matrix} \tag{58}$$

The (55), (57)-(58) determine an expression of a n n-th degree system into nested expansions of the 2nd degree.

Discussion on Implementation Details and Future Work

A prompt use of the previous expressions of the n-th degree is not straightforward if not aided by some automation. While the formulas in Figure 1 can be still useful to detect a bottleneck in an n-th degree system, the correct amount of increment on a variable is difficult to be determined. By decomposing in 2^{nd} degree sub-trees the tables in Appendix can provide the constraints, the gains, and so effective and efficient decisions on next actions on a system. Until a problem can be conveniently modeled with a 2nd degree HMT, the computations and the associated algorithm are prone even to a manual procedure. From the 3rd degree, things get quickly more complicated—though not more complex. As previously observed, the passing from the 2nd degree to n-th degree is not unique and not isomorphic, and it does not provide the possibility to stabilize an interpretation of the level decompositions into the 2nd degree. The decomposition into the 2nd degree changes at every change of the parameters of the higher level. In general, a new decomposition into 2nd degree trees is obtained at each change of the n-th degree variables due to the improvement actions. Besides, the most problematic thing is that in the decomposition the physical meaning of the improvement actions is lost. Incrementing a variable with index different from *(l,0)* means to act on an abstract system. Thus, every decision of increment on a variable with index *(l+g,g)* must be materialized back into actions over variables with *(l,0)* index. In general, this is straightforward with the formulas obtained in the previous section, apart from the (58). For the expansion case, a desired increment of O_m at index *(l+g,g)* might be realized by an implicit not invertible function of k_m, Q_m, O_e, and O_m at index *(l,0)*. This presents a difficulty in the implementation due to the loss of isomorphism provoked by the *min* operator. Besides, the extent and tractability of this difficulty can be taken under control if some constraints can be meaningfully added among the involved variables at level *(l,0)* depending on the initial modeling of the n-th degree system.

A situation similar to the problem in (58) appears for the handling of the improvements of the *k* variable. By looking at the relationships imposed in (34)-(36), there are infinite ways (or policies) in which the modification of *k* at a certain level affects all the *k* in the decomposition. A modification of *k* has to happen under the constraints given by (34)-(36) and the conditions in the tables provided in Appendix that constraint the available range of change of each the *k* in the 2nd degree abstract sub-trees.

Future work should test some heuristics that help in the choice of a generic and a simple implementation that overcomes the discussed problems. Nevertheless, we expect that the Occam's razor principle will be in act to prefer the simplest among the solutions. Moreover, future work in complementary direction might involve other techniques of the decomposition here obtained that preserve the isomorphism between the higher and the lower degrees. This would bypass the aforementioned problems, but the expectation is that they are to be traded off with a complication to the same extent of the decomposition expressions.

A third way remains also available as the use of degrees greater than the second can be intentionally avoided in many cases. The decision to tackle a modeling of a system with a 2nd degree tree with respect to taking advantage of n-th degree expressions is prone to some arbitrary decision. A trade-off is faced between deepening the levels of 2nd degree trees and, at the same time, modifying the tree structures and their interpretation, or leaving a level composed of more units in case the degree and the number of nodes model more naturally parts of the system. Actually, the technique of decomposition in the previous section relies on some abstract automation of the decomposition, which avoids the reinterpretation of the 2nd order nested structures, making it implicitly. In this case, the effort of interpretation of a more granular 2nd degree representation is traded for some automation and the addition of some general constraints over the system as discussed before. Anyway, the step from 2nd to higher degrees does not come for free, although in this paper some efficient ways of take the problems under tractability have been suggested. In the case studies so far conducted by the authors, we stopped the system representation to the 2nd degree. Thanks to the present work, we have a new stronger modeling and managing tool. By giving the opportunity to use n-th degree systems representations, and then automate their computing, the step of modeling is in general easier and more natural, both for humans and machines.

The formulas of section 3 have been used in particular to implement a distributed and autonomic computing system for the improvement of industrial processes (Bonci, 2018a). The monitoring and computing units, seen as holons, can be spread across the process in the form of a network of tiny computing CPS actors (Bonci, 2018d). In particular, the network is used for communications between the holons and its connections to determine the hierarchy in a holarchy, a specific organization of holons. Theoretically, any holon can be described recursively by a holarchy until the desired granularity level description is reached (Calabrese et al., 2010; Bonci, 2018f). The lower granularity in the presented method is the leaf units (cells) where OEE is evaluated. Nonetheless, the roles of the holons can dynamically evolve and re-assemble into a holarchy depending on the knowledge flow that, at any level, is necessary to accomplish the system goal (Calabrese et al.,

2010) and in response to unexpected situations. Indeed, the applicability of this method goes beyond the classical production structures. The ontological and semantic interpretation of a cell or a whole structure depends on the interpretation of a productivity measurement with respect to a goal. The nature and the meaning of the goal are local to each holon in the holarchy. It can express at the same time process flows and machines, as treated explicitly and formally in (Indri, 2018).

Conclusion

This paper has presented the detailed definition and discussion of a technique that has been deemed for the first time here as Holonic Management Tree (HMT). The technique is focused on the continuous performance improvement over self-similar system structures that aim to model and to control problems where complexity is an unavoidable characteristic. The systems under control are typically the hierarchies of tasks and goals that can be associated to a process and an organization of a business. The four fundamental structures in this first instantiation of HMT are the constituents of dynamic holarchies that are constructed towards production goals determined at different levels of the system of system. The HMT constitutes the core element for the automation of the management in relation to the findings and the foundations of management cybernetics. The strength of the technique resides in its continuous pursuing of simplicity, as an effective tool against complexity. In this paper there are developed the expressions valid for all the trees of fundamental productive units up to the n-th degree. The expressions obtained allow transforming an n-th degree problem into recursively equivalent 2nd degree problems, for which a complete and determined computational procedure has been provided. The technique here exposed brought about the n-th degree extension for an already well known method that let 2nd degree nested structures be easily computed at low cost. The formulas here used should be checked for an even possible simpler representation. With the HMT tree-based representation of the business system, a recursive computing for the improvement of the effectiveness of productive components and of the overall system can be performed recursively. This simplification enables large-scale implementations with the aid of lightweight computing devices that communicate the sequence of the corrective actions both to humans and to holonic sub-systems able to actuate them autonomously depending on the context.

While the automation technique here proposed represents an alternative that simplifies the problem of optimization in complex systems, the major open problem remains the research on the possible automation for the acquisition of the knowledge and the model representation, in association to the interpretations that can be specified on the HMT structures. In this sense, HMT has been contextualized as a seminal tool for a broader framework, currently under development, about the management, design, modeling and control of cyber-physical systems of systems seen as viable systems.

References

Abbass, H. A., Petraki, E., Merrick, K., Harvey, J., & Barlow, M. (2016). Trusted autonomy and cognitive cyber symbiosis: Open challenges. *Cognitive Computation, 8*(3), 385-408.

Abbott, R. (2006). Open at the top; open at the bottom; and continually (but slowly) evolving. In *System of Systems Engineering, 2006 IEEE/SMC International Conference on System of Systems Engineering*. Los Angeles, CA, USA.

Albertazzi, L. (2010). The ontology of perception. In *Theory and applications of ontology: Philosophical perspectives* (pp. 177-206). Dordrecht: Springer.

Antomarioni, S., Bevilacqua, M., & Ciarapica, F. E. (2018). More sustainable performances through lean practices: A case study. In *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), IEEE* (pp. 1-8). Retrieved from https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8423590

Badinelli, R., Barile, S., Ng, I., Polese, F., Saviano, M., & Di Nauta, P. (2012). Viable service systems and decision making in service management. *Journal of Service Management, 23*(4), 498-526.

Beer, S. (1981). *Brain of the firm* (2nd ed.). London: Chichester Wiley.

Bevilacqua, M., Ciarapica, F. E., & De Sanctis, I. (2017). Lean practices implementation and their relationships with operational responsiveness and company performance: An Italian study. *International Journal of Production Research, 55*(3), 769-794.

Bonci, A., Pirani, M., & Longhi, S. (2016). A database-centric approach for the modeling, simulation and control of cyber-physical systems in the factory of the future. *IFAC-PapersOnLine, 49*(12), 249-254.

Bonci, A., Pirani, M., & Longhi, S., (2017a). Robotics 4.0: Performance improvement made easy. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1-8.

Bonci, A., Pirani, M., & Longhi, S. (2017b).An embedded database technology perspective in cyber-physical production systems. *Procedia Manufacturing*, 11, 830-837.

Bonci, A., Pirani, M., & Longhi, S. (2018a). Tiny Cyber-Physical Systems for Performance Improvement in the Factory of the Future. *IEEE Transactions on Industrial Informatics, 15*(3), 1598-1608.

Bonci, A., Pirani, M., Mansanta, C, & Longhi, S. (2018b). Performance Improvement in CPSs over Self-similar System Structures. *IFAC-PapersOnLine, 51*(11), 570-575.

Bonci, A., Pirani, M., Cucchiarelli, A., Carbonari, A., Naticchia, B., & Longhi, S. (2018c). A Review of Recursive Holarchies for Viable Systems in CPSs. In *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 37-42.

Bonci, A., Pirani, M., & Longhi, S. (2018d). A Database-Centric Framework for the Modeling, Simulation, and Control of Cyber-Physical Systems in the Factory of the Future. *Journal of Intelligent Systems*, 27(4), 659-679.

Bonci, A., Pirani, M., Bianconi, C., & Longhi, S. (2018e).RMAS: Relational Multiagent System for CPS Prototyping and Programming. In *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 1-6.

Bonci, A., Pirani, M., Carbonari, A., Naticchia, B., Cucchiarelli, A., &Longhi, S. (2018f). Holonic overlays in cyber-physical system of systems. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1, 1240-1243.

Burgin, M. (2006). *Super-recursive algorithms*. Berlin/Heidelberg, Germany: Springer Science & Business Media.

Burgin, M. (2016). *Theory of knowledge: Structures and processes (Vol. 5)*. Singapore: World Scientific.

Cadsby, T. (2014). *Closing the mind gap: Making smarter decisions in a hypercomplex world*. Canada and the US: BPS Books.

Calabrese, M., Amato, A., Di Lecce, V., & Piuri, V. (2010). Hierarchical-granularity holonic modeling. *Journal of Ambient Intelligence and Humanized Computing, 1*(3), 199-209.

Carbonari, A., Messi, L., Naticchia, B., Vaccarini, M., & Pirani, M. (2018) Development of a BIM-based cyber-physical system for facility management of buildings. In *Proceedings of the Creative Construction Conference, 2018*.

Carter, B. (2016). Systems theory based architecture framework for complex system governance (Dissertation, Old Dominion University).

Conant, R. C., & Ross Ashby, W. (1970). Every good regulator of a system must be a model of that system. *International Journal of Systems Science, 1*(2), 89-97.

Indri, M., Trapani, S., Bonci, A., & Pirani, M. (2018). Integration of a production efficiency tool with a general robot task modeling approach. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1, 1273-1280.

Koestler, A. (1970). Beyond atomism and holism—the concept of the holon. *Perspectives in Biology and Medicine, 13*(2), 131-154.

Maciag, R. (2018). Discursive space and its consequences for understanding knowledge and information. *Philosophies, 3*(4), 34.

Mella, P. (2009). *The holonic revolution: Holons. Holarchies and holonic networks: The ghost in the production machine*. Pavia: Scientifica.

Mella, P. (2014). T*he magic ring. Systems thinking approach to control systems*. New York, NY and Berlin: Springer Verlag.

Mella, P., & Gazzola P. (2015). Capitalistic firms as cognitive intelligent and explorative agents. The Beer's VSM and Mella's most views. *Management Dynamics in the Knowledge Economy, 3*(4), 645-674.

Muthiah, K. M. N., & Huang, S. H. (2007). Overall throughput effectiveness (OTE) metric for factory-level performance monitoring and bottleneck detection. *International Journal of Production Research, 45*(20), 4753-4769.

Nagorny, K., Scholze, S., Ruhl, M., & Colombo, A. W. (2018). Semantical support for a CPS data marketplace to prepare big data analytics in smart manufacturing environments. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)* (pp. 206-211). Saint-Petersburg, Russia, May 15-18, 2018.

Nechansky, H. (2010). The relationship between: Miller's living systems theory and Beer's viable systems theory. *Systems Research and Behavioral Science, 27*(1), 97-112.

Olsen, G. (1990). Newick's 8:45. Tree format standard. Retrieved from http://evolution.genetics.washington.edu/phylip/newick_doc.html

Pearl, J., Glymour, M., & Jewell, N. P. (2016). *Causal inference in statistics: A primer*. Hoboken, NJ: John Wiley & Sons.

Pearl, J., & Mackenzie, D. (2018). *The book of why: The new science of cause and effect*. NY: Basic Books.

Pirani, M., Bonci, A., & Longhi S. (2016). A scalable production efficiency tool for the robotic cloud in the fractal factory. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, 6847-6852.

Pirani, M., Messi, L., Carbonari, A., Bonci, A., & Vaccarini, M. (2018). Holonic Management Systems for Resilient Operation of Buildings. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, 35, 1-8.

Poli, R., & Obrst, L. (2010). Ontology: The categorial stance. In *Theory and applications of ontology: Philosophical perspectives* (pp. 1-22). Dordrecht: Springer.

Senge, P. M. (2014). *The fifth discipline fieldbook: Strategies and tools for building a learning organization*. US: Crown Business.

Stadnicka, D., Pirani, M., Bonci, A., Ratnayake, R. C., & Longhi, S. (2017a). Self-similar Computing Structures for CPSs: A case study on pots service process. In *Working Conference on Virtual Enterprises*, 157-166.

Stadnicka, D., Bonci, A., Pirani, M., & Longhi, S. (2017b). Information management and decision making supported by an intelligence system in kitchen fronts control process. In *International Conference on Intelligent Systems in Production Engineering and Maintenance*, 249-259.

Steinhaeusser, T., Elezi, F., Tommelein, I. D., & Lindemann, U. (2015). Management cybernetics as a theoretical basis for lean construction thinking. *Lean Construction Journal, 2015*, 1-14.

Vahidi, A., Aliahmadi, A., & Teimoury, E. (2018). Researches status and trends of management cybernetics and viable system model. *Kybernetes*. Retrieved from https://doi.org/10.1108/K-11-2017-0433

Valckenaers, P., & Van Brussel, H. (2016). *Design for the unexpected: From holonic manufacturing systems towards a humane mechatronics society*. Oxford: Butterworth-Heinemann.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., & Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry, 37*(3), 255-274.

VDI/VDE Society Measurement and Automatic Control. (2015). Status report: Reference Architecture Model Industries 4.0 (RAMI 4.0).

Wang, J. Q., Chen, J., Zhang, Y., & Huang, G. Q. (2016). Schedule-based execution bottleneck identification in a job shop. *Computers & Industrial Engineering, 98*, 308-322.

Weiser, M., & Brown, J. S. (1997). The coming age of calm technology. In *Beyond calculation* (pp. 75-85). New York: Springer.

Appendix A

The following tables describe in the condition/action behaviour expected from the performance improvement algorithm.

Note that actions with OTE gain equal to 0 are still useful to move the system out of the stable state conditions and possibly store OTE improvement potential for the next steps.

Table A1

Condition-Action Table for Series Systems

Structure type: SERIES OTE = $\min\{O_1Q_2, O_2\}$			
Conditions		Next improvement actions; maximum increments available	Expected OTE Gain
s1	$O_1Q_2 < O_2$	$\Delta O_1 = \min\left\{\frac{O_2}{Q_2}, 1\right\} - O_1$	$\Delta O_1 Q_2$
		$\Delta Q_2 = \min\left\{\frac{O_2}{O_1}, 1\right\} - Q_2$	$O_1 \Delta Q_2$

Table A1 to be continued

s2	$O_1Q_2 = O_2$	$\Delta O_1 = 1 - O_1$ $\Delta O_2 = 1 - O_2$ $\Delta Q_2 = 1 - Q_2$	0 0 0
s3	$O_1Q_2 > O_2$	$\Delta O_2 = O_1Q_2 - O_2$	ΔO_2

Table A2

Condition-Action Table for Parallel Systems

Structure type: PARALLEL OTE = $\frac{O_1 + O_2}{2}$			
Conditions		Next improvement actions; maximum increments available	Expected OTE Gain
p1	$\frac{O_1 + O_2}{2} \le 1$	$\Delta O_1 = 1 - O_1$ $\Delta O_2 = 1 - O_2$	$\Delta O_1 / 2$ $\Delta O_2 / 2$

Table A3

Condition-Action Table for Assembly Systems

Structure type: ASSEMBLY OTE = $\Delta k_2 = k_1 - \frac{O_1}{O_e Q_1}$			
Conditions		Next improvement actions; maximum increments available	Expected OTE Gain
a1	$\frac{O_1}{k_1} < \frac{O_2}{k_2} < \frac{O_a}{Q_a}$	$\Delta O_1 = \min\left\{\frac{k_1}{k_2}O_2, 1\right\} - O_1$	$\frac{\Delta O_1 Q_a}{k_1}$
		$\Delta O_2 = \min\left\{\frac{k_2 O_a}{Q_a}, 1\right\} - O_2$	0
		$\Delta k_2 = \min\left\{\frac{k_1 O_2 - k_2 O_1}{O_1 + O_2}, k_1\right\}$	$Q_a\left[\min\left\{\frac{O_1}{k_1 - \Delta k_2}, \frac{O_2}{k_2 + \Delta k_2}, \frac{O_a}{Q_a}\right\} - \frac{O_1}{k_1}\right]$
		$\Delta Q_a = \min\left\{k_2\frac{O_a}{O_2}, 1\right\} - Q_a$	$(Q_a + \Delta Q_a)\min\left\{\frac{O_1}{k_1}, \frac{O_a}{Q_a + \Delta Q_a}\right\} - \frac{Q_a O_1}{k_1}$
a2	$\frac{O_1}{k_1} = \frac{O_2}{k_2} < \frac{O_a}{Q_a}$	$\Delta O_1 = \min\left\{k_1\frac{O_a}{Q_a}, 1\right\} - O_1$	0
		$\Delta O_2 = \min\left\{k_2\frac{O_a}{Q_a}, 1\right\} - O_2$	0
		$\Delta Q_a = \min\left\{\frac{O_a}{O_1 + O_2}, 1\right\} - Q_a$	$(Q_a + \Delta Q_a)\min\left\{\frac{O_1}{k_1}, \frac{O_a}{Q_a + \Delta Q_a}\right\} - \frac{Q_a O_1}{k_1}$
a3	$\frac{O_1}{k_1} < \frac{O_2}{k_2} = \frac{O_a}{Q_a}$	$\Delta O_1 = \min\left\{\frac{k_1}{k_2}O_2, k_1\frac{O_a}{Q_a}, 1\right\} - O_1$	$\frac{\Delta O_1 Q_a}{k_1}$
		$\Delta k_2 = \min\left\{\frac{k_1 O_2 - k_2 O_1}{O_1 + O_2}, k_1\right\}$	$Q_a\left[\min\left\{\frac{O_1}{k_1 - \Delta k_2}, \frac{O_2}{k_2 + \Delta k_2}, \frac{O_a}{Q_a}\right\} - \frac{O_1}{k_1}\right]$
		$\Delta Q_a = \min\left\{k_1\frac{O_a}{O_1}, \frac{k_2}{O_2}, 1\right\} - Q_a$	$(Q_a + \Delta Q_a)\min\left\{\frac{O_1}{k_1}, \frac{O_a}{Q_a + \Delta Q_a}\right\} - \frac{Q_a O_1}{k_1}$

Table A3 to be continued

a4	$\frac{O_1}{k_1} < \frac{O_a}{Q_a} < \frac{O_2}{k_2}$	$\Delta O_1 = \min\left\{k_1 \frac{O_a}{Q_a}, 1\right\} - O_1$	$\frac{\Delta O_1 Q_a}{k_1}$
		$\Delta k_2 = \min\left\{k_1 - O_1 \frac{Q_a}{O_a}, O_2 \frac{Q_a}{O_a} - k_2, k_1\right\}$	$Q_a \left[\min\left\{\frac{O_1}{k_1 - \Delta k_2}, \frac{O_2}{k_2 + \Delta k_2}, \frac{O_a}{Q_a}\right\} - \frac{O_1}{k_1}\right]$
		$\Delta O_a = \min\left\{\frac{O_2 Q_a}{k_2}, 1\right\} - O_a$	0
		$\Delta Q_a = \min\left\{k_1 \frac{O_a}{O_1}, \frac{k_2}{O_2}, 1\right\} - Q_a$	$(Q_a + \Delta Q_a) \min\left\{\frac{O_1}{k_1}, \frac{O_a}{Q_a + \Delta Q_a}\right\} - \frac{Q_a O_1}{k_1}$
a5	$\frac{O_1}{k_1} = \frac{O_a}{Q_a} < \frac{O_2}{k_2}$	$\Delta O_1 = \min\left\{\frac{k_1}{k_2} O_2, 1\right\} - O_1$	0
		$\Delta k_2 = \min\left\{\frac{k_1 O_2 - k_2 O_1}{O_1 + O_2}, \frac{Q_a}{O_a} O_2 - k_2, k_1\right\}$	0
		$\Delta O_a = \frac{O_2 Q_a}{k_2} - O_a$	0
		$\Delta Q_a = \min\left\{\frac{k_2}{O_2}, 1\right\} - Q_a$	0
a6	$\frac{O_2}{k_2} < \frac{O_1}{k_1} < \frac{O_a}{Q_a}$	$\Delta O_1 = \min\left\{\frac{k_1 O_a}{Q_a}, 1\right\} - O_1$	0
		$\Delta O_2 = \min\left\{\frac{k_2}{k_1} O_1, 1\right\} - O_2$	$\frac{\Delta O_2 Q_a}{k_2}$
		$\Delta k_1 = \min\left\{\frac{k_2 O_1 - k_1 O_2}{O_1 + O_2}, k_2\right\}$	$Q_a \left[\min\left\{\frac{O_1}{k_1 + \Delta k_1}, \frac{O_2}{k_2 - \Delta k_1}, \frac{O_a}{Q_a}\right\} - \frac{O_2}{k_2}\right]$
		$\Delta Q_a = \min\left\{\frac{k_1 O_a}{O_1}, 1\right\} - Q_a$	$(Q_a + \Delta Q_a) \min\left\{\frac{O_2}{k_2}, \frac{O_a}{Q_a + \Delta Q_a}\right\} - \frac{Q_a O_2}{k_2}$
a7	$\frac{O_2}{k_2} < \frac{O_1}{k_1} = \frac{O_a}{Q_a}$	$\Delta O_1 = \min\left\{\frac{k_1 O_a}{Q_a}, 1\right\} - O_1$	0
		$\Delta O_2 = \min\left\{\frac{k_2}{k_1} O_1, 1\right\} - O_2$	$\frac{\Delta O_2 Q_a}{k_2}$
		$\Delta k_1 = \min\left\{\frac{k_2 O_1 - k_1 O_2}{O_1 + O_2}, k_2\right\}$	$Q_a \left[\min\left\{\frac{O_1}{k_1 + \Delta k_1}, \frac{O_2}{k_2 - \Delta k_1}, \frac{O_a}{Q_a}\right\} - \frac{O_2}{k_2}\right]$
		$\Delta Q_a = \min\left\{\frac{k_2 O_a}{O_2}, \frac{k_1}{O_1}, 1\right\} - Q_a$	$(Q_a + \Delta Q_a) \min\left\{\frac{O_2}{k_2}, \frac{O_a}{Q_a + \Delta Q_a}\right\} - \frac{Q_a O_2}{k_2}$
a8	$\frac{O_2}{k_2} < \frac{O_a}{Q_a} < \frac{O_1}{k_1}$	$\Delta O_2 = \min\left\{k_2 \frac{O_a}{Q_a}, 1\right\} - O_2$	$\frac{\Delta O_2 Q_a}{k_2}$
		$\Delta k_1 = \min\left\{k_2 - \frac{Q_a}{O_a} O_2, \frac{Q_a}{O_a} O_1 - k_1, k_2\right\}$	$Q_a \left[\min\left\{\frac{O_1}{k_1 + \Delta k_1}, \frac{O_2}{k_2 - \Delta k_1}, \frac{O_a}{Q_a}\right\} - \frac{O_2}{k_2}\right]$
		$\Delta O_a = \frac{Q_a O_1}{k_1} - O_a$	ΔO_a
		$\Delta Q_a = \min\left\{\frac{k_2 O_a}{O_2}, \frac{k_1}{O_1}, 1\right\} - Q_a$	$(Q_a + \Delta Q_a) \min\left\{\frac{O_2}{k_2}, \frac{O_a}{Q_a + \Delta Q_a}\right\} - \frac{Q_a O_2}{k_2}$

Table A3 to be continued

a9	$\frac{O_2}{k_2} = \frac{O_a}{Q_a} < \frac{O_1}{k_1}$	$\Delta O_2 = \min\left\{\frac{k_2}{k_1} O_1, 1\right\} - O_2$	0
		$\Delta k_1 = \min\left\{\frac{k_2 O_1 - k_1 O_2}{O_1 + O_2}, \frac{Q_a}{O_a} O_1 - k_1, k_2\right\}$	0
		$\Delta O_a = \frac{Q_a O_1}{k_1} - O_a$	ΔO_a
		$\Delta Q_a = \min\left\{\frac{k_1}{O_1}, 1\right\} - Q_a$	0
a10	$\frac{O_a}{Q_a} < \frac{O_1}{k_1} < \frac{O_2}{k_2}$	$\Delta O_1 = \min\left\{\frac{k_1}{k_2} O_2, 1\right\} - O_1$	0
		$\Delta k_1 = \min\left\{\frac{Q_a O_1}{O_a} - k_1, k_2 - Q_a O_2, k_2\right\}$	0
		$\Delta k_2 = \min\left\{\frac{k_1 O_2 - k_2 O_1}{O_1 + O_2}, k_1\right\}$	0
		$\Delta O_a = \frac{Q_a O_1}{k_1} - O_a$	ΔO_a
a11	$\frac{O_a}{Q_a} < \frac{O_1}{k_1} = \frac{O_2}{k_2}$	$\Delta O_1 = \min\left\{\frac{k_1}{Q_a}, 1\right\} - O_1$	0
		$\Delta O_2 = \min\left\{\frac{k_2}{Q_a}, 1\right\} - O_2$	0
		$\Delta k_1 = \min\left\{\frac{Q_a O_1}{O_a} - k_1, k_2 - Q_a O_2, k_2\right\}$	0
		$\Delta k_2 = \min\left\{\frac{Q_a O_2}{O_a} - k_2, k_1 - Q_a O_1, k_1\right\}$	0
		$\Delta O_a = \frac{Q_a O_1}{k_1} - O_a$	ΔO_a
a12	$\frac{O_a}{Q_a} < \frac{O_2}{k_2} < \frac{O_1}{k_1}$	$\Delta O_2 = \min\left\{\frac{k_2}{k_1} O_1, 1\right\} - O_2$	0
		$\Delta k_1 = \min\left\{\frac{k_2 O_1 - k_1 O_2}{O_1 + O_2}, k_2\right\}$	0
		$\Delta k_2 = \min\left\{\frac{Q_a O_2}{O_a} - k_2, k_1 - Q_a O_1, k_1\right\}$	0
		$\Delta O_a = \frac{Q_a O_2}{k_2} - O_a$	ΔO_a
a13	$\frac{O_1}{k_1} = \frac{O_2}{k_2} = \frac{O_a}{Q_a}$	$\Delta O_1 = \min\left\{\frac{k_1}{Q_a}, 1\right\} - O_1$	0
		$\Delta O_2 = \min\left\{\frac{k_2}{Q_a}, 1\right\} - O_2$	0
		$\Delta O_a = 1 - O_a$	0
		$\Delta Q_a = \min\left\{\frac{k_1}{O_1}, \frac{k_2}{O_2}, 1\right\} - Q_a$	0

Table A4
Condition-Action Table for Expansion Systems

Structure type: EXPANSION

OTE = $k_1Q_1 \min\left\{\frac{O_1}{k_1Q_1}, O_e\right\} + k_2Q_2 \min\left\{\frac{O_2}{k_2Q_2}, O_e\right\}$

Conditions		Next improvement actions; maximum increments available	Expected OTE Gain
e1	$\frac{O_1}{k_1Q_1} < O_e \wedge \frac{O_2}{k_2Q_2} < O_e$	$\Delta O_1 = \min\{k_1O_eQ_1, 1-O_2\} - O_1$	ΔO_1
		$\Delta O_2 = \min\{k_2O_eQ_2, 1-O_1\} - O_2$	ΔO_2
		$\Delta k_1 = k_2 - \frac{O_2}{O_eQ_2}$	0
		$\Delta k_2 = k_1 - \frac{O_1}{O_eQ_1}$	0
e2	$\frac{O_1}{k_1Q_1} < O_e \wedge O_e < \frac{O_2}{k_2Q_2}$	$\Delta O_1 = \min\{k_1O_eQ_1, 1-k_2O_eQ_2\} - O_1$	ΔO_1
		$\Delta k_2 = \min\left\{k_1 - \frac{O_1}{O_eQ_1}, \frac{O_2}{O_eQ_2} - k_2, \frac{1-O_1}{O_eQ_2} - k_2\right\}$	$\Delta k_2O_eQ_2$
		$\Delta O_e = \min\left\{\frac{O_2}{k_2Q_2}, \frac{1-O_1}{k_2Q_2}, 1\right\} - O_e$	$k_2\Delta O_eQ_2$
		$\Delta Q_2 = \min\left\{\frac{O_2}{k_2O_e}, \frac{1-O_1}{k_2O_e}, 1\right\} - Q_2$	$k_2O_e\Delta Q_2$
e3	$\frac{O_1}{k_1Q_1} < O_e \wedge \frac{O_2}{k_2Q_2} = O_e$	$\Delta O_1 = \min\{k_1O_eQ_1, 1-O_2\} - O_1$	ΔO_1
		$\Delta O_2 = 1 - O_1 - O_2$	0
		$\Delta k_2 = \min\left\{\frac{1-O_1}{O_eQ_2} - k_2, k_1\right\}$	0
		$\Delta O_e = \min\left\{\frac{1-O_1}{k_2Q_2}, 1\right\} - O_e$	0
		$\Delta Q_2 = \min\left\{\frac{1-O_1}{k_2O_e}, 1\right\} - Q_2$	0
e4	$O_e < \frac{O_1}{k_1Q_1} \wedge \frac{O_2}{k_2Q_2} < O_e$	$\Delta O_2 = \min\{k_2O_eQ_2, 1-k_1O_eQ_1\} - O_2$	ΔO_2
		$\Delta k_1 = \min\left\{k_2 - \frac{O_2}{O_eQ_2}, \frac{O_1}{O_eQ_1} - k_1, \frac{1-O_2}{O_eQ_1} - k_1\right\}$	$\Delta k_1O_eQ_1$
		$\Delta O_e = \min\left\{\frac{1-O_2}{k_1Q_1}, \frac{O_1}{k_1Q_1}, 1\right\} - O_e$	$k_1\Delta O_eQ_1$
		$\Delta Q_1 = \min\left\{\frac{O_1}{k_1O_e}, \frac{1-O_2}{k_1O_e}, 1\right\} - Q_1$	$k_1O_e\Delta Q_1$
e5	$O_e < \frac{O_1}{k_1Q_1} \wedge O_e < \frac{O_2}{k_2Q_2}$	$\wedge\ Q_1 \geq Q_2$ $\Delta k_1 = \min\left\{\frac{O_1}{O_eQ_1} - k_1, k_2\right\}$	$\Delta k_1O_e(Q_1 - Q_2)$
		$\wedge\ Q_1 \leq Q_2$ $\Delta k_2 = \min\left\{\frac{O_2}{O_eQ_2} - k_2, k_1\right\}$	$\Delta k_2O_e(Q_2 - Q_1)$
		$\Delta O_e = \min\left\{\frac{O_1}{k_1Q_1} - O_e, \frac{O_2}{k_2Q_2} - O_e, 1 - O_e\right\}$	$\Delta O_e(k_1Q_1 + k_2Q_2)$
		$\Delta Q_1 = \min\left\{\frac{O_1}{k_1O_e} - Q_1, 1 - Q_1\right\}$	$k_1O_e\Delta Q_1$
		$\Delta Q_2 = \min\left\{\frac{O_2}{k_2O_e} - Q_2, 1 - Q_2\right\}$	$k_2O_e\Delta Q_2$

Table A4 to be continued

e6	$O_e < \frac{O_1}{k_1Q_1} \wedge \frac{O_2}{k_2Q_2} = O_e$	$\Delta O_2 = 1 - O_2$	0
		$\wedge \quad Q_1 \geq Q_2 \qquad \Delta k_1 = \min\left\{\frac{O_1}{O_eQ_1} - k_1, k_2\right\}$	$\Delta k_1 O_e (Q_1 - Q_2)$
		$\Delta O_e = \min\left\{\frac{O_1}{k_1Q_1} - O_e, 1 - O_e\right\}$	$k_1 \Delta O_e Q_1$
		$\Delta Q_1 = \min\left\{\frac{O_1}{k_1O_e} - Q_1, 1 - Q_1\right\}$	$k_1 O_e \Delta Q_1$
		$\Delta Q_2 = 1 - Q_2$	0
e7	$\frac{O_1}{k_1Q_1} = O_e \wedge \frac{O_2}{k_2Q_2} < O_e$	$\Delta O_1 = 1 - O_1$	0
		$\Delta O_2 = \min\{k_2O_eQ_2, 1 - O_1\} - O_2$	ΔO_2
		$\Delta k_1 = k_2 - \frac{O_2}{O_eQ_2}$	0
		$\Delta O_e = 1 - O_e$	0
		$\Delta Q_1 = 1 - Q_1$	0
e8	$\frac{O_1}{k_1Q_1} = O_e \wedge O_e < \frac{O_2}{k_2Q_2}$	$\Delta O_1 = 1 - O_1$	0
		$\wedge \quad Q_1 \leq Q_2 \qquad \Delta k_2 = \min\left\{\frac{O_2}{O_eQ_2} - k_2, k_1\right\}$	$\Delta k_2 O_e (Q_2 - Q_1)$
		$\Delta O_e = \min\left\{\frac{O_2}{k_2Q_2} - O_e, 1 - O_e\right\}$	$k_2 \Delta O_e Q_2$
		$\Delta Q_1 = 1 - Q_1$	0
		$\Delta Q_2 = \min\left\{\frac{O_2}{k_2O_e} - Q_2, 1 - Q_2\right\}$	$k_2 O_e \Delta Q_2$
e9	$\frac{O_1}{k_1Q_1} = O_e \wedge \frac{O_2}{k_2Q_2} = O_e$	$\Delta O_1 = 1 - O_1$	0
		$\Delta O_2 = 1 - O_2$	0
		$\Delta O_e = 1 - O_e$	0
		$\Delta Q_1 = 1 - Q_1$	0
		$\Delta Q_2 = 1 - Q_2$	0