Multi-paradigm Modelling for Policy-driven Socio-technical Systems

Ada Diaconescu[1], Dominique Blouin[1] and Alice Ludvig[2]

[1]LTCI Lab, Telecom Paris, Institut Polytechnique de Paris, 19 place Marguerite Perey, 91120, Palaiseau, France
[2]University of Natural Resources and Life Sciences BOKU, 33 Gregor-Mendel Street, 1180, Vienna, Austria

Abstract

Today's socio-technical systems, intertwined with fast-changing cyber-physical infrastructures, are becoming increasingly complex and must be able to cope with major unexpected events - e.g. health, economic and ecology crisis; digital innovation, pandemics, etc. As a result, these systems, which are governed by policies, experience rapid developments in their regulation requirements. In this context, traditional policy-making processes are slow relative to the present pace of changes they face, often leading to outdated inappropriate governance. We present a research plan for developing a highly configurable framework and set of tools to help policy making by providing support to better specify, analyse, monitor and assess socio-technical systems, taking into account governing policies and their impacts. The framework is inspired from model-based systems engineering approaches, which have been successful for Cyber-Physical Systems, to better formulate, characterise and analyse socio-technical systems and their governing policies. It makes use at its heart of Multi-Paradigm Modelling, to develop, reuse and integrate appropriate domain-specific modelling languages and tools, views and analyses to better address policy making of nowadays rapidly changing complex heterogeneous socio-technical systems.

Keywords

Socio-Technical Systems, Cyber-Physical Systems, Self-Adaptive Systems, Multi-Paradigm Modelling, Model-Based Systems Engineering, Model Analyses, Simulation

1. Introduction

Modern societies, increasingly intertwined with fast-changing cyber-physical infrastructures (i.e. hence leading to socio-technical systems), experience rapid developments in their regulation requirements. In this context, traditional policy-making processes are rather slow relative to the present pace of societal changes, often leading to outdated policies – i.e. 'institutional lag'. Evolving such regulations involves several processes, including: collecting sufficient relevant information, assessing the latest changes, determining unsuitable or lacking policies, proposing suitable updates and going through various debating, negotiation and approval processes, before implementing actual amendments or extensions.

AMPM'21: First Workshop in Agent-based Modelling & Policy-Making, December 8, 2021, Vilnius, Lithuania
ada.diaconescu@telecom-paris.fr (A. Diaconescu); dominique.blouin@telecom-paris.fr (D. Blouin); alice.ludvig@boku.ac.at (A. Ludvig)
https://adadiaconescu.there-you-are.com/ (A. Diaconescu); https://www.telecom-paris.fr/dominique-blouin (D. Blouin); https://boku.ac.at/personen/person/7B8ED36D42BDAF0C (A. Ludvig)
0000-0002-2279-0846 (A. Diaconescu); 0000-0001-7606-0251 (D. Blouin); 0000-0003-1323-2276 (A. Ludvig)

As Socio-Technical Systems (STSs) are becoming increasingly complex, so are the policy-making processes that aim to regulate them. This, in turn, leads to situations where policy-makers are overloaded and unable to keep-up with latest developments. Notable difficulties stem from the lack of resources to sustain long-term policies, the preference towards short-term planning within the electoral process, and, more essential with respect to this work, the lack of standardised frameworks and tools for monitoring and assessing policy impacts [1].

Modern Cyber-Physical Systems (CPSs) experience similar challenges, chiefly related to the complexity of their adaptation management processes in a context of rapid and unpredictable changes. Research solutions have been developed to deal with such challenges – spanning from initial data collection, system modelling and analysis (e.g. data-mediation frameworks [2], digital twins [3], Multi-Paradigm Modelling (MPM) [4]) and all the way to runtime decisions and system adaptations (e.g. self-aware [5] and self-managing systems [6], decentralised and multi-scale controllers [7], models at runtime [8], self-integrating systems [9]).

In this vision paper, based on our expertise with some of the above-mentioned topics, and on our experience on policy-making for STSs of the forestry management domain [10, 11, 12], we survey existing systems engineering and self-adaptation solutions for complex CPSs that can potentially be adapted to better develop policies for modern STSs. Our objective is not to propose a new policy making process, but rather to allow improving existing ones using multi-paradigm modelling to support their specification and application, inspired from successful solutions issued from systems engineering and self-adaptive goal-oriented, multi-scale control CPSs, which exhibit characteristics similar to STSs. As a survey of approaches, we do not present any specific case studies but focus on similarities between the domains of STSs CPSs to establish an initial research plan. Developing specific case studies will be part of future work.

We start by introducing minimal notions on STSs and existing approaches for policy making in section 2. We then introduce CPSs and associated successful engineering methods in section 3, including the main approaches on which our framework will build (i.e. Goal-Oriented Requirements Engineering (GORE), Design-Space Exploration (DSE) and Goal-Oriented Multi-Scale Control Systems (GOMSS). In section 4, we introduce enablers for theses approaches (i.e. Model-Based Engineering (MBE) and its Multi-Paradigm Modelling (MPM) extension). We then present the blueprint of our approach based on these preliminary notions in section 5 – this includes the existing modelling languages and tools it may reuse, as well as the languages and tools that we aim to develop for policy making activities and domain-specif modelling in the planned case studies. We conclude the paper in section 7.

2. Policy Making for Socio-Technical Systems

STSs involve wide-spread interactions amongst people and technologies, hence integrating complex social and technical infrastructures with human behaviour. The term STS and its theory was coined during World War II to characterise systems such as coal mines in England [13]. The purpose of STS theory was to study how these systems' management processes could be improved so as to increase overall system performance and quality in people's work lives.

STSs management involves governance, which relies on a set of policies developed by policy-makers. Policy-making is a complex process that involves many stakeholders, often pursuing different or incompatible interests. It may also span over long periods during which stakeholder interests may vary. As policies are strongly context-depend, they should consider, e.g., national, economic, political, cultural and social structures.

For instance, [14] presents an interesting overview of the conceptual and methodological aspects of policy-making for the European Training Foundation (ETF). It introduces basic notions related to policy-making and its different approaches developed over time. We will briefly present some of these notions so that we can later-on emphasise on their similarities with systems engineering methods to illustrate how our proposed framework can build upon these to better support policy making with adequate modelling languages and tools.

Policy-making processes build upon a set of policy analysis activities, which mainly investigate alternative policy options by gathering and integrating the advantages and inconveniences of each option. It is a problem-solving activity that attempts to predict the consequences of alternative courses of action. Numerous perspectives and frameworks exist for policy analysis. The ETF outlines three main approaches – namely, the *analycentric*, the *policy process* and the *meta-policy* – each dealing with problems at three different scales. The *analycentric* approach typically focuses on individual, technical problems at the micro-scale and aims to identify the most effective and efficient solution in technical and economic terms (e.g. the most efficient resource allocation). The *policy process* approach usually focuses on political problems (e.g. involved stakeholders) at the meso-scale, aiming to determine employed processes and means and to explain the stakeholders' role and influences. Problem solutions are identified by changing the relative power and influence of certain groups (e.g., enhancing public participation and consultation). Finally, the *meta-policy* approach focuses on the structural problems of the system and its context, at the macro-scale (e.g. an economic system or political institution). It aims to explain the contextual factors of the policy process– i.e., what are the political, economic and socio-cultural factors influencing it.

Such policy analyses constitute the backbone activities of overall policy-making processes, aiming to specify and validate, at least to a minimal level, the fact that a set of policies fits the problem that must be solved within a targeted STS. Within this overall process, the ETF proposes: first, to use elements of the policy process and meta-policy approaches to set policy priorities; and secondly, to employ the analycentric approach when formulating the actual policy option.

Other policy-making support solutions are also discussed in [14], such as the policy cycle framework and the policy network perspective. The *policy cycle* aims to split complex policy-making processes into manageable steps, breaking them down into sequential stages, examining what happens within each individual stage, and assuming that each stage influences the following one. The *policy network* offers a different way to tackle policy-making complexities. It concentrates on the (meta-)policy process and the relations amongst the actors (i.e. the network) who participate in it, seeking to explain policy outcomes in relation to these characteristics.

The policy-making process proposed by the ETF combines the aforementioned policy cycle with the policy networks. Here, policy networks are approached as a model of collective decision making– an exchange process between actors operating within a market to gain control and influence over resources. This leads to the overall-policy making process depicted in Fig. 1. Fig. 2 shows the detailed steps decomposing the Policy Formulation stage of Fig. 1.

One key aspect as stressed by the ETF is that there is no single or best way to conduct policy

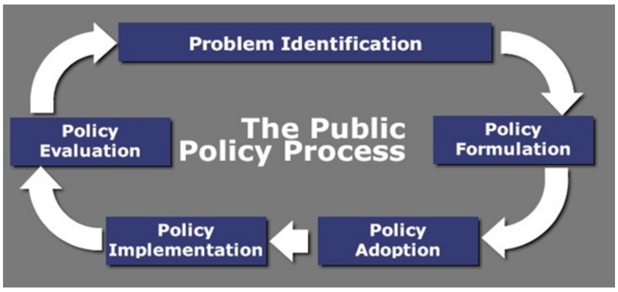

Figure 1: The public policy process proposed by the ETF [14].

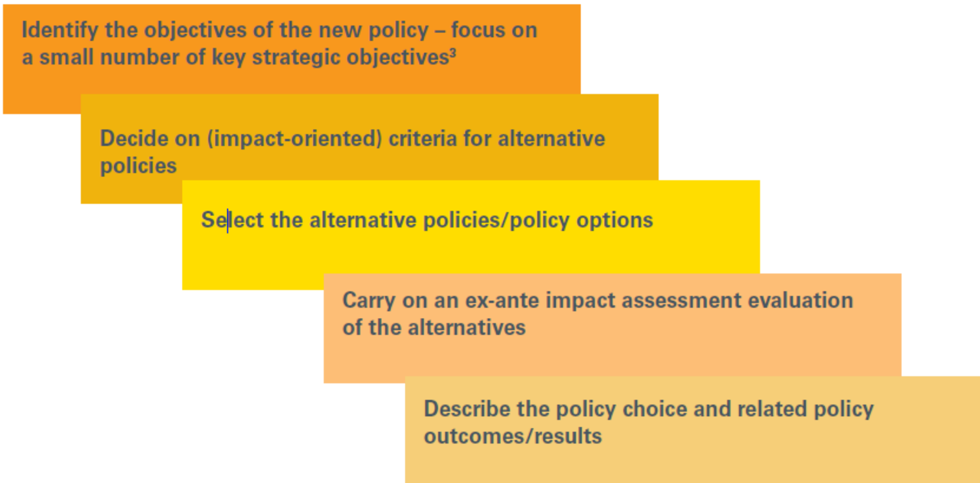

Figure 2: Detailed steps for the *Policy Formulation* stage of Fig. 1 [14].

analyses, due to the multi-faceted nature of policy analyses. Such is also the case for systems engineering where each organisation will typically need to customise standard development processes in order to satisfy organisation and project specific needs. Hence, a policy making framework must allow to seamlessly customise processes and tools by allowing to define and integrate new activities and workflows as needed by the specific policy-making organisations, projects and STSs under consideration.

Another interesting aspect mentioned by the ETF regarding policy analysis is the importance of problem identification, as many policy-making failures are due to solving the wrong problem (rather than to proposing wrong solutions to the right problem). This means that significant effort should go into the formulation of the policy problem. This difficulty is also largely observed in systems engineering, for which several requirements engineering approaches have been developed, which as we will see later could be beneficial when applied to policy making.

Finally, the ETF note that under these circumstances, there is an increasing need to professionalise policy making and its activities to ensure effective governance of these processes and the capacity to anticipate problems. This will help avoiding traps such as policy overload, which happens when governments develop policy plans that are too complex or too vague, containing too many priorities. This results in focus less fragmented priorities leading to endless stream of ad-hoc initiatives. This was illustrated repeatedly during the COVID19 crisis, e.g., in France. The lesson here is that policy plans must be actionable and clear, so as to ensure wide-spread adoption by concerned stakeholders. Therefore, better tool support to policy making is required to help avoid falling into these traps.

3. Engineering Cyber-Physical Systems

Compared to STSs, Cyber-Physical Systems (CPSs) are also complex systems, yet generally excluding the more unpredictable human dimension. The CPS term emerged around 2006 at the National Science Foundation in the United States [15] to identify systems that integrate multi-physical processes (e.g. mechanical, electrical, biochemical) and computational processes (e.g. control, signal-processing, logical inference, planning); and that typically run within uncertain environments. These systems are part of many of our daily activities and drive innovation in important domains (e.g. Automotive, Avionics, Civil Engineering, Industry 4.0, Robotics, smart systems).

While CPS are generally engineered to be predictable – especially for safety-critical systems, e.g. air planes – including humans in the loop brings about new dimensions of complexity and unpredictability. Indeed, the latest trends in CPS development research is to better include human factors – considering a system's human actors and their socioeconomic context [16]. This brings about the new challenge of dealing with highly unpredictable entities. Therefore, CPSs indirectly inherit several characteristics of STSs.

Similar to complex STSs, for which policy-making processes have been developed, well-known engineering processes have been proposed for different kinds of engineered systems; and for CPSs in particular. The most well-known is the Systems Engineering process promoted by the INCOSE (International Council on Systems Engineering) [1].

Systems Engineering is a trans-disciplinary integrative approach covering the entire system life-cycle– starting from its development, its operation, maintenance and all the way to its disposal. It consists of establishing stakeholders' goals and required system functionalities, including an appropriate system life-cycle model, process approach and governance structures; while taking into account the system's complexity, uncertainty, change and variety.

A critical activity in the systems engineering process – often called Design Space Exploration (DSE) – is to consider alternative system designs and configurations, so as to find the best design for the given requirements. Based on these results, system synthesis, verification and validation from the system design can be performed. Fig. 3 illustrates the overall systems engineering process. We can easily map several of its stages to the ones of the policy making process of Fig. 1. For instance, each process starts with a stage including strong emphasis on problem identification, as explicitly stated for policy making. This also takes place for system engineering during the customer needs analysis stage. The Policy Formulation stage on the policy making side, which consists of formulating adequate policies to solve the identified

[1] INCOSE: https://www.incose.org/

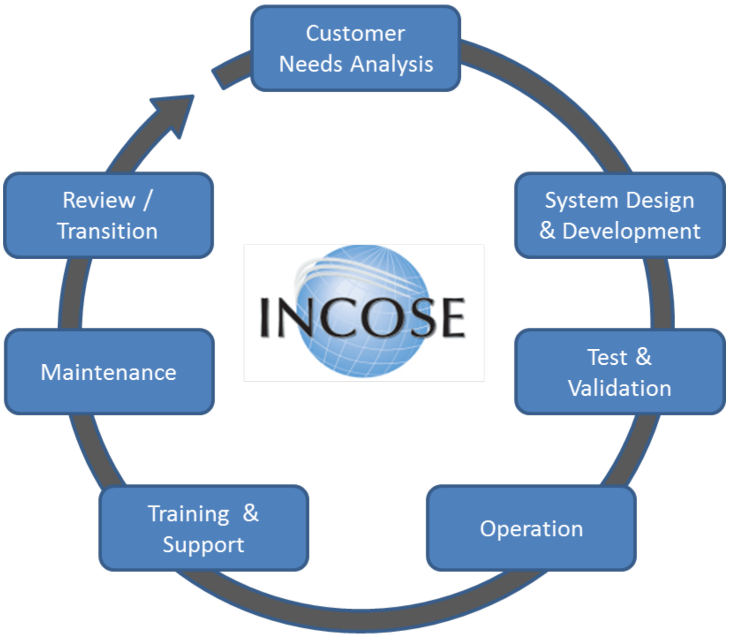

Figure 3: The systems engineering process as proposed by the INCOSE.

problem, corresponds to System Design and Development on the systems engineering side, where engineers Specify a solution to the identified customer problem. We note, from Fig. 2, that each of these solution building stages include alternative solutions evaluation steps in order to come up with the best solution to the specified problems. Policy Adoption and Implementation stages correspond to the systems engineering Tests & Validation and Operation stages, and the Policy Evaluation stage to the Review / Transition stage of systems engineering. Finally, we note that each process circularly comes back to its initial problem identification stage with the whole process being applied iteratively for both policy making and systems engineering cases.

3.1. Goal Oriented Requirements Engineering (GORE)

As we have seen previously, similar to policy-making, systems engineering also emphasises the problem and solution domains. The problem phase has been given high attention by the Requirements Engineering (RE) community. The notions of goals, or requirements, were developed to specify the problem with the solution expressed via system architecture. However, it is difficult to set a precise boundary between the problem and solution parts. Developing a solution is iterative, and design choices often implying new more detailed requirements to be added to the problem part, and so on.

Nevertheless, experience showed that most problems are introduced during the RE stage. This triggered the development of Goal Oriented Requirements Engineering (GORE) approaches [17, 18, 19, 20]. For instance, the KAOS approach defines four complementary and interrelated views on the system and its environment:

- *Goals* representing owners, users, business managers, regulations, etc., which are analysed taking into account conflicts and their resolution.
- *Responsible agents*, both human and machine from the system and its environment, which captures system structure in terms of subsystems and their interactions.
- The specific *problem domain* represented by concepts and their relationships.
- *Agent behaviours* that agents must exhibit in order to achieve goals as well as possible.

The combination of these different views on the system allowed developing several activities to better analyse and validate requirements and significantly improve system design and development.

3.2. Goal-Oriented Multi-Scale Control Systems (GOMSS)

While GORE traditionally focuses on the system's offline specification and development, the high unpredictability of execution environments requires to push these processes into the run-time (i.e. during system execution). This is the case for most complex STSs, e.g. smart homes and cities, power grids, autonomous vehicle networks, robotic swarms and integrated Industry 4.0 systems-of-systems.

To deal with runtime changes, while requiring minimal human intervention and ideally no service disruption, such systems must adapt themselves to dynamic changes in their internal resources, execution environment and even targeted goals. Several relatively recent research areas have been tackling this challenge, notably including Autonomic Computing (AC) [6], Organic Computing (OC) [21], Self-Adaptive Systems (SAS) and Self-aware Computing (SeAC) [5].

To enable self-adaptation, systems generally feature internal control feedback loops – for system and context *monitoring*; problem detection and *analysis*; solution *planning*; and *execution*. Such feedback loops rely on various types of *knowledge* (e.g. system, environment and goal models), which can be updated and extended at runtime (e.g. learning).

In addition to specifying the design and algorithms of such control feedbacks (i.e. defining *how* the system should self-adapt), more recent solutions have also considered explicitly specifying the goals of such control feedbacks (i.e. *what* the self-adaptive system should achieve). Hence, goal-oriented self-adaptive systems go beyond the dynamic selection of predefined adaptive behaviours, allowing the system to learn and discover new adaptation behaviours, as suited for dealing with problematic situations that couldn't be predicted at design-time. E.g. [22] propose to enable systems to self-integrate from an extensible set of control components, which can be discovered at runtime, so as to achieve explicitly-specified goals when new problems occur.

Considering the very nature of most complex systems (i.e. typically composed of a large number of interconnected entities), the *scalability* of self-adaptation solutions is always a major issue. Various forms of hierarchical approaches have been proposed to handle this issue, either via generic architectures (e.g. [23]) or domain-specific designs (e.g. traffic control [24] and manufacturing [25]). Drawing inspiration from both natural and artificial domains, the Multi-Scale Feedbacks (MSF) design pattern was proposed to offer a generic, reusable solution to control scalability problems across various contexts [7]. Hence, goal-oriented, multi-scale self-adaptation solutions (e.g. [26, 22]) combine goal-orientation – for enabling new self-adaptation behaviours to be identified during runtime; with multi-scale designs – for ensuring the scalability of self-adaptation processes.

4. Systems Engineering Enablers

Traditional systems engineering processes such as the one mentioned above used to be mostly supported by natural language documents or at best structured documents such as spreadsheets. Given the increasing complexity of systems, this turned out to be unmanageable to build today's systems at affordable costs. This triggered the development of the Model-Based System Engineering paradigm, where natural language documents are replaced by models of Domain-Specific Modelling Languages (DSML).

Models have been used for a long time to help us understand and analyse complex systems, processes, or artefacts that we are studying, interacting with, managing or developing (e.g. geographical maps, construction blueprints). They are abstractions of reality built for particular purposes [27, 28]. Representing a simplified reality, a model is easier to process than the real thing. However, the way in which a model abstracts reality must be carefully chosen according to the model's purpose. For example, a road map will abstract the land as a set of two dimensional lines indicating road paths so that people can find their way to go from one place to another.

To be usable, models must be understandable in some way. Hence, models must be expressed in some modelling language (e.g. the map's legend) that must be known to understand the models. Such modelling languages can be specified using dedicated metalanguages such as meta-models or grammars. Being expressed in terms of formally specified modelling languages, models can be analysed with tools to automatically detect errors in the system early, which is not possible when only natural language documents are used.

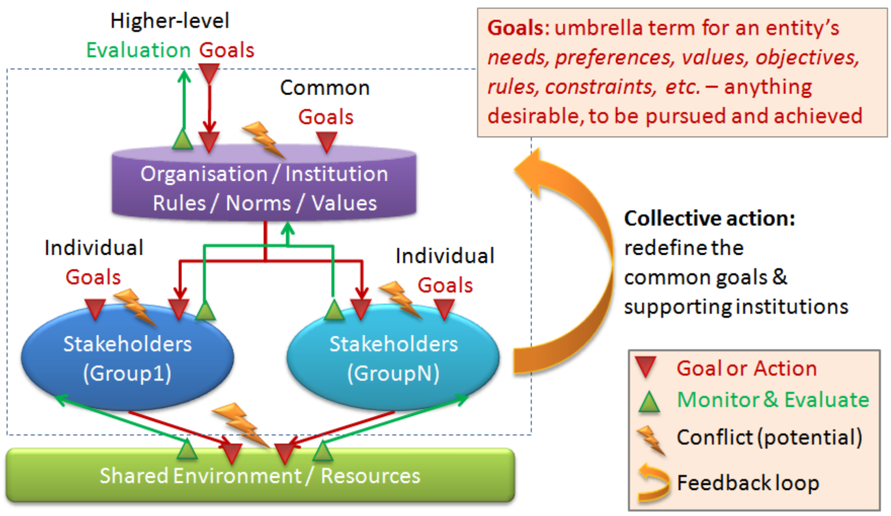

Figure 4: Goal-oriented multi-scale architecture for socio-technical systems

Applied to developing Cyber-Physical Systems (CPS) such Model-Based Engineering paradigm was proven effective leading to better system quality and important cost savings [29]. Besides providing model analysis capabilities, modelling languages also provide a common understanding of the domain intended to be covered by the language. This is achieved through a set of precise concepts and their relationships that formally specify the language and can be used to constitute a common vocabulary for domain actors. It has been shown that expressing something in terms of a properly constructed Domain-Specific Modelling Language (DSML) greatly helps in reducing risks of imprecision, miss-communication and inconsistencies, compared to using natural language. As pointed out in [30], by definition a model is always precise, since its vocabulary is formally defined through a modelling language, being at least precise enough to be understood by computer programs independently of the level of abstraction of the represented reality.

An issue when using such DSMLs is that while expressiveness increases with the specificity of the language, what can be expressed using the language is limited to the covered domain (scope). Therefore, several DSMLs must often be used conjointly to cover all required domains. This observation leads to Multi-Paradigm Modelling (MPM) [31, 4, 32, 33, 34], which advocates the use of a set of modelling languages, each one being most appropriate for the particular subset of activities to be performed with models of the language, rather than trying to build a single monolithic language to support all activities. In particular, Multi-Paradigm Modelling builds upon the following principles:

- Model each part and aspect of a system explicitly, so as to capture all relevant information in terms of formal languages that can be understood and processed by computers;
- Model at the appropriate level(s) of abstraction (or scales) using the most appropriate formalism to avoid introducing accidental complexity.

While the main asset of MPM is to tackle the multi and heterogeneous aspects of systems, it comes at the expense of needing to properly combine them. This is a research topic in its own — model management — where the set of employed models, their modelling languages and the relationships between the models are also formally specified with dedicated DSMLs [33, 35, 36]. Thanks to MPM, language extensibility is supported by allowing the integration of new languages configured for project or organisation-specific needs.

5. Our Approach

Inspired by model-based approaches and paradigms that were successful for Cyber-Physical Systems (as presented earlier) and considering the previously identified commonalities between system engineering and policy making processes, and the commonalities between goal-oriented multi-scale control systems and STSs, we propose to develop a conceptual framework that better supports existing and new policy-making processes specification and application. Such conceptual framework will rely on a set of combined DSMLs and associated tools, to help specify, analyse and understand policy-driven STSs. This will help to detect missing, misapplied or outdated policies; as well as simulate and predict specific effects of various policy updates. This approach will be based on MPM– to benefit from its modelling assets and model integration capabilities.

5.1. Multi-Paradigm Modelling for Socio-Technical Systems

Following MPM principles, we will reuse, adapt, develop and combine the most appropriate modelling languages to cover all aspects of STSs including their policy making processes, as relevant for the modelling activities to be performed at the appropriate abstraction levels. This will provide a precise, shared vocabulary for the policy-making domain, which is already an improvement (compared to using only natural language documents) as it helps to avoid misunderstandings among project stakeholders. For instance, looking at the policy making approach specified in the ETF document [14], we believe that expressing the presented notions and their relationships formally with an appropriate modelling language (ontology) would enforce making the underlying knowledge more explicit and help developing tools to better support the ETF processes.

Reusing or adapting existing modelling languages instead of reinventing new ones can be very beneficial to policy making by taking advantage of knowledge gained in other domains that experienced similar problems. In particular, for the problem identification and policy formulation phases of the policy making process of Fig. 1, policy makers can benefit from the extensive work of the RE community and in particular its aforementioned GORE approaches. For this, we intend to reuse as much as possible existing GORE languages and tools such as the User Requirements Notation (URN)[2], which incorporates a large part of the i* language and approach [17] to capture and analyse goals, stakeholders, their conflicts, resources and so on. Besides, we will benefit from its use case maps sub-language, which allows specification of use case scenario at a high level of abstraction for early policy making phases, and from its goal-oriented decision making methodology based on Key Performance Indicators (KPI) [37]. Thanks to its jUCMNav tool[3] supporting these features, several analyses to evaluate KPIs can easily be developed and evaluated from goal models. In addition, use case scenarios can be simulated with the built-in simulator to detect inconsistency in system behaviour early, thanks to a simple action language. Several benefits of using this tool early have been shown in [30], among others.

Furthermore, existing successful solutions for adaptation management in complex CPSs will be extended and customised to bring some of the necessary support to modern policy-driven STSs. A starting point will be to propose a DSML to represent, analyse and understand policy-driven adaptations for the specific STS domain (e.g. forestry management, co-design of shared living spaces). Coupled with the aforementioned goal models using MPM, the language will support modelling goal-oriented, multi-scale architectures for controlling complex STSs (e.g. smart homes and power grids [26, 38]), Fig. 4. Here, *goals* are first-class system entities, representing anything that a system is willing to achieve or enforce – e.g. objectives, values, constraints, rules, policies, norms, priorities, actions. Each goal is defined by: i) a specific *evaluation function*, defining how to assess goal achievement; and ii) a *spatio-temporal scope*, defining where and when to achieve the goal.

[2]URN: https://www.itu.int/rec/T-REC-Z.151/en/
[3]jUCMNav: http://softwareengineering.ca/jucmnav/

A system may pursue several goals simultaneously. Goals may be in *conflict* if they pursue incompatible objectives over intersecting spatio-temporal scopes. A continuous control feedback loop aims to achieve each system's goal, based on internal resources and the external context. To reach a system goal, such control loop may aim to achieve 'lower-level' goals first, and so forth, recursively. This leads to a multi-scale organisation of control loops, which run simultaneously and coordinate to achieve goals and manage conflicts.

Modelling complex systems in this manner helps preliminary analysis at design time – by rendering goals, system controllers and relevant context aspects explicit and allowing to detect conflicts and/or tune controllers accordingly. It also becomes essential for system adaptation at run-time, when dynamic changes occur – e.g. detecting new conflicts and adapting controllers to resolve them, when new goals or resources are added-on. The goal-oriented aspect contrasts traditional control approaches based on predefined rules and static system models, which could not adapt to unforeseen changes. The multi-scale design also helps manage system complexity, by abstracting system concepts at different scales and hiding irrelevant details from the scales below – hence applying a divide-and-conquer technique that limits the amount of resources necessary at each scale [39].

We believe that applying these concepts and modelling technique to policy-driven STSs will help formalise policy-management processes and render them more traceable and rational. It will help, for instance: to identify all system goals (objectives, policies, values, constraints, etc); to analyse and identify potential conflicts; to detect missing, misapplied or ineffective policies; and, to simulate some of the effects of various policy changes. Providing a multi-scale design also matches the current structure of most political institutions, where the aforementioned processes may run simultaneously, while dealing with cross-scale conflicts and synchronisation issues. Existing knowledge from cross-domain multi-scale control systems may also be useful here, to better understand and manage such multi-scale policy-making processes.

In order to better guide users in following adopted policy-making processes such as the one depicted by Figs. 1 and 2, our framework will also allow modelling these processes using standard workflow models and tools, adapted to the needed modelling activities, similar to what is proposed in [40, 34]. By guiding stakeholders in performing their modelling and analysis activities thanks to these workflows, we can propose more advanced automatic tools for problem detection, solution proposals and impact predictions. For instance, more sophisticated simulations may be proposed via agent-based modelling (ABM) relying on existing platforms – e.g., Repast[4] (social sciences), GAMA[5] (spatially-aware agents), SARL[6] (holonic agents); NetLogo[7] (general-purpose entry-level ABM platform), JADE[8]. In particular, if other simulating tools are found beneficial, we will use MPM co-simulation approaches [41] to coordinate existing simulators and improve simulation by integrating different paradigms.

---

[4]Repast: https://repast.github.io
[5]GAMA: https://github.com/gama-platform
[6]SARL: http://www.sarl.io
[7]NetLogo: https://ccl.northwestern.edu/netlogo
[8]JADE: https://jade-project.gitlab.io/

5.2. Assumptions, Threats and Limitations

While promising, and despite the observed similarities between complex CPSs (including human actors) and STSs, our approach relies on several assumptions that can only be verified by pursuing the proposed research direction, developing associated tools and applying them to policy making in various real-world STSs. We discuss these assumptions and how they can be mitigated below.

The main question is whether what worked for CPS can also work, at least partially, for STSs. Regarding DSMLs, developing formal ontologies for certain domains has proven rather tedious a process, due to the difficulty in converging on a single set of commonly-accepted domain concepts. For instance, for the CPS domain, modelling languages such as AADL[9] have been developed for more than 15 years. However, mitigating this threat, we note that this situation did not prevent approaches and tools from being developed early-on during the language specification, allowing benefits to be obtained right from the incipient phases. This is also the case for programming languages. For example, discussing the appropriateness of one language over another often leads to long debates without resolution. What matters is that DSMLs, even if imperfect, can already bring benefits.

Besides, while DSML development is a complicated, iterative process, approaches such as [42] exist to take into account users and usage context at the very beginning of DSML development. This will help ensure that the targeted stakeholders, such as policy makers (often with social science backgrounds), will be able to use modelling processes and tools despite their differences from typical model and tools users with engineering backgrounds.

We must also consider the fact that the proposed modelling and analysis processes would be carried-out by human actors (rather than automatic agents) and within open socio-technical contexts (rather than constrained environments that can be completely formalised and controlled). Hence, we aim to make sure that the provided modelling platform offers sufficient flexibility and expressiveness to suit complex unforeseen socio-technical situations; and avoid over-constraining and pre-formatting all possible system representations. Indeed, our intention is to provide a helpful tool, rather than a limiting control framework. Thus, we will ensure that actors can easily extend or update the underlying languages and tools to express exceptional cases and diverse view-points.

Another important aspect is that this proposal can only provide support for achieving well-defined goals, in conformity with well-established societal values. While it can allow actors to change and evolve their goals over time, it provides no magic solution against conflicting goals, misaligned values or lack of political will for systemic change.

Finally, we hope that substantial gain from modelling can be achieved by providing a common vocabulary, facilitating understanding and encouraging better precision in policy specifications; and by providing better support for policy analyses, including simulation. Such benefits must be sufficiently significant to overcome the overheads involved in learning new tools and modelling processes.

[9] AADL: http://www.aadl.info/

6. Related Work on Policy-Making Tools for Socio-Technical Systems

To our knowledge, there has been limited efforts to provide policy-making support tools. For instance, we note the GRACeFUL project (Global systems Rapid Assessment tools through Constraint Functional Languages)[10], which aims to develop smart tools that guide decision makers during the implementation of urban projects. The main objective of this project was to develop a simulation platform for modelling the possible responses of citizens and other stakeholders involved, so that the tool can generate predictions from multiple view-points about various policy implementations. Similar to our approach, one concern of the project was to make this kind of modelling accessible to nonspecialists. This was achieved apparently via a graphic platform that facilitated the development of custom software tools.

We plan to achieve similar advantages with our approach, while adopting MPM at the heart of our language engineering, tool building and model integration processes. Still, while the focus of GRACeFUL was on functional languages, where policy making is seen as a constraints-solving problem, our contribution is considerably wider. It aims to develop not only the decision making support tools but also to establish the necessary foundations and conceptual framework for the policy-making domain; and enable these to be customised and extended for various application domains and project requirements. Moreover, we believe that the policy-making domain will be better addressed by a combination of paradigms rather than by a single one (constraints solving).

Regarding process modelling, [43] investigated the feasibility of using workflow tools to support policy-making processes. It presents lessons learned from cases in four European countries and concludes that the processes show several commonalities (common tasks and sequence of steps) despite that they belong to different policy domains. The authors also identify a lack of any supporting information technology, which prevents transparency in policy making processes. These are shortcomings that our proposal aims to address.

7. Conclusion and Future Work

This position paper identified the need for more formal support in policy-making processes, to help policy-makers deal with inherent complexities: identifying and considering the multiple stakeholders, objectives and viewpoints relevant to the specified policies; taking into account the diverse facets to be managed in the targeted application domain (via the specified policies); understanding the potential effects of policy updates and interpreting the observable effects of existing policies upon their application domains. Drawing inspiration from successful solutions to similar problems in technical systems (e.g. CPS), we propose to develop a conceptual modelling framework and support tools to help such policy-making processes. We rely on our previous experience with: i) domain-specific modelling languages and multi-paradigm modelling; and ii) goal-oriented multi-scale architectures for complex self-adaptive systems. We aim to extend and adapt relevant solutions available from these areas to address the particular challenges of the policy-making domain, within the socio-technical systems context. We believe that

[10]GRACeFUL: http://www.fetfx.eu/project/graceful/

such modelling framework, even if necessarily imperfect and continuously evolving, would provide essential support for improving the efficacy, efficiency, comprehension and traceability of policy-making processes.

References

[1] J. M. Barrutia, C. Echebarria, M. R. Paredes, P. Hartmann, V. Apaolaza, From rio to rio+20: twenty years of participatory, long term oriented and monitored local planning?, Journal of Cleaner Production 106 (2015) 594–607.

[2] P. Lalanda, C. Hamon, An autonomic mediation framework for complex physical environments, in: 2014 IEEE International Congress on Big Data, Anchorage, AK, USA, June 27 - July 2, 2014, IEEE Computer Society, 2014, pp. 506–513. URL: https://doi.org/10.1109/BigData.Congress.2014.80. doi:10.1109/BigData.Congress.2014.80.

[3] I. Errandonea, S. Beltrán, S. Arrizabalaga, Digital twin for maintenance: A literature review, Comput. Ind. 123 (2020) 103316. URL: https://doi.org/10.1016/j.compind.2020.103316. doi:10.1016/j.compind.2020.103316.

[4] M. Amrani, D. Blouin, R. Heinrich, A. Rensink, H. Vangheluwe, A. Wortmann, Multi-paradigm modelling for cyber–physical systems: a descriptive framework, Software and Systems Modeling 20 (2021). doi:10.1007/s10270-021-00876-z.

[5] S. Kounev, P. R. Lewis, K. L. Bellman, N. Bencomo, J. Cámara, A. Diaconescu, L. Esterle, K. Geihs, H. Giese, S. Götz, P. Inverardi, J. O. Kephart, A. Zisman, The notion of self-aware computing, in: S. Kounev, J. O. Kephart, A. Milenkoski, X. Zhu (Eds.), Self-Aware Computing Systems, Springer International Publishing, 2017, pp. 3–16.

[6] P. Lalanda, J. A. McCann, A. Diaconescu, Autonomic Computing - Principles, Design and Implementation, Undergraduate Topics in Computer Science, Springer, 2013. URL: https://doi.org/10.1007/978-1-4471-5007-7. doi:10.1007/978-1-4471-5007-7.

[7] A. Diaconescu, L. J. D. Felice, P. Mellodge, Exogenous coordination in multi-scale systems: How information flows and timing affect system properties, Future Gener. Comput. Syst. 114 (2021) 403–426. URL: https://doi.org/10.1016/j.future.2020.07.034. doi:10.1016/j.future.2020.07.034.

[8] R. B. France, B. Rumpe, Model-driven development of complex software: A research roadmap, CoRR abs/1409.6620 (2014).

[9] K. L. B. et al., Self-improving system integration: Mastering continuous change, Future Gener. Comput. Syst. 117 (2021) 29–46. URL: https://doi.org/10.1016/j.future.2020.11.019. doi:10.1016/j.future.2020.11.019.

[10] J. Wilkes-Allemann, A. Ludvig, S. Gobs, E. Lieberherr, K. Hogl, A. Selter, Getting a grip on negotiation processes: Addressing trade-offs in mountain biking in austria, germany and switzerland, Forest Policy and Economics 136 (2022) 102683. URL: https://www.sciencedirect.com/science/article/pii/S1389934121002896. doi:https://doi.org/10.1016/j.forpol.2021.102683.

[11] G. Weiss, E. Hansen, A. Ludvig, E. Nybakk, A. Toppinen, Innovation governance in the forest sector: Reviewing concepts, trends and gaps, Forest Policy and Economics 130

(2021) 102506. URL: https://www.sciencedirect.com/science/article/pii/S138993412100112X. doi:https://doi.org/10.1016/j.forpol.2021.102506.
[12] A. Ludvig, M. Braun, F. Hesser, L. Ranacher, D. Fritz, T. Gschwantner, R. Jandl, G. Kindermann, T. Ledermann, W. Pölz, K. Schadauer, B. F. Schmid, C. Schmid, P. Schwarzbauer, G. Weiss, B. Wolfslehner, P. Weiss, Comparing policy options for carbon efficiency in the wood value chain: Evidence from austria, Journal of Cleaner Production 292 (2021) 125985. URL: https://www.sciencedirect.com/science/article/pii/S0959652621002055. doi:https://doi.org/10.1016/j.jclepro.2021.125985.
[13] S. Long, Socioanalytic Methods: Discovering the Hidden in Organisations and Social Systems, Karnac Books Limited, 2013. URL: https://books.google.fr/books?id=LWL7krX7jgAC.
[14] Madlen Serban, Supporting Public Policy Making through Policy Analysis and Policy Learning , Technical Report, European Training Foundation, 2015. doi:10.2816/721877.
[15] E. Lee, The past, present and future of cyber-physical systems: A focus on models, Sensors 15 (2015) 4837–4869. URL: http://dx.doi.org/10.3390/s150304837. doi:10.3390/s150304837.
[16] D. Nunes, J. S. Silva, F. Boavida, A Practical Introduction to Human-in-the-Loop Cyber-Physical Systems, Wiley, 2018. URL: https://www.wiley.com/en-us/A+Practical+Introduction+to+Human+in+the+Loop+Cyber+Physical+Systems-p-9781119377788.
[17] E. Yu, N. Giorgini, Paola anb Maiden, J. Mylopoulos, Social Modeling for Requirements Engineering, MIT Press, 2011.
[18] A. Dardenne, A. van Lamsweerde, S. Fickas, Goal-directed requirements acquisition, Science of Computer Programming 20 (1993) 3–50. URL: https://www.sciencedirect.com/science/article/pii/016764239390021G. doi:https://doi.org/10.1016/0167-6423(93)90021-G.
[19] A. I. Antón, C. Potts, The use of goals to surface requirements for evolving systems, in: Proceedings of the 20th International Conference on Software Engineering, ICSE ’98, IEEE Computer Society, USA, 1998, p. 157–166.
[20] J. Horkoff, F. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja, M. Salnitri, L. Piras, J. Mylopoulos, P. Giorgini, Goal-oriented requirements engineering: an extended systematic mapping study, Requirements Engineering 24 (2019). doi:10.1007/s00766-017-0280-z.
[21] R. P. Würtz (Ed.), Organic Computing, Understanding Complex Systems, Springer, 2008.
[22] A. Diaconescu, S. Frey, C. Müller-Schloer, J. Pitt, S. Tomforde, Goal-oriented holonics for complex system (self-)integration: Concepts and case studies, in: G. Cabri, G. Picard, N. Suri (Eds.), 10th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2016, Augsburg, Germany, September 12-16, 2016, IEEE Computer Society, 2016, pp. 100–109.
[23] H. A. Simon, The architecture of complexity, in: Facets of systems science, Springer, 1991, pp. 457–476.
[24] K. Fischer, Holonic multiagent systems – theory and applications, in: P. Barahona, J. J. Alferes (Eds.), Progress in Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 34–48.
[25] M. Ulieru, R. A. Este, The holonic enterprise and theory emergence: On emergent features of self-organization in distributed virtual agents, Cybern. Hum. Knowing 11 (2004) 79–98.
[26] S. Frey, A. Diaconescu, D. Menga, I. M. Demeure, A generic holonic control architecture for heterogeneous multiscale and multiobjective smart microgrids, ACM Trans. Auton. Adapt.

Syst. 10 (2015) 9:1–9:21. URL: https://doi.org/10.1145/2700326. doi:10.1145/2700326.
[27] J. Rothenberg, The Nature of Modeling, John Wiley & Sons, Inc., USA, 1989, p. 75–92.
[28] B. Combemale, R. France, J.-M. Jezequel, B. Rumpe, J. Steel, D. Vojtisek, Engineering Modeling Languages - Turning Domain Knowledge into Tools, Chapman and Hall/CRC, 2017.
[29] J. Hansson, S. Helton, P. Feiler, ROI Analysis of the System Architecture Virtual Integration Initiative, Technical Report CMU/SEI-2018-TR-002, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2018. URL: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=517157.
[30] D. Blouin, H. Giese, Combining requirements, use case maps and aadl models for safety-critical systems design, in: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2016, pp. 266–274. doi:10.1109/SEAA.2016.15.
[31] H. Vangheluwe, J. de Lara, P. Mosterman, An introduction to multiparadigm modelling and simulation, in: Proceedings of the AIS'2002 Conference 9–20 (2002). 35, Press, 2002.
[32] D. Blouin, R. Al-Ali, M. Iacono, B. Tekinerdogan, H. Giese, Chapter 2 - an ontological foundation for multi-paradigm modelling for cyber-physical systems, in: B. Tekinerdogan, D. Blouin, H. Vangheluwe, M. GoulÃ£o, P. Carreira, V. Amaral (Eds.), Multi-Paradigm Modelling Approaches for Cyber-Physical Systems, Academic Press, 2021, pp. 9–43. URL: https://www.sciencedirect.com/science/article/pii/B9780128191057000076. doi:https://doi.org/10.1016/B978-0-12-819105-7.00007-6.
[33] H. Giese, D. Blouin, R. Al-Ali, H. Mkaouar, S. Bandyopadhyay, M. Iacono, M. Amrani, S. Klikovits, F. Erata, Chapter 4 - an ontology for multi-paradigm modelling, in: B. Tekinerdogan, D. Blouin, H. Vangheluwe, M. GoulÃ£o, P. Carreira, V. Amaral (Eds.), Multi-Paradigm Modelling Approaches for Cyber-Physical Systems, Academic Press, 2021, pp. 67–122. URL: https://www.sciencedirect.com/science/article/pii/B978012819105700009X. doi:https://doi.org/10.1016/B978-0-12-819105-7.00009-X.
[34] D. Blouin, R. Al-Ali, H. Giese, S. Klikovits, S. Bandyopadhyay, A. BariÅ¡iÄ‡, F. Erata, Chapter 5 - an integrated ontology for multi-paradigm modelling for cyber-physical systems, in: B. Tekinerdogan, D. Blouin, H. Vangheluwe, M. GoulÃ£o, P. Carreira, V. Amaral (Eds.), Multi-Paradigm Modelling Approaches for Cyber-Physical Systems, Academic Press, 2021, pp. 123–145. URL: https://www.sciencedirect.com/science/article/pii/B9780128191057000106. doi:https://doi.org/10.1016/B978-0-12-819105-7.00010-6.
[35] D. Blouin, Y. Eustache, J.-P. Diguet, Extensible global model management with meta-model subsets and model synchronization, in: GEMOC@MoDELS, 2014.
[36] T. Beyhl, R. Hebig, H. Giese, A model management framework for maintaining traceability links, in: Software Engineering, 2013.
[37] A. Pourshahid, I. Johari, G. Richards, D. Amyot, O. Akhigbe, A goal-oriented, business intelligence-supported decision-making methodology, Decision Analytics 1 (2014). doi:10.1186/s40165-014-0009-8.
[38] A. Diaconescu, J. Pitt, Holonic institutions for multi-scale polycentric self-governance, in: A. Ghose, N. Oren, P. R. Telang, J. Thangarajah (Eds.), Coordination, Organizations, Institutions, and Norms in Agent Systems X - COIN 2014 International Workshops, COIN@AAMAS, Paris, France, May 6, 2014, COIN@PRICAI, Gold Coast, QLD, Australia, December 4,

2014, Revised Selected Papers, volume 9372 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 19–35.

[39] A. Simon, The architecture of complexity, in: Proceedings of the American Philosophical Society 106, 1962, pp. 467–482.

[40] L. Lúcio, S. Mustafiz, J. Denil, H. Vangheluwe, M. Jukss, FTG+PM: An Integrated Framework For Investigating Model Transformation Chains, in: International SDL Forum, 2013, pp. 182–202.

[41] C. Gomes, C. Thule, D. Broman, P. G. Larsen, H. Vangheluwe, Co-simulation: A survey, ACM Computing Surveys 51 (2018) 49.

[42] A. Barišić, D. Blouin, V. Amaral, M. Goulão, A requirements engineering approach for usability-driven dsl development, in: Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2017, Association for Computing Machinery, New York, NY, USA, 2017, p. 115–128. URL: https://doi.org/10.1145/3136014.3136027. doi:10.1145/3136014.3136027.

[43] V. Weerakkody, H. Lee, A. Ko, T. D. Medeni, R. El-Haddadeh, A. Tsohou, K. Al-Yafi, Z. Irani, L. M. Campos, Supporting public policy making processes with workflow technology: Lessons learned from cases in four european countries, Int. J. Electron. Gov. Res. 8 (2012) 63–77. URL: https://doi.org/10.4018/jegr.2012070104. doi:10.4018/jegr.2012070104.