30th Annual INCOSE international symposium
Cape Town, South Africa
July 18 - 23, 2020

Orchestrating Human Systems Integration
Looking for the Right Mix for Human-Machine Teaming

Guy André Boy[1,2]
[1] CentraleSupélec, Paris Saclay University, 9 rue Joliot Curie, 91190 Gif-sur-Yvette, France
[2] ESTIA Institute of Technology, 92 allées Théodore Monod, 64210 Bidart, France
+33 6 73 11 79 38
guy-andre.boy@centralesupelec.fr & g.boy@estia.fr

Abstract. Human systems integration (HSI) is commonly thought as the association of human-centered design (HCD) and systems engineering (SE). HCD relies on human-in-the-loop simulation (HITLS) and artificial intelligence (AI). In addition, AI and SE terminologies, methods, and tools should be harmonized in the context of human-machine teaming (HMT). Evolutions from single-agent to multi-agent approaches in AI, and from isolated-system to system-of-systems in SE are comparable. System and agent representations commonly apply to humans and machines. They are defined by their structures and functions. Based on research currently developed on HCD of increasingly-autonomous complex systems, this paper uses the Orchestra metaphor model that supports HMT organization design and management, based on: domain ontology (music theory); tasks and designers (scores and composers); activity and performance coordination (conductors); human/machine operators (musicians); end-users and consumers (audience). This approach requires elicitation, understanding and mastery of new interdependences, co-adaptation, and integration of agents' emerging functions using HITLS.

Introduction

This paper emphasizes relationships between three major concepts: integration, orchestration and teaming. In current industrial projects and programs, integration is often performed too late. It is time to better consider integration during the whole life cycle of a system, and even more importantly in sociotechnical systems where people and organizations matter. The design of technology, organizations and people's jobs or activities need to be better orchestrated in terms of common frame of reference (metaphorically, a music theory), tasks (scores), engineering designers (composers), performance coordinators (conductors), manufacturing personnel (musicians), and end users (audience). In addition, such multi-agent system is a human-machine system, where machines have physical and cognitive functions and structures as people have. This is the reason why there is a need for a fundamental framework that conciliates disciplines such as systems engineering, artificial intelligence, human-computer interaction and ergonomics, aggregated into a field called human systems integration (HSI). Finally, since both human and machine agents (or systems) are evolving toward more autonomy, it is time to address the human-machine teaming at the same time.

Human systems integration: why is it so crucial?

Human systems integration (HSI) emerged as an approach to considering the human element in the design and management of complex sociotechnical systems in the beginning of the 21st century (Figure 1). Several communities have attempted to consider people in engineering design. Human factors and ergonomics (HFE) started after World War 2 considering physical ergonomics. Human-computer interaction (HCI) started in the 1980s when personal computers became part of our everyday lives. HFE was mainly based on activity analysis before a complex system was designed and built, working on existing system use, and after the system was built to improve usability. Consequently, human factors specialists were always fighting with engineers to make them credible for the implementation of significant changes. They were always very late in the design and development processes. HCI mainly focused on interaction design, considering task analysis as a major technique in computing design. The question came when HCI started to be applied to large complex systems, such as commercial aircraft. At that time, we did more than adding a user interface; we started to separate physical technology and actions on the real world from people in charge of the overall performance of these large complex systems by adding computers in between.

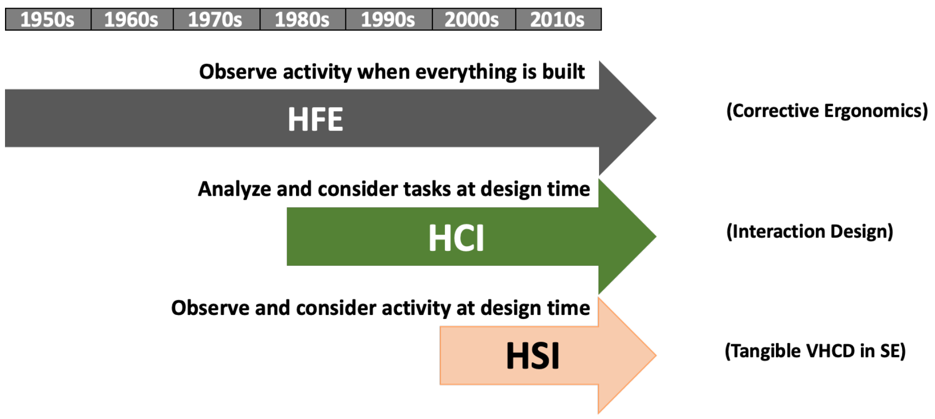

1950s
1960s
1970s
1980s
1990s
2000s
2010s
Observe activity when everything is built
HFE
(Corrective Ergonomics)
Analyze and consider tasks at design time
HCI
(Interaction Design)
Observe and consider activity at design time
HSI
(Tangible VHCD in SE)

Figure 1. Evolution of human-centered disciplines with respect to tasks and activity.

The term "interactive cockpits" significantly denotes the fact that pilots interact with computers (i.e., HCI) and no longer directly with mechanical devices that impact on aerodynamics and propulsion. More generally, massive incorporation of software in our everyday lives induced the establishment of human-centered design (HCD), which fostered usefulness and usability engineering (Nielsen, 1993) and the development of ISO 9241-210:2010(E) standard. At the same time, even if participatory design was recently developed within the HCI community, we should recognize that aeronautical engineering design was, and still is, based on participatory work with experimental test pilots. Participatory design requires everyone to understand each other, and more specifically a common language, based on the various concepts of the domain, should be defined to deal with complex systems. This language should provide common understanding of information to be shared by the various human and machine systems (or agents) involved in the overall system. It should be evolutionary because it is impossible to get a definitive ontology but a stabilized explicit ontology. This language should support analysis, design, and evaluation of complex systems.

In artificial intelligence (AI), an ontology is defined by the set of concepts of a domain of interest, their properties and the relations (links) among the various concepts (Musen, 1992; Gruber, 1993a), in the same way a terminology is defined by a set of terms (i.e., like a dictionary). Cognitive science defines a concept by associating a term to it and its meaning (i.e., context that removes possible ambiguities). More generally, "ontology is the philosophical discipline that tries to find out what there is: what entities make up reality, what is the stuff the world is made from?" (Hofweber, 2005). Concepts can be incrementally elicited in many ways: (1) from stories told by subject matter experts or analysis of questionnaires; (2) from observations of people and systems involved in the domain at stake; (3) from brainstorming sessions associating domain stakeholders and HCD teams. This process is typically called knowledge elicitation or knowledge acquisition from experts (Gaines, 2013).

Current sociotechnical systems are complex because they are extremely interconnected, and we always need to find out what people's roles are or should be within these systems. We are interconnected between people and technology, but also between people through technology. It is therefore important and timely to improve our understanding of what these systems are really about – this is what the next section of the paper is about. We also need to better understand how AI and systems engineering (SE) can be harmonized and work together. Why? This is because systems are becoming more autonomous, and therefore have to be more appropriately coordinated. Autonomy is important because we want to cover a maximum of situations that include very well-known situations as well as potentially unexpected situations. Automation has been developed using knowledge of very well-known situations. This is the reason why automation turns out to involve rigidity in operational activities dealing with unexpected situations (i.e., outside specific well-known contexts). In those situations, people require flexibility, creativity, and appropriate knowledge. They require autonomy! Autonomy needs to be thought out for both people and machines, as well as for levels of coordination among agents, whether they are humans or machines. Who is coordinating? Is coordination self-emerging from interactions among agents? Scenarios and HITLS need to be developed to support activity analyses, and further considerations of tangibility (Boy, 2020). Two tangibility meanings should be articulated in the design and management of complex sociotechnical systems: physical tangibility (e.g., grasping a physical object); and figurative tangibility (e.g., grasping a concept or an abstraction).

Developing an ontology of a (human and machine) system of systems

Why should we develop an ontology? According to Noy and McGuinness (2019), reasons are usually the followings: sharing common understanding of information structure among people or software agents; enabling reuse of domain knowledge; making domain assumptions explicit; separating domain knowledge from the operational knowledge; analyzing domain knowledge. Developing an ontology is performing an exploration, and an analysis of the domain we want to study. Basically, we want to improve our understanding of the various domain-entity types (e.g., executant, team, supervisor, mediator, coordinator), types of relationships among these entities (e.g., delegation, cooperation, authority, processing, temporal, spatial), and the various situations that make sense to be considered. We could say that an ontology is a conceptual model of the domain it represents (i.e., the way we see the domain) that should be tractable by a computer program.

The use of ontology is the development of systems is not new. Gruber proposed five design principles and criteria for ontologies whose purpose is knowledge sharing and interoperation among programs based on a shared conceptualization (Gruber, 1993b). Let's propose an adaptation of these five principles. (1) Clarity expressed in the form of effective and objective terms and relations among them (i.e., keep intended meaning). All definitions should be documented with natural language (NL). (2) Coherence between formally-defined concepts and links among concepts, and their definitions (i.e., what you see is what you get – inferences should keep consistency between formal concepts and definitions expressed in natural language). (3) Extendibility of the ontology in such a way that enables an HCD team to extend and/or specialize it monotonically (i.e., new terms and links could be added easily). (4) Minimal encoding bias that guaranties that the ontology should not be [too much] dependent on the way it is coded (i.e., conceptualization should be specified at the knowledge level without depending on a particular symbol-level encoding). (5) Minimal ontological commitment to support the intended knowledge sharing activities (i.e., an ontology should make as few claims as possible about the world being modeled – it should be easy to specialize and instantiate an ontology).

What is meant by NL? First of all, in order to remove any ambiguity, NL is used for knowledge acquisition from expert, and is not at all advised for subsequent interaction with the machine. It is domain-centered NL. Knowledge acquisition from experts is a matter of knowledge design. Design is a matter of making compromises or tradeoff, and domain NL is the best way to handle appropriate and correct discussions among subject matter experts. The Group Elicitation Method was developed, and is currently used to handle such group knowledge acquisition (Boy, 1997, 2020). Ontology design is not different. Compromises among the above-mentioned principles have to be made. "For example, … clarity criterion talks about definitions of terms, whereas ontological commitment is about the conceptualization being described. Having decided that a distinction is worth making, one should give the tightest possible definition of it" (Gruber, 1993b).

All five principles are not discussed in this paper, but only coherence and extendibility. The reader should go back to Gruber's original articles for the others. Coherence can be further expanded into lexical, syntactic, semantic, and pragmatic coherence. Lexical coherence is about terms that are used to denote concepts within the developed ontology. When a concept is denoted by two terms, then there is no lexical coherence. We often say that it is ambiguous. In natural language, we try to vary the use of words in order to avoid boring repetitions. However, in technical writing, it is better to use the same words to denote the same concepts in order to avoid confusion. When terms are formed of multiple words, aliases are fine and often used; this is common practice. Syntactic coherence is about sequences of terms. For example, on a desktop user interface, it is typical to see on the top menu bar, "File – Edit – View …", under "File", we have "New – Open – Open Recent …" If a new application is created, and the sequence of options is "Open – New – Open Recent …", then there is no syntactic coherence. Semantic incoherence is about concepts that are denoted by the same term. For example, the concept of "table" can denote a "table of contents" in a book or a physical table on which you can eat. When this is the case, you need to add context around the term to eliminate the potential confusion. For example, if you say "table of contents" or "the table in the book", we are likely to understand that it is a figurative table, and not a physical table. If you say "people seating around the table", we are likely to understand that it is a physical table. Pragmatic incoherence is about a concept that has different meanings in different cultures or ethnical groups. Extendibility is related to coherence and clarity. Indeed, by adding, removing or reformulating a concept in an ontology, we need to make sure consistency is insured as well as the various distinctions are meaningful and sustainable.

There is no industry standard used of the development and use of an ontology. We have been using associations of BPMN (White, 2004; White & Bock, 2011), Cognitive Function Analysis (Boy, 1998), and CMap tools[1]. We are currently working on the PRODEC method that integrates these procedural and declarative methods (Boy et al., to appear). For a system-of-systems (SoS) ontology, it is typically easy to define an initial structure of structures, as well as allocating functions of functions onto it. Down in this article, a generic system representation is presented, which could serve for further standardization. It is however important to introduce the Orchestra model before.

From the Old Army model to the Orchestra model

This paper advocates the metaphorical shift from the Old Army Model (OAM) to the Orchestra Model (Boy, 2013). Most industrial organizations work according to the OAM (i.e., a general at the top, then officers, and soldiers at the bottom). OAM information flow is vertical, and mostly top-down. Soldiers do not or barely exchange information horizontally. They are executants of low-level tasks. Decisions are made at the top. For the last few decades, horizontal information flow exists supported by information technology (e.g., phones, emails, Internet). This kind of information-

[1] https://cmap.ihmc.us.

technology-based horizontalization happens anywhere, and more specifically in industrial organizations. Consequently, the very concept of OAM has to evolve. In addition, the soldiers of the past are now highly specialized. They have become experts in a given field of practice. They even often belong to communities of practice. The overall organization concept shifts toward the Orchestra Model.

For an orchestra to be effective, and enable playing a symphony, all musicians are required to have a common frame of reference, which is music theory (e.g., they know how to read and understand the meaning of scores). Who is writing scores? This is the dedicated job of composers. A composer of a symphony needs to articulate various scores among each other. He or she has to coordinate tasks of musicians of the orchestra. More generally, this is task coordination that enables an organization to make some kind of product. At performance time, the orchestra requires a conductor to synchronize musicians' activities. In other words, the conductor coordinates an SoS (i.e., an orchestra of musicians). Musicians themselves are autonomous systems or agents capable of playing their part perfectly, but need their activity to be coordinated with the other musicians. Again, activity may be very different from the task with respect to various contextual facets. In addition, musicians also need to cooperate among each other to insure a reasonable amount of stability, and resilience in case of a musical mistake by one of several musicians – "One for all! All for one!" Another part of the orchestra metaphor is the audience – music stakeholders (i.e., composers, conductors, and musicians) produce pieces of music for potential listeners! More generally, designers, crafters, and engineers generate products for targeted people (i.e., we often call them users or human operators). The audience brings issues such as acceptability, usability, and usefulness.

In the same way we have several types of orchestras (e.g., symphonic, jazz), current industrial organizations may take several forms (e.g., highly structured and large, loosely structured and small). Structured and large organizations are typically based on proceduralized functions (e.g., symphonic orchestra). Loosely-structured and small organizations are typically based on problem-solving functions (e.g., jazz band). In all cases, we are facing agencies of agents (i.e., SoSs or teams of teams), and the more agents are autonomous, the more the agency should be coordinated.

The concept of system

A system is a representation of a natural or artificial entity. For example, physicians talk about cardiovascular or neural systems; anthropologists talk about communities of people and social groups as organizational systems; and engineers talk about mechanical and civil engineering systems. Figure 2 presents a synthetic view of what a system is about. A system is recursively defined as including people, machines, and systems. A system has at least a structure and a function that can be physical and/or cognitive. In practice, a system has several structures and several functions articulated within structures of structures and functions of functions.

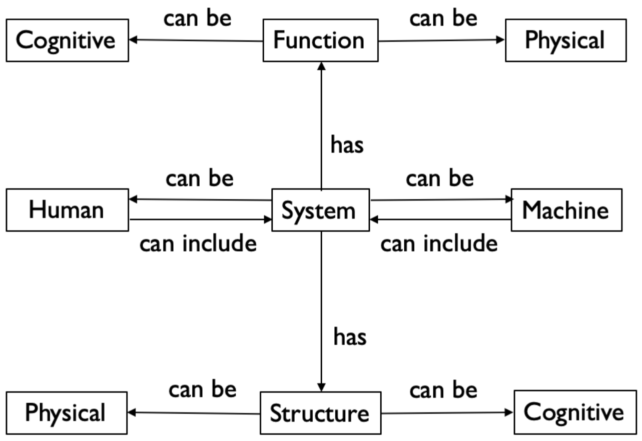


Figure 2. Synthetic view of the system representation.

A system of systems (SoS) is defined in the same way as Minsky (1986) defined an agent as a society of agents. Russell and Norvig (2010) defined an agent as an architecture (i.e., structure) and a program (i.e., function).

For a long time, engineers considered a system as an isolated system, or a quasi-isolated system. As for an agent in AI, which has sensors and actuators, a system in SE has sensors to acquire an input and actuators to produce an output (Figure 3).

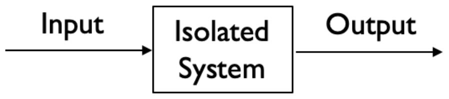


Figure 3. An isolated system.

In an SoS, each system is interconnected to other systems either statically (in terms of systems' structures) and/or dynamically (in terms of systems' functions).

Summarizing, a SoS is projected onto a structure of structures, usually called an infrastructure, where a network of functions could be allocated. It should be noted that the resulting network of functions is not necessarily a direct mapping on the related infrastructure (Figure 4).

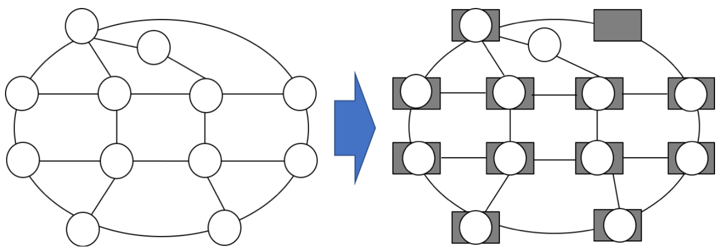


Figure 4. A function of functions mapped onto a structure of structures.

Emergent behaviors and properties. We need to make a distinction between deliberately established functions allocated onto an infrastructure and functions that necessarily emerge from system activity. Indeed, systems within a bigger system (i.e., an SoS) interact among each other to generate an activity. Bertalanffy (1968) said: "A System is a set of elements in interaction." Emerging functions are discovered from such an activity (Figure 5). The integration of such emergent functions into the SoS may lead to mandatory generation of additional structures, which we also call emerging structures.

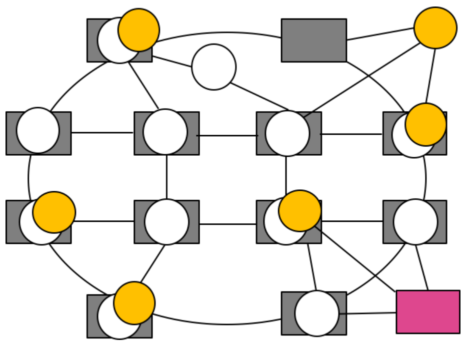


Figure 5. Emerging functions (yellow) and structures (pink) within an active system of systems.

Systems of systems properties. System's purpose is mainly defined by its task space (i.e., all tasks the system can perform successfully). Each task is performed by the system using a specific function that logically produces an activity that can be fully or partially observable (Figure 6).

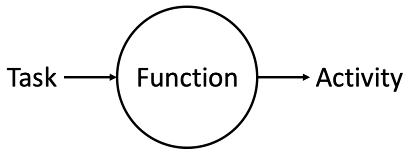


Figure 6. A function transforms a task into an activity.

A system's function is teleologically defined by three entities (Figure 7):

- its role within the system of systems (related to a task to be performed);
- its context of validity that frames the boundaries of system's performance; and
- its set of resources required to perform its role within its context of validity.

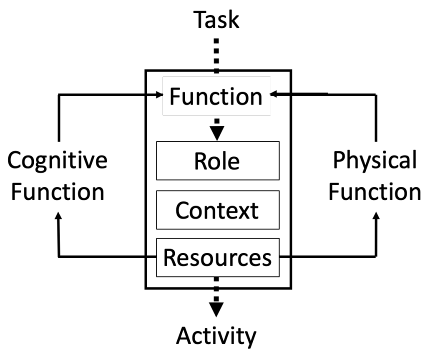


Figure 7. Combined logical/teleological definition of a function.

Resources are systems themselves that have their own cognitive and/or physical functions. System's functions can be recursively defined as functions of functions. They can be physical and/or cognitive.

System's functions include perception of the environment, inference of actions from percepts and execution of actions on the environment, producing an activity.

Let's consider a postman represented as a system with the function of delivering letters. The postman is part of an SoS, which is the Post. Actually, the role of this system is "delivering letters." Context of validity is, for example, seven hours a day five days a week (i.e., time-wise context), and a given neighborhood (i.e., space-wise context). Resources can be physical (e.g., a bicycle and a big bag) and cognitive (e.g., a pattern matching algorithm that enables the postman to match the name of the street, the number on the door and the name of the recipient). The corresponding pattern matching algorithm is a cognitive function. Let's consider now that there is a strike and most postmen are no longer available for delivering letters. Remaining postmen in duty should have longer hours of work and larger neighborhood, until this expansion is too extreme and postman's helpers are needed to achieve the delivery task. In this case, a tenure postman should have cognitive resources such as "training", "supervising", and "assessing" temporary personnel. We see that a cognitive function "delivering letters" owned by a postman (i.e., an agent or a system) has to be decomposed into several other functions to manage temporary postmen. We start to see an organization developed as an answer to a strike. This example shows how a function of functions can be distributed among a structure of structures.

The multi-dimensional concept of context

The development of a function analysis is based on the teleological definition of functions (see Figure 6). One of the major factors is "context of validity" of physical and/or cognitive functions being developed and analyzed. In HCD, context is a very difficult concept to grasp. Context is related to several concepts, such as situation, organization, culture, location, time, behavior, point of view, relationships among agents, discourse, dialogue, and so on. The difficulty that we usually have to define "context" mainly depends on the viewpoint that we have (e.g., whether being an engineer making a system or a human operator using the system). The following distinctions provide a framework that supports context definition (Figure 8): cognition/physical (i.e., what is generated by people, and what is not); design/operations (i.e., functioning logic versus use logic); and normal/abnormal (i.e., normal versus abnormal situations).

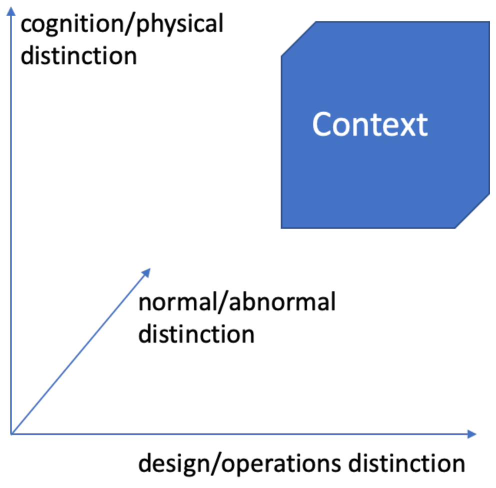

cognition/physical distinction
Context
normal/abnormal distinction
design/operations distinction

Figure 8. A three-dimensional framework for context definition.

Organizational context can be investigated as a socio-cognitive topology (i.e., cognitive structures of structures as described in the synthetic view of the system representation presented on Figure 1). For example, an orchestra is an organization that includes musicians and various kinds of instruments (technology), which are topologically interrelated to produce a coherent symphony (i.e., the product). We can say that a musician plays in the context of a specific orchestra, and a specific audience (i.e., he or she would not play the same in a different orchestra for a different audience). Situational context can be investigated as a physical topology (i.e., physical structures of structures of the location where the symphony is being performed).

Systemic interaction models and system's capabilities

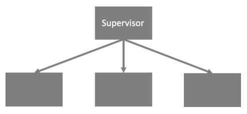
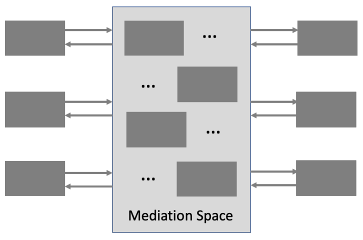
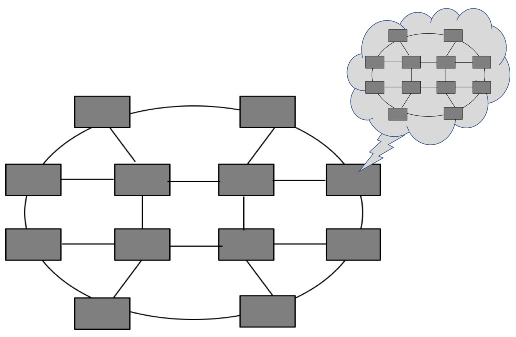
There are three main kinds of systemic interaction models (Table 1): supervision; mediation; and cooperation.

Supervision is when a system (i.e., a supervisor) supervises interactions among other systems. Supervision is about coordination. This interaction model is used when systems do not know each other or do not have enough resources to properly interact among each other toward a satisfactory performance of the SoS that they constitute.

Mediation is when systems are able to interact among each other through a mediation space composed of a set of mediating systems (i.e., like diplomats). This interaction model is used when systems barely know each other, but easily understand how to use the mediation space.

Cooperation is when systems are able to have a socio-cognitive model of the SoS which they are part of. Each system uses its socio-cognitive model to interact with the other systems to maximize some kinds of performance metrics. Note that this principle can be collective and democratic. Other principles could be used such as dominance of a system over the other systems (i.e., a dictatorial principle). Cooperation interaction model is used when systems know each other through their own socio-cognitive model, which is able to adapt through learning from positive and negative interactions. Up to now, cooperation is mostly a capability of humans. However, artificial intelligence (AI) brings new ways of providing machines with such cooperation capabilities. AI can provide situation awareness, decision-making and planning capabilities and support in specific contexts. In addition, AI can also provide machines with learning capabilities, and more specifically possibility of upgrading system's socio-cognitive model from experience, in specific contexts. Note that cooperation requires coordination, and situation awareness is key (Endsley, 2019; Stanton et al., 2017).

Table 1: Systemic interaction models.

Supervision of systems by a system	Mediation among systems through a mediation space	Systems cooperating among each other thanks to their knowledge of the others
		

Using these three systemic interaction models, it is clear that systems, as agents, become more autonomous when they go from being supervised to being mediated to cooperating among each other.

Conclusion

Human-centered design (HCD) benefits from observing activity in human-in-the-loop simulations (HITLS) based on incrementally-more-tangible prototypes. At this point, human-systems integration (HSI) can be summarized into two simple equations. HSI is the association of HCD and systems engineering (SE): HSI = HCD + SE; and HCD = agile process that includes task analysis + HITLS + activity analysis.

Human-machine teaming in complex multi-agent systems (also called systems of systems in SE) requires more efforts to be correctly modeled, and further measured. This paper provides a contribution of a draft ontology of a sociotechnical system of systems, using the Orchestra metaphor. It is based on the definition of a system as a representation of humans and/or machines that can be physical and/or cognitive, a multi-dimensional concept of context, and systemic interaction models. Our goal is to provide a framework for symbiotic integration of human and machine agents to make a livable sociotechnical system of systems.

References

Boy, G.A., Dezemery, J., Lu Cong Sang, R., Hein Makoto, A., Masson, D., Morel, C. & Villeneuve, E. (to appear), 'PRODEC: Combining Procedural and Declarative Knowledge for Human-Centered Design', White paper, FlexTech Chair, CentraleSupélec & ESTIA Institute of Technology, France.

Boy, G.A. 2020, *Human Systems Integration: From Virtual to Tangible*, CRC Press – Taylor & Francis Group, USA.

Boy, G.A. 2013, *Orchestrating Human-Centered Design*, Springer, UK.

Boy, G.A. 1998, *Cognitive Function Analysis*, Contemporary Studies in Cognitive Science and Technology Series, Ablex-Praeger Press, USA.

Boy, G.A. 1997, 'The Group Elicitation Method for Participatory Design and Usability Testing', *Interactions Magazine*, March issue, published by ACM Press, New York; also, in *Proceedings of CHI'96, the ACM Conference on Human Factors in Computing Systems 1996*, held in Vancouver, Canada.

Endsley, M. 2019, 'Human teaming with autonomous systems: Building support for situation awareness', *Keynote at INCOSE HSI2019 International Conference*, Biarritz, France.

Gaines, B.R. 2013, 'Knowledge acquisition: Past, present and future', *International Journal of Human-Computer Studies*, Volume 71, Issue 2, February: 135-156. doi>10.1016/j.ijhcs.2012.10.010.

Gruber, T.R. 1993a, 'A Translation Approach to Portable Ontology Specification', *Knowledge Acquisition* 5: 199-220.

Gruber, T.R. 1993b, 'Toward Principles for the Design of Ontologies Used for Knowledge Sharing', *International Journal Human-Computer Studies*, 43: 907-928. Also available as Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University, USA.

Hofweber, T. 2005, 'A puzzle about ontology', *Noûs*. 39 (2): 256–283 (retrieved on July 31, 2019: https://philpapers.org/rec/HOFAPA-2).

Minsky, M. 1986, *The Society of Mind*, Touchstone Book. Published by Simon & Schuster, New York, USA.

Musen, M.A. 1992, 'Dimensions of knowledge sharing and reuse', *Computers and Biomedical Research* 25: 435-467.

Nielsen, J. 1993, *Usability Engineering*, Morgan Kaufman, USA, ISBN 0-12-518406-9.

Noy, N.F. & McGuinness, D.L. 2019, *Ontology Development 101 – A Guide to Creating Your First Ontology*. Stanford University, Stanford, CA 94305 (retrieved on August 27, 2019: https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html).

Russel, S. & Norvig, P. 2010, *Artificial Intelligence – A Modern Approach*, Third Edition. Prentice Hall, Boston, USA. ISBN 978-0-13-604259-4.

Stanton, N., Salmon, M., Walker, G.H., Salas, E. & Hancock, P.A. 2017, 'State-of-Science: Situation Awareness in individuals, teams and systems', *Ergonomics*, 60(4): 247-258. DOI: 10.1080/00140139.2017.1278796.

Von Bertalanffy, L. 1968, *General System Theory: Foundations, Development, Applications*, New York: George Braziller.

White, S.A. 2004, Business Process Modeling Notation. Retrieved on November 23: https://web.archive.org/web/20130818123649/http://www.omg.org/bpmn/Documents/BPMN_V1-0_May_3_2004.pdf.

White, S.A. & Bock, C. 2011, *BPMN 2.0 Handbook Second Edition: Methods, Concepts, Case Studies and Standards in Business Process Management Notation*, Future Strategies Inc. ISBN 978-0-9849764-0-9.

Biography

Guy André Boy. FlexTech Chair Institute Professor at CentraleSupélec (Paris Saclay University) and ESTIA Institute of Technology, Fellow of the *Air and Space Academy,* and Chair of the Human-Systems Integration Working Group of International Council on Systems Engineering (INCOSE). He was University Professor and Dean, Human-Centered Design Institute, and founder of HCD Ph.D. & Master's Programs, at *Florida Institute of Technology*, and a Senior Research Scientist at the *Florida Institute for Human and Machine Cognition* (IHMC). He was Chief Scientist for Human-Centered Design at *NASA Kennedy Space Center.*