Systems **2014**, *2*, 273-296; doi:10.3390/systems2030273

OPEN ACCESS

systems

ISSN 2079-8954

www.mdpi.com/journal/systems

Article

Towards a Metamodel to Support the Joint Optimization of Socio Technical Systems

Paola Di Maio

Institute of Socio Technical Systems Research, Edinburgh EH16 5XR, UK;
E-Mails: pdimaio@istcs.org or paola.dimaio@gmail.com

*Received: 30 January 2014; in revised form: 22 May 2014 / Accepted: 6 June 2014 /
Published: 25 June 2014*

Abstract: Designing and implementing functional Socio Technical Systems (STS) is becoming increasingly important, as technologies become more pervasive and critical to everyday life. Socio technical systems are said to be efficient and useful when they are "jointly optimized" yet few system designers understand what joint optimization is, and how to achieve it. The paper explains the core tenets of Joint Optimization and identifies the need for artifacts to support the built in design of joint optimization in socio technical systems from the early stages of development. JOM (Joint Optimization Metamodel) is proposed, as a cognitive artifact to help conceptualize and model joint optimization, and five types of JO are identified resulting from conceptual evaluation of the metamodel.

Keywords: socio technical system; joint optimization; metamodel

1. Introduction

Technologies, especially information technologies are becoming more pervasive and embedded into everyday objects, as a result the complexity and size of systems which permeate all aspects of human life is increasing exponentially: dependencies, interconnections, and causal loops, typically hard to map, are even more challenging in the case of socio technical systems, to the point that many aspects of their development are considered "intractable" because they constantly change and evolve in response to conditions and demands [1]. There are no known mechanisms capable of mapping the interdependences in Socio Technical Systems (STS) because these are emergent and cannot be determined in advance. Given the requirement for flexible boundaries capable of rapid dynamic reconfiguration [2] and large scalable open and participatory communication networks which characterize contemporary information architectures, more comprehensive techniques to model and address requirements are becoming necessary for every type of system development. This paper identifies some key issues characterizing discourse in this domain, and proposes a metamodel, a type of cognitive artifact, to support the inherent development of jointly optimized socio technical systems.

2. Motivation

Recurring patterns of behaviors across different types of systems have been observed regularly throughout the history of science and engineering in many different fields from Leibniz to Ashby [3] however only relatively recently, with Bertalanffy, the field of systemics [4] emerged as a discipline in its own right, and a theory of socio technical systems was codified [5]. Today, socio technical theory permeates and drives, directly and indirectly, large portions of engineering, and has contributed to the emergence of socio technical systems development practice, referred to by some as socio technical systems engineering [6]. Given the size and complexity that characterize STS, and, most importantly, given the fact that each STS is "unique" [7], many efforts toward the standardization of methodologies and artifacts get stalled. There is strong emphasis in contemporary systems engineering upon "whole systems" approaches, self-organization, and resilience [8,9]. Among key issues are criticality, vulnerabilities and failures arising not from technical flaws but due to due to human factors [10], some of which are observable—such as behaviors and some are not—such as cognition. The new paradigm of Cognitive Computing has emerged from research as a "category of technologies that uses natural language processing and machine learning to enable people and machines to interact more naturally to extend and magnify human expertise and cognition" [11]. Since cognitive computing revolves around Natural Language Processing, the ability to devise and implement adequate knowledge representation structures and techniques is essential to systems development [12]. Arguments for embracing STS approaches are increasingly abundant in the engineering practice: "Infrastructure may be viewed as part of a continuous relationship with the human and natural environments" [13]. "It's the combination of complex engineering factors, with equally difficult human considerations, that makes complex socio technical so frustratingly difficult to address. Solving this problem requires a new way of looking at systems, one that takes the unpredictable nature of human behavior into account" [14]. "The engineer, trained and rewarded for technical excellence, is frequently frustrated with what are perceived as extra "social or environmental design constraints". However, far from constraints, broadening the social objectives of engineering presents opportunities for engineering service, if one makes the effort to look" [15].

A complete understanding of complex socio technical requires diving into the specifics of each system by adopting a domain-specific perspective. Research establishes the need for better theoretical understanding to help discriminate between what is relevant and what is superfluous in the description of socio technical systems [16]. In addition to the arguments found in literature cited above, essentially identifying a need for novel instruments to support the development of STS, observations carried out in fieldwork point to the need for streamlined, lightweight socio technical artifacts to support management, which is routinely caught in the cross fire between investors, shareholders and workforce.

3. Goal of the Research, Scope and Paper Organization

In our research (ISTCS.org) we concern ourselves with the design and implementation of information centric socio technical systems, and rely heavily on "models"—as in "model driven engineering"—using a variety of techniques and approaches. The goal of this research is to devise artifacts to support socio technical systems development. This paper discusses contemporary issues and core references in socio technical systems modeling and identifies the need for extended technical instruments to support and guide development to ensure that socio technical principles and core tenets are built into the system from inception. The Joint Optimization Metamodel (JOM) is presented and discussed together with an outline of plans for future work.

4. Method

A mixed method approach is adopted including observations from fieldwork and literature review. Shortfalls in existing models/metamodeling techniques and approaches to support joint optimization currently in use are identified, and provide justification for the metamodel. A *model driven engineering* (*MDE*) approach is loosely adopted to devise the proposed metamodel and associated constructs.

5. Socio Technical Systems

Sociotechnical systems (STS) are dynamic, adaptive systems resulting from the interaction of people using tools, techniques and knowledge (the technical subsystem) to produce a product or service. This definition considers organizations consisting of three interdependent subsystems: social, technical, and environmental. Each must be aligned and work together so the organization can function optimally [17]. STS theory was codified [18] in research of the mechanization of coal mines in Britain. Typically, the STS knowledge domain is by nature interdisciplinary: in organizational development and social sciences the focus is mostly on complex organizational work design that recognizes the interaction between people and technology, while in computer science and engineering is an approach to develop functional jointly optimized whole systems. STS approaches to systems design and engineering are becoming increasingly necessary: as research shows [19] critical system failures often occur for "non technical" reasons, despite flawless technical implementations, due to human (cognitive and behavioral) and socio technical factors. In most systems, functional performance depends on the interaction of people and technologies, whereby the correct functioning of systems depend largely on interactions with stakeholders. A sociotechnical system can be described as a complex system in which social and technical sub-systems influence one another through feedback loops [1], producing emergent system behavior, and where the relationship between social and technical components are constantly redefined and evolve adaptively and dynamically. "Whereas the traditional organization is maintained in a steady state only by the constant and arduous efforts of management, a jointly optimized one proves to be inherently stable and self correcting" [20] because a jointly optimized behaves as an open system, where equilibrium is reached thanks to free flow of inputs and outputs exchanged with "the environment" irrespective of its states, a phenomenon referred to as "self-synchronization" [21] where "…all aspects of organizational functioning are interrelated" [22].

6. Joint Optimization

Because the social and technical elements must work together to accomplish tasks, work systems produce both physical products and social/psychological outcomes. The key issue is to design work so that the two parts yield positive outcomes; this is called joint optimization [23].

The emphasis of STS is the inter relatedness of the systems components, to the point that they cannot be separated. "Joint optimization" (JO) refers to the notion that to achieve optimal performance a must configured to take into account the entanglement of its social, technical and environmental aspects. When the social and the technical subsystems are targeted separately, important dynamics of the ST system "as a whole" may be lost or misunderstood. When sub-systems are optimized in isolation (suboptimized), degradation of performance or even failure can be caused due to unforeseeable local interactions. An organization can perform optimally only if the social and technical dimensions are designed to fit the demands of each other and of the environment [24,25]. Trying to optimize the technical or social dimension alone will result in the suboptimization of the socio technical whole [20]. Since the socio technical systems functionality is optimized only when there is synergy and collaboration between people, technology and the environment, tasks and processes and the whole system design must be devised taking into account the three and their interactions. Achieving joint optimization is challenging necessary to benefit from the increased efficiency offered by STS *vs.* pure technical systems [23]. It remains complicated, due to the simultaneous need to address multiple systems and unpredictable, due to dynamic interrelationships. No single model or modeling technique exists to guide development to achieve joint optimization, however some of the underlying principles are:

(1) **Responsible autonomy** individuals/stakeholders take responsibility for the outcome/performance can make decisions autonomously;
(2) **Adaptability** tasks and schedules and overall work procedures can be adjusted by individual team members to increase optimization (distributed optimization *vs.* central optimization);
(3) **Meaningfulness of tasks** [26] including skill variety [27] task identity and "entire cycle of operations" or whole tasks [17];
(4) **Feedback loops** and recursive interactions [28].

Additional principles of good practices to facilitate joint optimization are the designing of organizational processes and components capable of evolving iteratively based on feedback, and the principle of minimal critical specification, which states that "While it may be necessary to be quite precise about what has to be done, it is rarely necessary to be precise about how it is done" [29] whereby designers specify only crucial relationships, functions, and controls, leaving the details open to iterative evolutions generated by usage.

7. About Models and Meta Models

Novel concepts and conceptual categories co-evolving rapidly, driven by the fast pace of technological change, make it increasingly challenging to identify and cluster each new set of artifacts that emerges from rapid socio technical evolutionary cycles into discrete classes. For example: "Object oriented designs, meant for programming design, often take on the distinct appearance of system models of physical networks and devices” [30]. In the definition of the Object Management Group “A model represents some concrete or abstract thing of interest, with a specific purpose in mind. The model is related to the thing by an explicit or implicit isomorphism. Models in the context of the MDA (Model Driven Architecture) Foundation Model are instances of metamodels and, therefore, consist of model elements and links between them” (OMG). A metaphor used to distinguish models from objects is that the former can represent a theory [31], while the latter correspond to artifacts as digital instantiations of the theory [32], with built in behaviors. A “model” in common terminology is inherently reflective; that is, to say you have a model is to say there exists something that you are modeling. Models exist to aid in making predictions about the outside reality. They carry no reality of their own. Models can be adjusted so they better predict reality based on the data to which they have access, and better infer information from the data they possess [33] “Metamodeling” is the construction of a collection of “concepts” (things, terms, *etc.*) within a certain domain. A model is an abstraction of phenomena in the real world; a metamodel is yet another abstraction, highlighting properties of the model itself. A model conforms to its metamodel in the way that a computer program conforms to the grammar of the programming language in which it is written. Metamodeling can be considered as an explicit description (constructs and rules) of how a domain-specific model is built. In particular, this comprises a formalized specification of the domain-specific notations. Typically, metamodels are—and always should follow—a strict rule set [34].

Different approaches exist to derive a metamodel [35]. In software engineering, several types of models and corresponding activities exist, for example MetaData Model and MetaProcess Model. The model proposed in this paper is devised following a hybrid approach that incorporates and modifies elements of various approaches. From the Meta Object Facility (MOF, [36]) a standard for model driven engineering the model proposed here finds overall justification for the requirement of a metamodel in the first place. The metaprocess model Software Process Engineering Metamodel (SPEM) justifies and guides the process metamodel for joint optimization proposed here and the content metamodel from the Open Group provides a reference for the core/extension architecture of JOM.

8. JOM—A Metamodel for the Joint Optimization of Socio Technical Systems

In systems engineering, various kinds of technical models and artifacts are used to guide critical aspects of development. The search for a model or a process to guide joint optimization of socio technical systems yields interesting related frameworks and models, although nothing definitive exists that satisfies the requirement for a lightweight framework of reference that can be used in agile, iterative design. In the following paragraphs some existing approaches are summarized, and where relevant, related aspects of these frameworks are incorporated in our proposed approach, described in the next section.

(1) **In Tracing the Roots: The Influence of Socio Technical Principles on Modern Organizational Change**, Munkvold [37] distinguishes between ***Social Aspects*** such as Attributes of people (attitudes, skills, values, *etc.*) their Relationships among people, Reward systems, Authority, structures and Technical Aspects, such as Processes, tasks, technology. In this approach, however, the two are still considered somewhat “separate”;

(2) **In Metamodel for Understanding, Analyzing, and Designing Sociotechnical Systems**, Alter [38] formulates a valid integrated metamodel expressed as a technical artifact (a series of diagrams capturing the core socio technical system's entities and processes). It doesn't however—surprisingly perhaps—directly address "joint optimization" as a requirement for effective STS and does not support JO as a continuum, which this research targets and emphasizes as an essential inextricable triad of entities in STS. Furthermore, the STS is viewed primarily as a "work system", while in our research STS are considered irrespective of whether the goal is productivity—as in Alters [39] work system—or not. Our approach incorporates some of the core entities and processes identified in this model, albeit rearranged and with a higher level of abstraction;

(3) **In Organizational Transformation a Model for Joint Optimization of Culture Change and Evidence Based Design**, Hamilton *et al.* [40] provide a narrative form a model for joint optimization, with focus on cultural change in the context of evidence based research, however the outcome of the analysis is not codified and does not yield a technical artifact that can be used by developers to guide practical implementations of joint optimization;

(4) **In Putting Action into Sociotechnical Systems Theory**—a proposed analysis of the Australian Film Industry Socio technical action regulation using START [41] draws from the People Technology Organization (PTO) framework. The model is domain specific with the focus on an "industry", that is a cluster of companies operating in the same field. It identifies four levels of organizations. The model proposed here does not draw from this approach at this stage;

(5) One of the frameworks of reference used to capture high level aspect of STS is found in Shaping socio technical System Innovation Strategies using a **Five Aspects Taxonomy** by Rhodes and Ross [42]. The Taxonomy identifies and models five aspects common to most socio technical systems: structural, behavioral, contextual, temporal, and perceptual aspects. According to its proponents, the taxonomy is used to characterize methods to ensure coverage of essential aspects of engineering complex systems, to provide a focusing framework to develop and select systems engineering innovation strategies and an organizing structure for classifying methodological research projects. JOM considers the five aspects as possible facets and interpretive lenses, used mostly as conceptual reference and for model evaluation.

8.1. JOM (V1)

This first (alpha) version of the ***Joint Optimization Metamodel*** (**JOM**) presented in this paper adopts loosely the Content Metamodel of the Open Group as a blueprint, however while the Content Metamodel follows the steps of the TOGAF Architecture Development Method (ADM) process, JOM adheres to an archetypal development process commonly known in systems engineering as "lifecycle" approach. This version of JOM is envisaged to have two main artifacts, one "static" consisting of a representation of a set of core entities and a dynamic process consisting of a set of steps devised to capture the joint optimization as a process, and to ensure continuity of the sociotechnical principles throughout development. JOM represents a first step towards defining a formal structure for a process to address joint optimization from the very beginning of development. It is intended as a conceptual reference to support some level of consistency and to provide guidance for developers and for managers and various STS stakeholders. As JOM matures and develops iteratively throughout the course of its application in the field, it is expected to undergo several rounds of refinement and transformations to evolve it into a fully functional tool.

8.2. JOM Vision and Concepts

Information architectures are based on defining "building blocks" within an architecture, specifying the relationships between them. The key constructs within the JOM metamodel are: Basic metamodel architecture based on Core and extension structure, the Extended Boundary, a Core entity, task and process, and Extensions.

8.3. JOM Architecture

One of the first challenges is to set a comprehensive system boundary inclusive of people and environment, in addition to pure technical system boundaries [43]. The metamodel starts as a basic model with minimum feature set (Core) and is designed to support the future inclusion of optional sets (Extensions) which can be developed and adopted at any stage of development. Core and Extensions architectures are widely in use in various types of software artifacts, as well as to the Content Metamodel which serves as blueprint for this version of JOM. The JOM core architecture consists of:

- Core Entities and relations;
- Core STS Tasks;
- Core Processes.

8.4. Extended Boundary

Modeling large scale, complex systems is non trivial, in particular addressing boundary issues between the technical system and the environment: "Considerations of large-scale engineering systems often present a dilemma of where to draw the line between a system and its environment. How are social, political, economic, and institutional issues addressed? How can the techniques of engineering science be connected with a modern understanding of human decision making, organizational behavior, and institutional inertia?" [43]. To build joint optimization into a STS from inception, the system boundary must necessarily include people and environment, resulting in what can be defined as an extended boundary. It is not possible to achieve joint optimization if the socio and the technical aspects are modeled separately as different things, because all the modeling, development and engineering activities that follow are influenced and guided by the boundary. Setting an inclusive boundary results in the STS being considered as a whole, rather than emerging from the interaction of different sub-systems. The three aspects of the system contribute different kinds of *resources* to the resulting STS, which are complementary and extend each other with corresponding attributes ad relations:

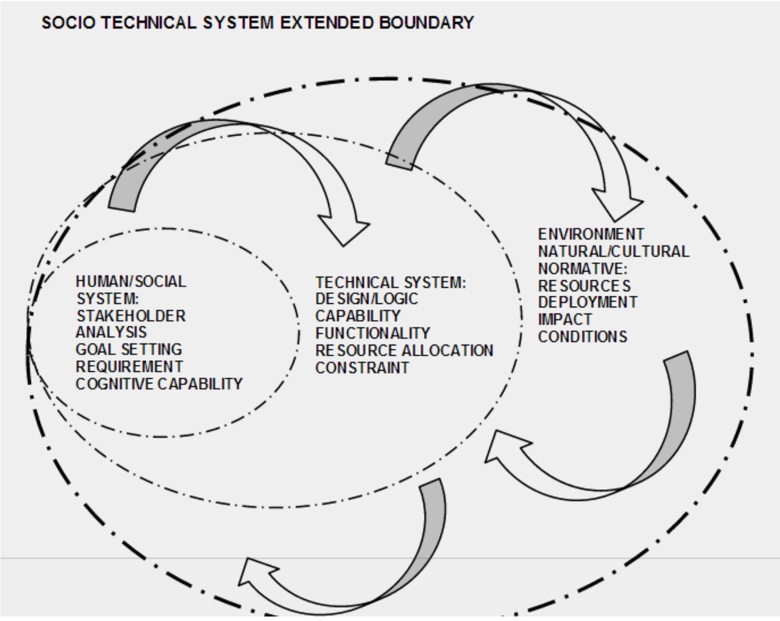
The **Human/Social** aspect of the system contributes (1) Stakeholders, (2) Goal/scope for the system/desired functionality, (3) Requirements, (4) Human level intelligence/cognition, (5) Specialized Knowledge and skills;

The **Technical** aspect is derived from some of the "attributes" of the Human/Social aspect, in the sense that it is generated from requirements and limited by resource constraints, however technology also recursively extends and influence the Human/Social aspect, for example with extended cognition and productivity capabilities, in what can be defined as a self determining hermeneutic cycle [44];

The **Environmental** aspect can be distinguished into Physical and Cultural/Social, contributing respectively natural resources (such as materials, physical laws, deployment grounds), as well as norms, languages, and intellectual context. The environment is where the impact of the STS can be observed and measured, it is also where physical and cultural conditions in turn influence the other two aspects.

Figure 1 represents a STS as a nested holarchy [45,46], where the core element is the human system, which influences (top arrow) and is influenced by (bottom arrow) the technical system, which in turn impacts (top arrow) as is constrained by (bottom arrow) the natural system/environment. The arrows represent both the correlation and interdependence of the three aspects, as well as the recursive *resonance* [1] in which each aspect influences on the others. In future work, resources permitting, we plan to study in more detail this interdependence by modeling and observing changes in behaviors between the three. A concept known as Total System Boundary [47] found in literature, is analogous to our STS Extended Boundary illustrated in Figure 1, however it was not applied to joint optimization.

Figure 1. The STS Extended Boundary.


SOCIO TECHNICAL SYSTEM EXTENDED BOUNDARY
HUMAN/SOCIAL SYSTEM: STAKEHOLDER ANALYSIS GOAL SETTING REQUIREMENT COGNITIVE CAPABILITY
TECHNICAL SYSTEM: DESIGN/LOGIC CAPABILITY FUNCTIONALITY RESOURCE ALLOCATION CONSTRAINT
ENVIRONMENT NATURAL/CULTURAL NORMATIVE: RESOURCES DEPLOYMENT IMPACT CONDITIONS

8.5. Core Entities

This is intended as a static part of the metamodel, the purpose of which is mostly to serve as a "catalog" of core entities and relations. No behavior is ascribed. They are derived from a systematic review and meta analysis of models of joint optimization in use, some of which have been outlined in Section 8 of this paper. The Core entities are intended to be a super-abstraction (a high level abstraction) and a meta synthesis of archetypes which can be observed at some level in "any" STS—at least to some degree—however these artifacts are designed to be "extensible" and become more detailed to any desired degree of granularity and to be revised iteratively against the outcomes of ongoing evaluation exercises. Table 1 lists the initial set of core entities identified in literature and from observations in the field, as the minimum essential set without which a STS does not exist. The other columns correspond to the three core dimensions of a STS, showing how each core entity plays a role in one or more dimensions. Feedback Loop is introduced as a core element, for completeness of design. Feedback is also a part of the Core Process Input-Output, discussed below. The next section provides some justification and an explanation of for these entities and the corresponding dimensional associations are derived.

Table 1. Schema of Joint Optimization Metamodel (JOM) core entities, and how these relate to the three main aspects of Socio Technical Systems (STS).

Core Entity	Social/Human	Technical	Environment
Stakeholder	X	X	X
Goal/Requirements	X	X	X
Culture	X	X	X
Task (Function/Process)	X	X	X
Resource/Input	X	X	X
Resource/Output	X	X	X
Feedback	X	X	X

Elements of the work system are abstracted from literature [39] in Tables 2 and 3, and rehashed to provide a higher level of abstraction and reduce redundancy.

Table 2. Elements of a work system [39].

Customers are recipients of a work system's products and services for purposes other than performing work activities within the work system. For example, a doctor who receives a lab result while treating a patient might be viewed as a customer of the lab, but is not a customer of the work system of providing medical care. Internal customers are customers of a work system who happen to be employees or contractors of the firm within which the work system operates. External customers are not employees or contractors for that firm.
Products and services are the physical things, information, and actions that are received or used by the work system's customers and that provide direct benefit for its customers. Different groups of customers may benefit from different products and services produced by the work system.
Processes and activities are actions that occur within the work system to produce products and services for its customers. Processes and activities include much more than totally structured processes that appear in some IS definitions as "procedures". As happens in collaboration systems and many other situations, various processes and activities performed by participants may be structured, semi-structured, or unstructured.
Participants are people who do the work within the work system. They include both IT users and non-users. Thus, non-users of IT may still be important to include as participants in the work system. A work systems's customers are also participants in many work systems that provide services. In self-service work systems, customers may be the only participants.
Information includes codified and non-codified information created and/or used by a work system.
Technologies include tools that are used by work system participants, plus other tools that perform totally automated activities. Technologies within a work system include IT and other, non-IT technologies that perform physical movements or transformations that are visible to users. Many technologies that perform physical activities (e.g., automobiles and microwave ovens) include embedded IT.
Infrastructure includes relevant human, information, and technical resources that are used by the work system but are managed outside of it and are shared with other work systems.
Environment includes the relevant organizational, cultural, competitive, technical, and regulatory environment within which the work system operates. A work system's environment should be considered when analyzing a work system because its success depends partly on surrounding factors that are not part of the work system itself.
Strategies that are relevant to a work system include strategies of the work system, the organization, and the enterprise. Strategies may or may not be articulated. When articulated, strategies may help in understanding a work system and its environment.

Table 3. Elements of a work system streamlined and regrouped as JOM elements.

Customers, Participants = Stakeholder
Products Processes and activities = Process
Strategy = Culture
Information =Resource
Infrastructure = Resource

An outline of JOM elements follows:

Stakeholder. Based on a Stakeholder Taxonomy [48] from literature, a segmentation of different roles of system stakeholders is identified which includes Beneficiary/User, Negative Stakeholder, Developer, Operator, Purchaser, Sponsor/Investor, Consultant, Regulator. It should, however, be noted that (a) in real world STS settings the roles are increasingly interchangeable (b) a stakeholder entity can have multiple roles, sometimes simultaneously. While Stakeholder primarily pertains to the first dimension (human/social) as stakeholders are inherently people/human, JOM purports that both the other two dimensions are directly related the human dimension. For example (a) stakeholders use/need/create/co-create technologies, and there is interdependence between users cognitive and behavioral traits technological systems they adopt and *vice versa* [49,50], as well as by their physical and cultural environments surrounding them There is a growing consensus that timely and broad based stakeholder involvement is a vital ingredient for effective environmental assessment, as it is for project planning, appraisal and development in general. The World Bank has found that public participation in EIA (environmental impact assessment) tends to improve project design, environmental soundness and social acceptability [51] and (b) the environment is also a type of external stakeholder in any given STS [52].

Goal/Function. What purpose should the system fulfill? What problem does it solve? The crux of what a STS consists of, is initially determined by standard functional requirements. There is a strong correlation between "goal" and "stakeholder" identification, in that different groups of stakeholders may have different goals as the priority for the system. How to reconcile different views and stakeholders expectations is part of a balancing exercise which continues throughout the life of a STS, and relies on multistakeholder modeling techniques such as IEEE1471 to capture different views and viewpoints.

Culture. Under Culture everything that defines to the socio technical praxis. It includes organizational culture, axioms, values, beliefs, attitudes, worldviews, paradigms, policies, rules, laws, procedures. Culture is inherently a social entity, albeit largely intangible, but clearly reflected in the governance structure and policies not only of the organization (as a cluster of stakeholders) but also of the STS itself. Technology influences culture, and culture influences technology, culture also impacts the environment [53,54].

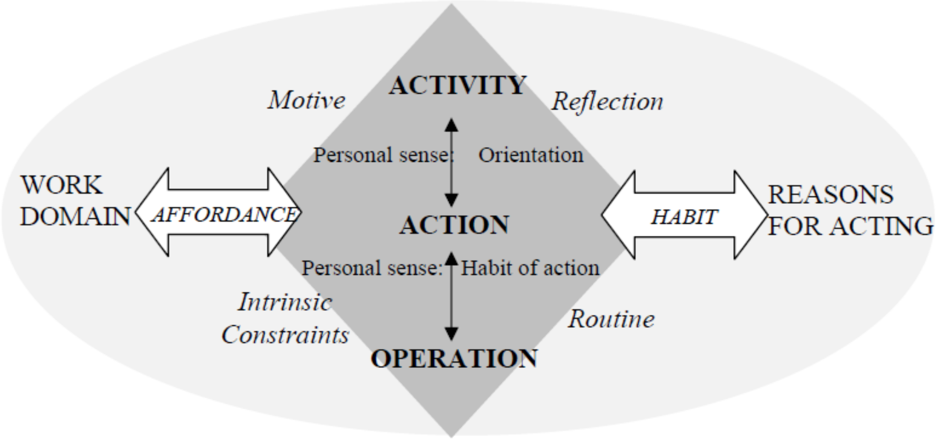
Task (Function/Process). A task is any activity which supports the system goal and objectives, via "functions", which are implemented through processes. Typically task models distinguish between system task model that provide information about how the designed system requires tasks to be performed, and user task model which is how users perform system tasks, however, in STS terms, a socio technical task model devised to support joint optimization, must inevitably be the synthesis of both.

Defining tasks in complex systems is problematic as illustrated by text book example: "if you go down to a shipyard and look around for as shipbuilder you will find welders, carpenters, foremen, engineers, and many other specialist, but no shipbuilders. True, the company executives may call themselves shipbuilders, but if you observe them at their work… consists of writing contracts, planning budgets, and other administrative activities… they are not in any concrete sense building ships… a system is building ships, and the system is the shipbuilder" [55].

A concept defined as "core task" for modeling the "outcome critical content" of process control work in complex, dynamic and technologically mediated environments (such as air traffic control and nuclear power plant control room), shown in Figure 2, is defined in the Core Task Analysis (CTA) Framework [56] and used for in analyzing working practices and personal work orientations:

"The result-critical content of a particular work, which defines both the possibilities for action and the demands that must be fulfilled in all situations in order to maintain appropriate interaction with the environment". Nevertheless, the core task concept should not be restricted to individual actions, but action should be interpreted in a wider perspective of a societal activity that is carried out co-cooperatively by a number of actors. Therefore, leveraging the same fundamental assumption, the element TASK in JOM is represented with the set operation, action, activity, motivation.

Figure 2. Core Task Model, Norros [56].



Tasks can be of different kinds, for example Cognitive Manual/Physical Digital/Virtual, but in contemporary information technology intensive scenarios, it is likely to be a hybrid, which is becoming increasingly intertwined: cognitive systems automate at least to some extent intelligent tasks carried out by humans, and also increasingly present in tools and processes which were traditionally manual and "physical". Operational tasks, from complex critical systems to everyday chores, are increasingly technology intensive, and as such implemented via processes, whether routinely automated or *ad hoc*. The "environment" is both the space where all tasks and actions take place, and where their impact/effect becomes observable and measurable. Therefore, the element TASK exists in all three dimensions of STS models.

Resource (input/output). A STS resource is whatever the system requires to operate and fulfill its function/goal, as well as the product/outcome of the any system function. It's a very high level class to represent everything that is required and everything that is produced, No single standard model exists of "System resources", let alone a model for socio technical systems resources, which is going to be tackled in more detail in future work. For the purpose of this paper JOM characterizes resources: tangible/physical (as in materials), cognitive (knowledge, information), and non-tangible (organizational, belief systems, socio-political infrastructures). STS resources are typically a combination of human, technical and environmental assets, and as such this element exists also in all three dimensions.

Feedback. Feedback is inherently the life and soul of STS, the single mechanism thanks to which the system and its various components exchange information to enable goals to be achieved, functions to be performed correctly and eventually, for all systems states to be reached and maintained, thus enabling joint optimization. The structure of feedback is recursive: feedback loops gather responses to outputs and return value-actions and need to be iterative, continuous. Some design tradeoffs have been considered, whether feedback should be one of the core elements, or part of the core process. Given the importance of feedback in achieving JO, JOM enlists feedback as both.

8.6. Core Dynamic Relations

Mapping socio technical systems relations, using for example the ER (entity relationship) formalism, although theoretically feasible, is not within the scope of this metamodel. Given the unpredictable nature of STS states, scenarios and responses, JOM does not capture nor attempts to characterize predefined relations, which are left deliberately undefined because these are potentially infinite in number and behaviors, the identification of which exceeds the scope of a metamodel which represents reality at a high level of abstraction. Ultimately, everything is related to everything else, and science is only beginning to discover the subtle dependencies and far sighted correspondences to and causal dependencies that organize reality in the universe. Given the need to set constrains to support functional requirements, JOM adopts an existential construct for dynamic relations, which determines whether there exist a relation between system elements and components, or not, and whether the relation is unidirectional or bidirectional. Relations in JOM come into existence "at runtime", whenever:

- There is a dependency between an entity and any other entity or attribute or part of the system;
- There is a communication flow (exchange of data, information or other resource) between entities and any other dimension or part of the system.

Relations are dynamic, in that over time they can change in nature, intensity frequency, meaning in reaction to ongoing adjustments of the system to reach and maintain optimal equilibrium to fulfill its functional capability, irrespective of shifts in conditions and states of other systems elements. In addition JOM can capture some degree of cardinality if and when this can be realistically determined using standard notation of the modeler choice.

8.7. Core STS Process

A process essentially consists of what goes on in the system defining its dynamics and efficiency. That's where everything that really matters happens, the rest of the architecture consists of mere "ingredients" for a successful socio technical recipe. Differences in process execution impact the outcome and outputs. A system process can be defined as "a business process" in the sense that it is any procedure relevant for creating value [57]. A common example in systems engineering is the

Business Process Model. What is unique to a STS process, and that has not yet been identified in standard process models in use, is the process which enables and supports (or disables and inhibits) Joint Optimization (JO). Joint optimization is the result of a "business process" carried out through iterative recursions across the three STS dimensions, People Technology and Environment. Each task, function, process, and activity pertaining to the system must be carried out in relation to all of its elements and across its core dimensions.

To facilitate Joint Optimization, JOM adopts the central elements of the process architecture of ARIS (Architecture of Integrated Information Systems) Architecture and Reference Models for Business Process Management [57], which distinguishes between four interlinked levels of processes:

Process engineering—processes are modeled in accordance with a work schedule. Various methods for optimizing, evaluating and ensuring the quality of the processes are adopted;

Process planning and control—processes are planned and controlled, with methods for scheduling and capacity. Process monitoring is the control mechanism;

Work flow control—data information and knowledge objects are scheduled according to work design criteria, and implemented via work flow processes;

Process execution—the function of enacting the business process.

The specific characteristic of ARIS inherited by JOM (Figure 3) is that "the four levels are interdependently connected: Information at Level II regarding the profitability of current processes, is the point of departure for continuous adjustment and improvement of the business processes at Level I. Work flow Control is linked to Level I, because Work flow Control at Level III requires the description of business processes. At the same time, Work flow Control reports actual data regarding the processes to be executed (amounts, times, organizational allocation) back to Level II. Applications at Level IV are executed from the work flow system at Level III and configured according to the business process models at Level I" [57]. Another characteristic of the JOM process is that it is inherently iterative, and self determining, supporting "coevolution" (Figure 4) whereby Interactions and feedback loops are represented by arrows. Interactions are iteratively defined, as in the Rational Unified Process (RUP) [58].

JOM core process has a dual function:

(1) Support the development of a STS, by ensuring that (a) comprehensive stakeholder perspective is adhered to; and (b) each phase of the system development process is carried out iteratively taking into the account as input the output of the precedent phase;

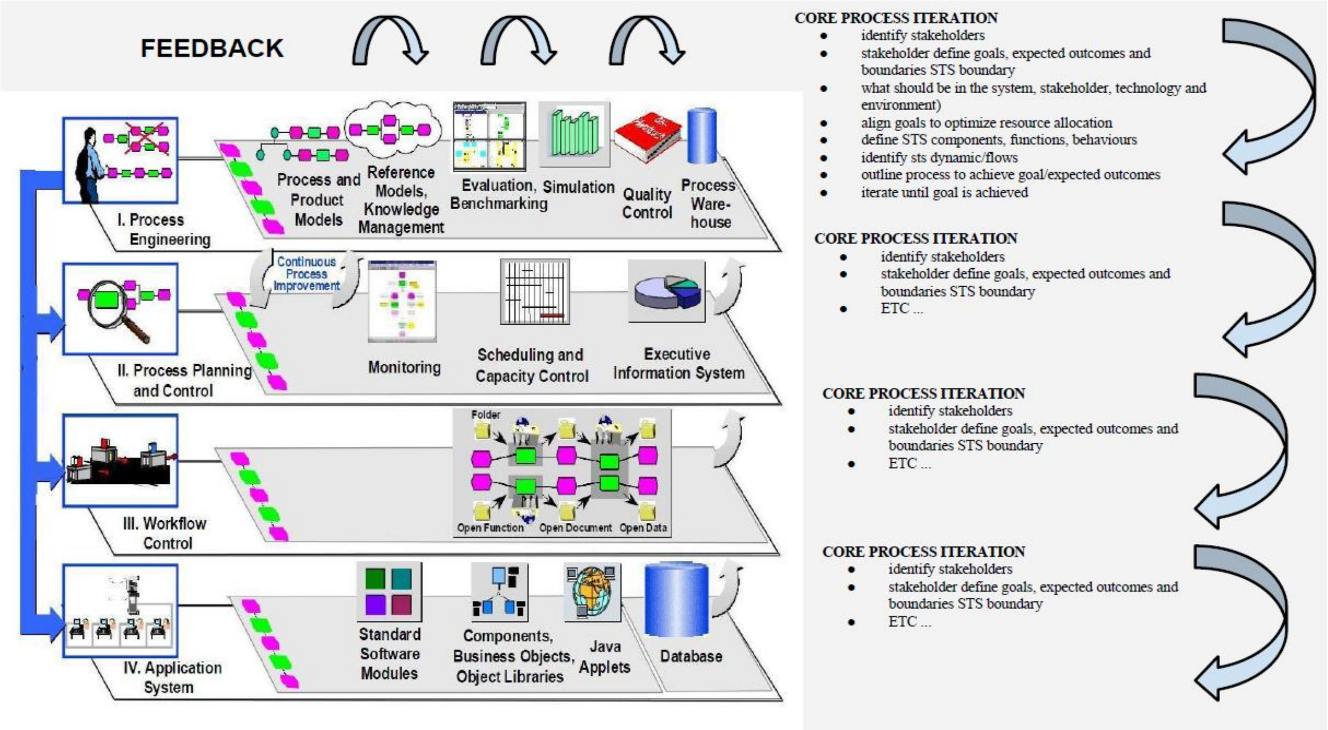
(2) To ensure that the STS which is the outcome of JOM core process, adheres to its blueprint, IE, that replicates its core structure and features;

Therefore the JOM core process, essentially consists of a series of steps to e carried out alongside the four dimensions, which maximize the inclusion of STS stakeholders and is capable of adjusting

- Identify stakeholders;
- Stakeholder define goals, expected outcomes and boundaries STS boundary;
- (What should be in the system, stakeholder, technology and environment);
- Align goals to optimize resource allocation;
- Define STS components, functions, behaviors;

- Identify STS dynamics/flows;
- Outline process to achieve goal/expected outcomes;
- Iterate until goal is achieved.

Figure 3. JOM Process Architecture Modified from ARIS (Architecture of Integrated Information Systems) [57].


FEEDBACK
I. Process Engineering
Process and Product Models
Reference Models, Knowledge Management
Evaluation, Benchmarking
Simulation
Quality Control
Process Ware-house
II. Process Planning and Control
Continuous Process Improvement
Monitoring
Scheduling and Capacity Control
Executive Information System
III. Workflow Control
Folder
Open Function
Open Document
Open Data
IV. Application System
Standard Software Modules
Components, Business Objects, Object Libraries
Java Applets
Database

CORE PROCESS ITERATION

- identify stakeholders
- stakeholder define goals, expected outcomes and boundaries STS boundary
- what should be in the system, stakeholder, technology and environment)
- align goals to optimize resource allocation
- define STS components, functions, behaviours
- identify sts dynamic/flows
- outline process to achieve goal/expected outcomes
- iterate until goal is achieved

CORE PROCESS ITERATION

- identify stakeholders
- stakeholder define goals, expected outcomes and boundaries STS boundary
- ETC ...

CORE PROCESS ITERATION

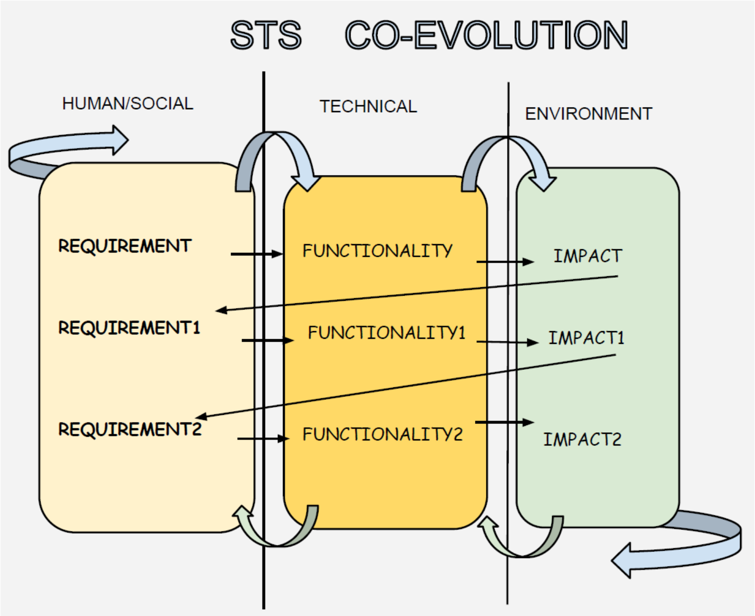
- identify stakeholders
- stakeholder define goals, expected outcomes and boundaries STS boundary
- ETC ...

CORE PROCESS ITERATION

- identify stakeholders
- stakeholder define goals, expected outcomes and boundaries STS boundary
- ETC ...

Note: curved arrows indicate iteration and feedback flows.

Figure 4. STS co-evolution.

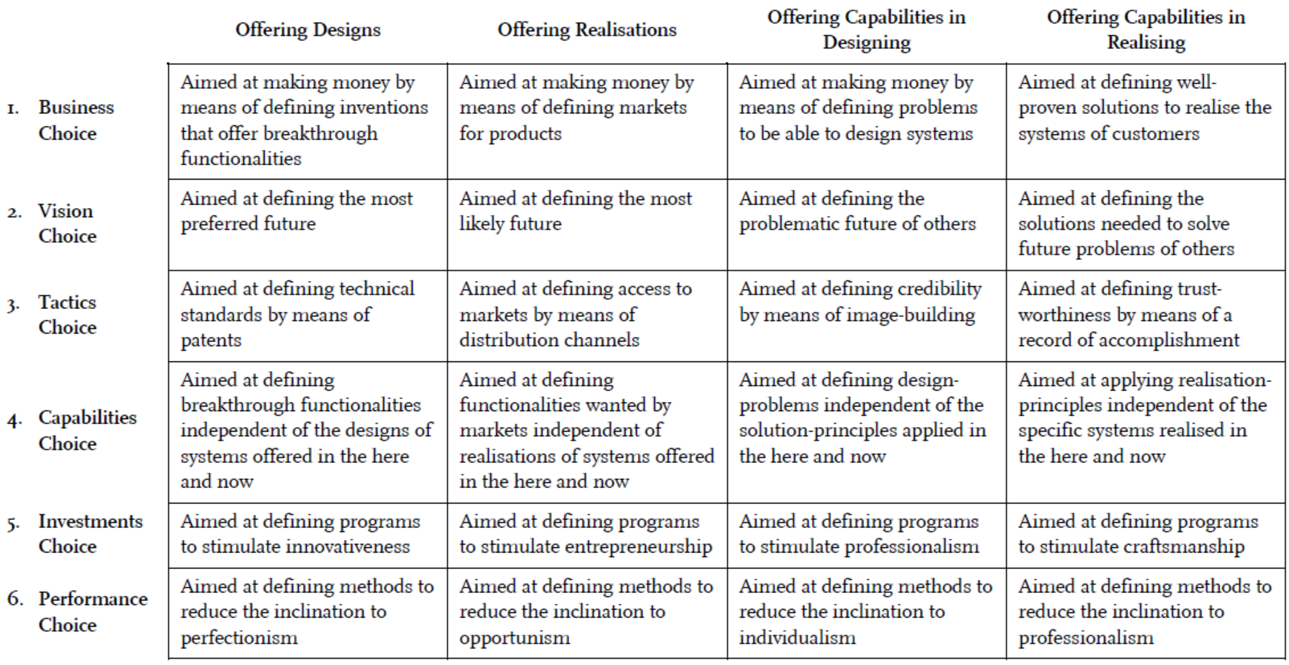


The resulting JOM core process outline therefore consist of these steps carried out iteratively across the four level process architecture defined earlier, as shown diagrammatically in Figure 3.

8.8. Extensions

Extensions are additional model component which amplify and expand the capability of JOM, and which exceed the scope of the core elements of the metamodel. It is difficult at this stage to anticipate fully what extensions will be developed with usage, because these will depend on the desired function they need to support, potentially leading to the modeling of multiple incommensurable perspectives which cannot be predicted. Extensions presented here so far are envisaged to be shaped following a strategic management model (Figure 5) that enables the interactions to evolve in such a way capable to capture the strategic reality they intend to explore and create via the system. Adopting a model which identifies a series of archetypal choices namely Business, Vision, Tactics, Capabilities, Investment, Performance "In designing or defining the content of strategies, by means the scientific design cycle, each design state of this cycle relates to a strategic concept" [59].

Figure 5. Configuration of strategic realities matrix [59].



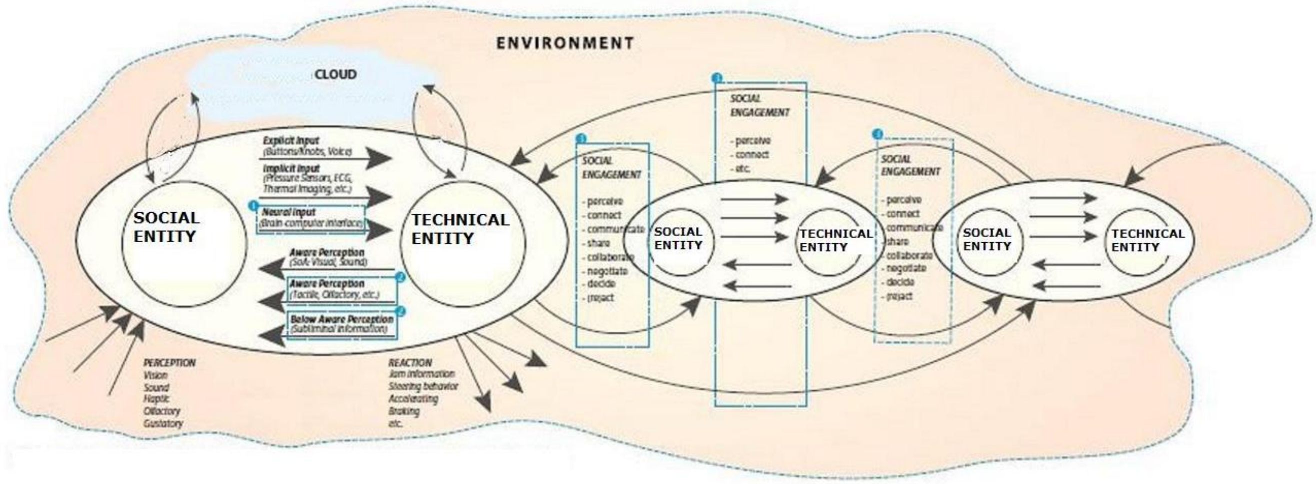
	Offering Designs	Offering Realisations	Offering Capabilities in Designing	Offering Capabilities in Realising
1. Business Choice	Aimed at making money by means of defining inventions that offer breakthrough functionalities	Aimed at making money by means of defining markets for products	Aimed at making money by means of defining problems to be able to design systems	Aimed at defining well-proven solutions to realise the systems of customers
2. Vision Choice	Aimed at defining the most preferred future	Aimed at defining the most likely future	Aimed at defining the problematic future of others	Aimed at defining the solutions needed to solve future problems of others
3. Tactics Choice	Aimed at defining technical standards by means of patents	Aimed at defining access to markets by means of distribution channels	Aimed at defining credibility by means of image-building	Aimed at defining trust-worthiness by means of a record of accomplishment
4. Capabilities Choice	Aimed at defining breakthrough functionalities independent of the designs of systems offered in the here and now	Aimed at defining functionalities wanted by markets independent of realisations of systems offered in the here and now	Aimed at defining design-problems independent of the solution-principles applied in the here and now	Aimed at applying realisation-principles independent of the specific systems realised in the here and now
5. Investments Choice	Aimed at defining programs to stimulate innovativeness	Aimed at defining programs to stimulate entrepreneurship	Aimed at defining programs to stimulate professionalism	Aimed at defining programs to stimulate craftsmanship
6. Performance Choice	Aimed at defining methods to reduce the inclination to perfectionism	Aimed at defining methods to reduce the inclination to opportunism	Aimed at defining methods to reduce the inclination to individualism	Aimed at defining methods to reduce the inclination to professionalism

The extension groupings outlines so far are intended to remain indicative ad purely providing a conceptual space for further tailoring which has to be carried out by stakeholders and developers on an "*ad hoc*" basis to target the uniqueness of each system JOM is going to be applied to. The core and extension blueprint is supports various aspects of agile/iterative development such as the method plug-in concept found within the Software Process Engineering Metamodel (SPEM) developed by the Object Management Group (OMG).

9. Summary Overview of JOM

Addressing socio technical challenges is a new imperative across all areas of organizational development, but it is becoming increasing important also for all technological systems developers, from software to cyber-physical infrastructures which societies are becoming increasingly reliant upon. JOM is a conceptual artifact, providing a set of references to help define socio technical optimization to guide design and implementations. It emphasizes iterations, feedback and essentially puts forward the notion that joint optimization can only occur if it takes place across the three essential dimensions of STS. JOM is mainly a reference artifact to guide (a) developers to create intelligent evolutionary systems "aware" of the complex multidimensionality of reality, which exceed the scope of technical boundary traditionally practiced in engineering and (b) managers and other stakeholders, to design their processes and work flows to consider the extended boundary, and the co-evolutionary dynamics across the many socio technical dimensions of a system. Figure 6 illustrates an example of a jointly optimized STS modeled using the principles underlying JOM, where an extended boundary, elements, processes are evident, and the emphasis on iteration, information exchange and feedback, is clearly shown by the arrows.

Figure 6. Outline of Jointly Optimized STS system (modified from [60]).



10. Evaluation

There is no standard universally accepted method to evaluate neither metamodels nor STS Joint Optimization. Evaluation questions are useful in this respect, such as:

- Does JOM help joint optimization?
- Does JOM describe joint optimization of STS?
- How can JO be evaluated?
- What are the tangible indicators of JO?

The first version of JOM has been developed iteratively leveraging the experience and heuristics gathered during professional and research activities spanning several years, which also provided the motivation for this research, therefore itself is the result of ongoing evaluations. So far JOM has been used in graduate and postgraduate courses in STS modeling (2013) informally by peers in workshops, and it has been applied through consulting practice in a series of real world cases and scenarios to address and resolve different categories of socio technical problems, from lack of employee satisfaction to lack of adoption of information technology resources, to sloppy (suboptimal) collaboration and team performance. The initial evaluation received so far via feedback from the experimental user community can be summarized as:

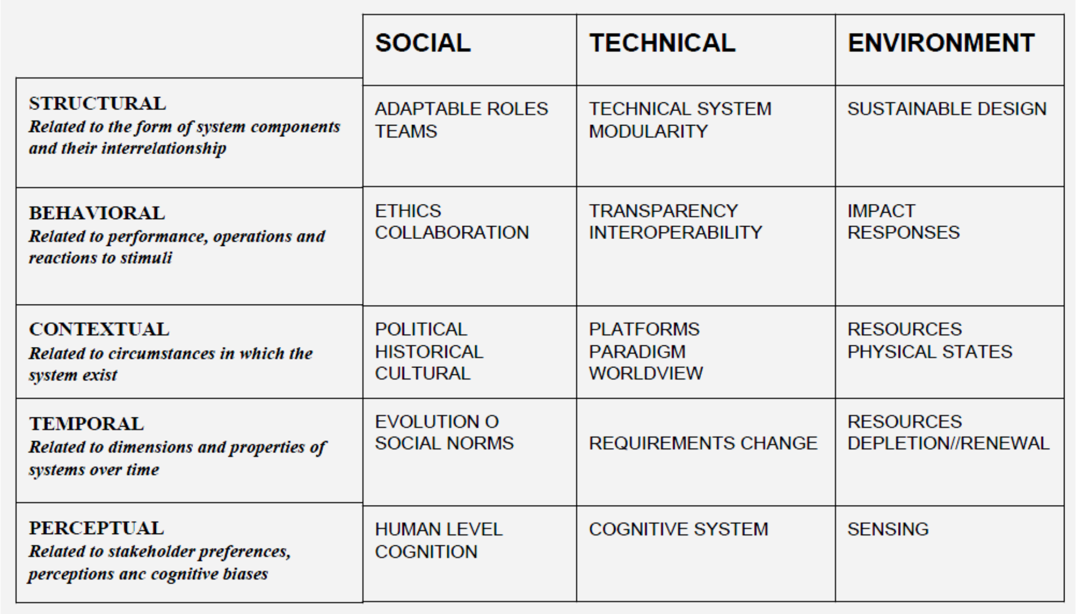
JOM is useful to learn about Joint Optimization, otherwise very little known concept, because it is not taught in engineering curricula, not taught in management;

JOM is useful to some extent to guide the basic aspects of socio technical modeling and joint optimization, especially at early stage of system conceptualization.

It is however not yet sufficiently formalized to be usable as a standalone modeling tool, but needs to be used in conjunction with other tools such as UML and various object oriented and ontological approaches.

The publication of a catalog of case studies and evaluation scenarios will be considered in future work. A conceptual evaluation has been carried out by cross referencing JOM constructs with existing frameworks such as the five aspects taxonomy by Rhodes [42], showing that each aspect of the taxonomy finds correspondence in each STS dimension, namely social, technology, and the environment (Figure 7).

Figure 7. Five Level Taxonomy [42] used to populate a STS matrix.



	SOCIAL	TECHNICAL	ENVIRONMENT
STRUCTURAL ***Related to the form of system components and their interrelationship***	ADAPTABLE ROLES TEAMS	TECHNICAL SYSTEM MODULARITY	SUSTAINABLE DESIGN
BEHAVIORAL ***Related to performance, operations and reactions to stimuli***	ETHICS COLLABORATION	TRANSPARENCY INTEROPERABILITY	IMPACT RESPONSES
CONTEXTUAL ***Related to circumstances in which the system exist***	POLITICAL HISTORICAL CULTURAL	PLATFORMS PARADIGM WORLDVIEW	RESOURCES PHYSICAL STATES
TEMPORAL ***Related to dimensions and properties of systems over time***	EVOLUTION O SOCIAL NORMS	REQUIREMENTS CHANGE	RESOURCES DEPLETION//RENEWAL
PERCEPTUAL ***Related to stakeholder preferences, perceptions anc cognitive biases***	HUMAN LEVEL COGNITION	COGNITIVE SYSTEM	SENSING

The evaluation so far has resulted in the identification of five different types of Joint Optimization, each corresponding to one aspect of the taxonomy and briefly characterized as:

Structural JO—Takes place when the structure of the sociotechnical system is determined by implementing STS design principles across all three dimensions;

Behavioral JO—Takes place when the system behavior is synchronized across all three dimensions;

Contextual JO—Context is optimized when it is captured and represented systematically across all three dimensions;

Temporal JO—When systemic change and evolution are synchronized across three dimensions;

Perceptual JO—When cognitive structures, knowledge representation and information flows are harmonized and consistent across the three dimensions.

11. How to Use JOM

When designing systems, "business" analysis is performed to understand the system goals and scope, as well as to gather information and knowledge about all the factors that are relevant to the system and its development and operation. When designing socio technical systems, taking into account the extended boundary, the number of variables and scenarios increases, and so do the uncertainties typically associated with large numbers of variables. No development process exists to support the design of socio technical systems capable of achieving joint optimization, and very limited examples of guiding principles to guide socio technical development in general. Therefore, in the first instance, JOM is to be used as a reference for all stakeholders, systems engineers and developers to keep in mind and learn before and during development what is joint optimization, why it is important and how it can be achieved. JOM artifacts are to be used to cross reference entities identification and process design in conjunction with traditional system modeling approaches—whether Object Modeling, or Entity Relationship techniques—to identify the ingredients (the elements) and the method (the process) of STS design and engineering so that JO can be targeted as a built in property (or function) of the system. Considering the trend toward automation—automation of systems design, coding, operation and to some extent system management—JOM can also be used as an explicit knowledge representation artifact, which can be encoded and programmed as a learning algorithm, to facilitate computational modeling and simulation of STS. The author hopes that with usage and refinement JOM can serve as the basis for supporting the automation of socio technical design and development and management, and possibly evolve into a metamodeling language of its own.

12. JEO, and the Joint Optimization of Ontology Development

As an example showing how JOM can be used to support the joint optimization in ontology development, this paragraph provides an outline of *Just Enough Ontology* (JEO) [61] an agile approach to ontology development, where bot the ontology and the development process are considered from a socio technical viewpoint. The majority of information systems today are designed to leverage knowledge expressed via natural language, where symbols and meanings (semiotics and semantics) need to be captured and represented adequately for these systems to function. Ontologies are conceptual and semantic representations widely used, in different forms, to capture and express models of reality. Digital systems are inherently "information centric", and ontologies have become essential shared and explicit knowledge representation mechanisms for open information systems. Until recently, web based ontologies have largely been considered formal, strictly "technical" artifacts, with the main purpose to enable artificial intelligence and automated agents to carry out some reasoning. Their development and deployment has been typically driven by logicians and mathematicians, and heavily constrained by the implementation language of choice. Under a pragmatic worldview, it is noted that technically and logically correct artifact not always work smoothly in practice, for a number of reasons: web based ontologies have turned out to be expensive to develop and maintain, and because their development is often driven by upfront implementation choices (encoding language), they end up not being reusable and become owned, *de facto*, by those who have the skillset to encode and maintain the artifacts in a particular language, and in a particular environment. From a sociotechnical point of view, it is important to recognize that are ontologies are "socio technical systems", in at least two ways:

(1) The development of ontologies—also known as ontology engineering—inevitably relies on a diverse set of humans who make decisions in relation to goals, scope, methods, requirements, desired functionalities, *etc.*, but also to its management (costing, resourcing);

(2) The deployment of ontologies, in the majority of cases, impacts humans because it determines the functionalities and behaviors of information systems that people increasingly depend on for their daily lives.

To overcome the shortfalls of traditional formal ontology engineering method, JEO was devised, as the first socio technical approach to ontology development which incorporates stakeholder identification as the first step, and where the development of the system is not driven by the upfront choice of implementation (encoding happens toward the end of the process). Most importantly, by conjoining stakeholder involvement at every phase of the lifecycle, and supporting "autonomy" of various stakeholder teams, as well as the principles of flexibility, meaningfulness of tasks and iterations/feedback, both the development process and the artefacts that derive as final outcomes of such process can be considered "jointly optimized". JEO considers an ontology as a system, defined as "a collection of components organized to accomplish a specific function or set of functions". The term system encompasses individual applications, systems in the traditional sense, subsystems, systems of systems, product lines, product families, whole enterprises, and other aggregations of interest. A system exists to fulfill one or more missions in its environment. (IEEE 1471). The rationale process and outcomes of JEO are presented in detail in corresponding publications [62–64]. Below an outline of the core JEO process, modeled on JOM.

Just Enough Ontology Process (modeled on JOM):

(1) Identify stakeholders, outline stakeholder profile;

(2) Define the purpose of the ontology (emphasis on representation/indexing, problem solving/ reasoning);

(3) Outline requirements;

(4) Identify and survey existing knowledge sources and existing ontologies, elicit existing knowledge Assess why the existing knowledge resources do not meet the intended user requirements, update the requirements with the output of activities above (iterative);

(5) Scoping ontology (defining the boundaries and level of granularity, according to goal and stakeholders requirements) Update the requirements (iterative);

(6) Devise and implement quality assurance plan Add quality parameters to the requirements (iterative);

(7) Define the field of competence to identify the knowledge boundaries (competence questions) Match the field of competence with the knowledge sources;

(8) Define the ontology artifacts: Vocabulary Identify concepts/entities/classes relations, axioms Refine and map vocabularies to artifacts;

(9) Transfer concepts to ontology language representation: Select knowledge representation formalisms and annotation depending on stakeholder requirements, scope and goal;

(10) Deploy/systems integration (modular, incremental);

(11) Testing Evaluation quality monitoring competence assessment;

(12) Publishing;

(13) Maintenance/Reuse.

JEO adopts all of JOM elements (except Culture):

Stakeholder—where the widest possible stakeholder base in identified and included in the process;
Goal/Requirements—where each set of stakeholders contribute to define;
Task (Function/Process)—various tasks are identified;
Resource/Input—stakeholders inputs and knowledge sources;
Resource/Output—artifacts;
Feedback—iterations of all phases.

13. Limitations

This paper summaries efforts in taking steps towards making explicit, by means of a metamodeling approach, an emergent construct known as socio technical systems joint optimization (STSJO) which is considered fundamental in STS theory and practice, although still little understood. Given the high level of abstraction of the proposed conceptual representation of Joint Optimization as a metamodel, JOM is intended as a general working tool which may need to evolve further to fit disparate instantiation challenges such as different types of systems in different domains. Its main limitations, like most metamodels, it is that is very generic. JOM is not a metamodeling language.

14. Conclusions and Future Work

This paper presents the justification and the rationale for the development of artifacts to support the Joint Optimization of Socio Technical Systems. A brief analysis of related approaches found in literature is presented, together with an outline of JOM, a metamodel that furthers the state of the art by building on existing work and addressing issues not resolved in existing approaches. Some level of evaluation is discussed, and evaluation findings presented, including the identification of five types of Joint Optimization. In future work we expect to apply JOM extensively to a variety of use cases, and to manipulate JOM iteratively until an optimal artifact is achieved, that we may consider putting forward as a candidate for standardization with relevant bodies. In future work, variables abstracted from construct known as Levels of Joint Optimization [17] will be incorporated in JOM and a catalog of evaluation scenarios and case studies to be compiled in support of future iterations for the development of the metamodel.

Conflicts of Interest

The author declares no conflict of interest.

References

1. Hollnagel, E. Understanding Accidents—From Root Causes to Performance Variability. In Proceedings of the 7th IEEE Conference on Human Factors and Power Plants, Scottsdale, AZ, USA, 15–19 September 2002.
2. Lock, S. The management of Socio Technical Systems using configuration modeling. *Hum. Syst. Manag.* **2004**, *23*, 29–47.

3. Fran çois, C. Systemics and cybernetics in a historical perspective. *Syst. Res. Behav. Sci.* **1999**, *16*, 203–219.
4. Von Bertalanffy, L. An outline of general system theory. *Br. J. Philos. Sci.* **1950**, *1*, 134–165.
5. Stanton, N.A. *Human Factors Methods: A Practical Guide for Engineering and Design*; Ashgate Publishing: Farnham, UK, 2012.
6. Baxter, G.; Sommerville, I. Socio Technical Systems: From design methods to systems engineering. *Interact. Comput.* **2011**, *23*, 4–17.
7. Qureshi, Z.H. A Review of Accident Modelling Approaches for Complex Critical Sociotechnical Systems. Available online: http://crpit.com/confpapers/CRPITV86Qureshi.pdf (accessed on 30 January 2014).
8. Heylighen, F. The science of self-organization and adaptivity. *Encycl. Life Support Syst.* **2001**, *5*, 253–280.
9. Cohen, M.D.; Riolo, R.L.; Axelrod, R. The Emergence of Social Organization in the Prisoner's Dilemma: How Context-Preservation and Other Factors Promote Cooperation. Available online: http://www.santafe.edu/media/workingpapers/99-01-002.pdf (accessed on 30 January 2014).
10. Henry, C.L. *Why Information Systems Fail*; Columbia University Press: New York, NY, USA, 1975 (658.403/LUC).
11. IBM Research. Available online: http://www.research.ibm.com/cognitive-computing/index.shtml#fbid=ucEfSn05Oru (accessed on 30 January 2014).
12. Mitchell, V.L.; Nault, B.R. *The Emergence of Functional Knowledge in Sociotechnical Systems*; Haskayne School of Business, University of Calgary: Calgary, Canada, 2003.
13. Fischer, J.; Amekudzi, A. Quality of life, sustainable civil infrastructure and sustainable development. *J. Urban Plan. Dev.* **2011**, *137*, 39–48.
14. Sussman, J. Today's Sociotechnical Problems Require a New Field of Study. The Annual Charles L. Miller Lecture. 2012. Available online: http://esd.mit.edu/Headline/calendar/2012/042512millerlecture.html (accessed on 30 January 2014).
15. Priscoli, J.D. Retraining the modern civil engineer. *Environmentalist* **1983**, *3*, 137–146.
16. Vespignani, A. Modelling dynamical processes. *Complex Soc. Tech. Syst. Nat. Phys.* **2011**, doi:10.1038/NPHYS2160.
17. Grenville, N.D. A Sociotechnical Approach to Evaluating the Effects of Managerial Time Allotment on Department Performance. Master Thesis. Available online: http://scholar.lib.vt.edu/theses/available/etd-534692039701091/unrestricted/Etd.pdf (accessed on 30 January 2014).
18. Trist, E.L.; Bamforth, K.W. Some social and psychological consequences of the Longwall method. *Hum. Relat.* **1951**, *4*, 3–38.
19. Giese, H.; Brun, Y.J.; Serugendo, D.M.; Gacek, C.; Kienle, H.; M üller, H.; Pezz è, M.; Shaw, M. *Engineering Self-Adaptive and Self-Managing Systems*; Springer-Verlag: Berlinn-Heidelberg, Germany, 2009.
20. Trist, E. *The Evolution of Socio Technical Systems*; Ontario Ministry of Labour, Ontario Quality of Working Life Centre: Toronto, Canada 1981.
21. Jenkins, D.P.; Stanton, N.A.; Walker, G.H.; Salmo, P.M. *Command and Control: The Sociotechnical Perspective*; The Sociotechnical Human Factors in Defence; AshGate Publishing: Farnham, UK 2009.

22. Davis, L.E. Evolving alternative organization designs: Their sociotechnical bases. *Hum. Relat.* **1977**, *30*, 261–273.
23. Appelbaum, S.H. Socio Technical systems theory: An intervention strategy for organizational development. *Manag. Decis.* **1997**, *35*, 452–463.
24. Van de ven, A.H.; Joyce, W.F. Overview of Perspectives on Organization Design and Behavior; In *Perspectives on Organization Design and Behavior*; John Wiley and Sons: New York, NY, USA, 1981.
25. Pasmore, W.; Francis, F.; Haldeman, J.; Shani, A. Sociotechnical systems: A North American reflection on empirical studies of the seventies. *Hum. Relat.* **1982**, *35*, 1179–1204.
26. Hackman, J.R.; Oldham, G.R. Motivation through the design of work: Test of a theory. *Organ. Behav. Hum. Perform.* **1976**, *16*, 250–279.
27. Sitter, L.U.; Hertog, J.F.; Dankbaar, B. From complex organizations with simple jobs to simple organizations with complex jobs. *Hum. Relat.* **1997**, *50*, 497–536.
28. Carvalho, P.V.R.D. Ergonomic field studies in a nuclear power plant control room. *Prog. Nucl. Energy* **2006**, *48*, 51–69.
29. Cherns, A.B. Principles of sociotechnical design revisited. *Hum. Relat.* **1987**, *40*, 153–162.
30. Fishwick, P.A. Toward a Convergence of Systems and Software Engineering. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.3999&rep=rep1&type=pdf (accessed on 30 January 2014).
31. Doets, K. *Basic Model Theory*; CLSI Publications: Stanford, CA, USA, 1996.
32. Smiraglia, R.P. The progress of theory in knowledge organization. *Libr. Trends* **2002**, *50*, 330–349.
33. Cunningham Ward. Object *vs*. Model. 2010. Available online: http://c2.com/cgi/wiki?ObjectVsModel (accessed on January 2014).
34. Ernst, J. What is Metamodeling, and What is it Good for? Available online: http://infogrid.org/trac/wiki/Reference/WhatIsMetaModeling (accessed on 30 January 2014).
35. Benoit, C. A Tutorial about Metamodeling Using OMG Norms and Eclipse Modeling University of Rennes 1 (IRISA & ESIR). 2012. Available online: http://people.irisa.fr/Benoit.Combemale/teaching/mde/MetamodelingWithEclipseModelingTutorial-2x2.pdf (accessed on 30 January 2014).
36. Meta Object Facility (MOF). Available online: http://www.omg.org/mof/ (accessed on 30 January 2014).
37. Munkvold, B.E. Tracing the Roots: The Influence of Socio Technical Principles on Modern Organisational Change Practicesin Coakes. In *The New SocioTech*; Springer: London, UK, 2000.
38. Alter, S. Metamodel for Understanding, Analyzing, and Designing Sociotechnical Systems. In Proceedings of the JAIS Theory Development Workshop, Phoenix, AZ, USA, December 2009. Available online: http://sprouts.aisnet.org/9-59 (accessed on 30 January 2014).
39. Alter, S. The work system method for understanding information systems and information systems research. *Commun. Assoc. Inf. Syst.* **2002**, *9*, 90–104.
40. Hamilton, D.; Orr, K.; Diane, R.; Ellen, W. Organizational transformation: A model for joint optimization of culture change and evidence-based design HERD. *Health Environ. Res. Des. J.* **2007**, *1*, 40–60.

41. Jones, M.L. Putting Action into Sociotechnical Systems Theory—A Proposed Analysis of the Australian Film Industry Using START. In Proceedings of Transformational Tools for the 21st Century Conference (TT21C2006), Rockhampton, Australia, 24 October 2006; pp. 64–74.
42. Rhodes, D.H.; Ross, A.M. Shaping Socio Technical System Innovation Strategies Using a Five Aspects Taxonomy, 2010. Available online: http://seari.mit.edu/documents/preprints/RHODES_EUSEC10.pdf (accessed on 30 January 2014).
43. Laracy, J.R. Addressing system boundary issues in complex socio technical systems. *Syst. Res. Forum* **2007**, *2*, doi:10.1142/S1793966607000054.
44. Heidegger, M. *Being and Time*; Harper & Row (Originally published in German, in 1927): New York, NY, USA, 1962.
45. Edwards, M.G. The integral holon: A holonomic approach to organisational change and transformation. *J. Organ. Chang. Manag.* **2005**, *18*, 269–288.
46. Wells, S.; McLean, J. *One Way Forward* to beat the Newtonian habit with a complexity perspective on organisational change. *Systems* **2013**, *1*, 66–84.
47. Midgley, G. The sacred and profane in critical systems thinking. *Syst. Pract.* **1992**, *5*, 5–16.
48. Alexander, I.F. A taxonomy of stakeholders human roles in system development. *Int. J. Technol. Hum. Interact.* **2005**, *1*, 23–59.
49. Ya, S.; Rui, T. The Influence of Stakeholders on Technology Innovation: A Case Study from China. In Proceedings of the 2006 IEEE International Conference on Management of Innovation and Technology, Singapore, 21–23 June 2006.
50. Yang, L.-R.; O'Connor, J.T.; Chen, J.-H. Assessment of automation and integration technology's impacts on project stakeholder success. *Autom. Constr.* **2007**, *16*, 725–733.
51. Hughes, R. *Environmental Impact Assessment and Stakeholder Involvement*; IIED: London, UK, 1998.
52. Woodard, D.G. Is the Natural Environment a Stakeholder? Of Course it is (no Matter What the Utilitarians Might Say)! In Proceedings of the Critical Perspectives on Accounting Conference, New York, NY, USA, 25–27 April 2002.
53. Steensma, H.K.; Marino, L.; Weaver, K.M.; Dickson, P.H. The influence of national culture on the formation of technology alliances by entrepreneurial firms. *Acad. Manag. J.* **2000**, *43*, 951–973.
54. Thorburn, G.L.S. Technology and Culture: A Circle of Influence. 2010. Available online: http://nuglobalstudies.wordpress.com/global-studies/gls_499–2/ (accessed on 30 January 2014).
55. Weick, K.E. Organizational culture as a source of high reliability. *Calif. Manag. Rev.* **1987**, *29*, 112–127.
56. Norros, L. Acting under Uncertainty. The Core-Task Analysis in Ecological Study of Work. 2004. Volume Publications 546. Available online: http://www.vtt.fi/inf/pdf (accessed on 30 January 2014).
57. Scheer, A.W.; Nüttgens, M. *ARIS Architecture and Reference Models for Business Process Management*; Springer: Berlin/Heidelberg, Germany, 2000.
58. Per, K.; Kruchten, P. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*; Addison-Wesley Professional: Boston, MA, USA 2003.
59. Vos, J.-P. *The Making of Strategic Realities: An Application of the Social Systems Theory of Niklas Luhmann*; Technische Universiteit Eindhoven: Eindhoven, The Netherlands, 2002.

60. Riener, A.; Ferscha, A. *Enhancing Future Mass ICT with Social Capabilities. Co-Evolution of Intelligent Socio Technical Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 141–184.
61. Di Maio, P. 'Just Enough' Ontology Engineering. In Proceedings of the International Conference on Web Intelligence, Mining and Semantics, Sogndal, Norway, 25–27 May 2011.
62. Di Maio, P. Towards Just Enough Ontology Engineering. Cutter Consortium Reports. 2009. Available online: http://www.cutter.com/bia/fulltext/reports/2009/03/index.html 2009 (accessed on 30 January 2014).
63. Di Maio, P. Just Enough Ontology for Innovation in Systems Engineering. 2010. Available online: http://www.incose.se/eusec2010/Tutorial%203%20Just%20Enough%20Ontology%20for%20Innovation%20in%20Systems%20Engineering.pdf (accessed on 30 January 2014).
64. Di Maio, P. Just Enough Ontology. UK Ontology Network, 2014. Available online: http://ced.aston.ac.uk/ukont2014/programme.html (accessed on 30 January 2014).