Engineering Systems in Flux: Designing and Evaluating Interventions in Dynamic Systems

Bots, P.W.G.

DOI
10.1007/978-3-030-81159-4_19

Publication date
2022

Document Version
Final published version

Published in
Handbook of Engineering Systems Design

Engineering Systems in Flux: Designing and Evaluating Interventions in Dynamic Systems

21

Pieter W. G. Bots

Contents

Abstract

This chapter discusses the threefold challenge of designing effective interventions in engineering systems that are constantly changing: (1) a designed socio-technical artefact should improve system performance not only under present conditions, but it must also be functional when conditions change, be it autonomously or due to interventions performed by others, and (2) the actual intervention of implementing the artefact should be planned such that it does not disrupt functional processes elsewhere, while (3) the implementation process should be impervious to such *contingent processes*. To meet this challenge, engineers can deploy different strategies: design strategies that will enhance the robustness of an

P. W. G. Bots (✉)
Faculty of Technology, Policy and Management, Delft University of Technology, Delft, The Netherlands
e-mail: p.w.g.bots@tudelft.nl

A. Maier et al. (eds.), *Handbook of Engineering Systems Design*,
https://doi.org/10.1007/978-3-030-81159-4_19

artefact, its flexibility, or its capacity for (planned) evolution; strategies that will stabilise the context of the artefact; and implementation strategies that will contain and shield the intervention. This chapter reviews these strategies, discusses how they relate to systems engineering methodologies, and then highlights exploratory modeling and participatory modeling as methods for ex ante evaluation of interventions in dynamic engineering systems.

Keywords

Adaptive design · Engineering systems · Flexible design · Implementation plan · Institutions · Planned intervention · Robust design

Introduction

To intuitively grasp the concept of engineering systems in flux, consider the following joke:

> A cardiologist's car breaks down and she goes to a mechanic to get it fixed. After everything is done, the mechanic asks the cardiologist, "Here's what I don't understand. I fix engines, and so do you, albeit human ones, so why do you get paid ten times more than I do?" The cardiologist then turns the ignition on and says, "Try it with the engine running." (Anonymous 2019)

The mechanic points out the similarity of their profession: they are both engineers. Indeed, a cardiologist (or more precisely a cardiac surgeon) and a mechanic both perform a planned intervention that typically involves placing an *artefact* (e.g., a valve) in some *target system* (a heart; an engine) such that it affects a target process (pumping; a four-stroke cycle) such that it improves certain *measures of performance* of the target system (ejection fraction and valve gradient of the heart; horsepower and emissions of the engine) typically to enhance the performance of an encompassing *system of interest* (a human body; a vehicle) to serve the needs of some *client* (a patient; a driver).

The cardiac surgeon then pulls a bluff: she suggests that the target system she intervenes in is running during this intervention, whereas in practice she replaces a cardiac valve in the arrested heart while a heart-lung machine is keeping up the entire circulatory system. The actual difference between their engineering jobs lies in the properties of the *system of interest*: the organs of a living body degrade rapidly when its blood circulation is stopped, and the patient will die, whereas a car will function as new even when restarted after an engine overhaul that took weeks to complete.

The main takeaway of this metaphor is that to understand what it means to design in the context of engineering systems *in flux*, and appreciate the various design strategies and methods, key concepts like system, flux, and intervention must be clarified. Section "Engineering Systems in Flux: Some Terminology" therefore provides a basic terminology for this chapter. In the subsequent sections, different aspects of "engineering in flux" are elaborated in more depth and linked to related

bodies of academic thought. Section "Strategies for Designing Artefacts in Systems that Are in Flux" reviews generic strategies for designing artefacts that can cope with flux. Section "Systems Engineering Methodologies: Strategies for Managing Flux" discusses how engineering methodologies relate to flux by considering the complex internal dynamics of systems engineering projects and the structures and strategies for managing them. This highlights the pivotal role of institutional design and how this sets limits to interventions in engineering systems in flux. Section "Modeling for Design and Evaluation of Interventions in Dynamic Systems" then addresses the question how models can support the design and evaluation of engineering interventions despite the uncertainties inherent to flux. Section "Conclusion" concludes this chapter with a summary of the main ideas.

Engineering Systems in Flux: Some Terminology

Being the subject of this entire handbook, the concept of *engineering system* needs no introduction. Typical for engineering systems is that they are human-designed, dynamic systems that have significant human complexity as well as significant technical complexity (De Weck et al. 2011). Dynamics and complexity entail nested structures and processes, both physical and social, that relate and interact in many ways. Being human-designed entails that some subset of these structures is artificial (Simon 1981), i.e., have been intentionally created by humans. Systems engineering, then, is an intentional process of devising and implementing such artificial structures. This implementation process constitutes a planned intervention in the engineering system.

Engineers typically plan and then perform interventions to improve system performance on behalf of some client. What is seen as "measures of performance" and "significant improvement" is defined by the client and will be situated (i.e., relate to a particular subsystem) and subjective (i.e., depend on the client's perceptions and preferences). Given that humans will always seek opportunities for what they see as performance improvement, large-scale engineering systems are in perpetual flux simply because numerous interventions take place concurrently, targeting a variety of subsystems on behalf of a variety of clients. Being interrelated, processes in one subsystem will affect processes in other subsystems, these changes will prompt for new interventions, and so on.

Most artefacts are themselves nested structures, and interventions likewise are nested processes. For the sake of conceptual clarity, a single intervention is assumed to be aimed at improving the measure of performance of one particular process (the *target process*) within some subsystem (the *target system*) and to consist of implementing one particular artificial structure (the *artefact*) by placing it within the target system. This may involve connecting it to the structures – natural or arteficial – that were already in place prior to the intervention, shaping the target process as it was, and in this way co-determining its original performance. With the artefact in place, the target process will be shaped differently and perform better.

The relation between artefact and target process is called the *function* of the artefact. A *functional artefact*, then, is an artificial structure that is shaping the target

process as intended by the engineer. In the same vein, a *process* is considered as functional when it enhances the measure of performance of the client's *system of interest*. Conversely, processes (and the structures shaping them) are considered dysfunctional when they lower system performance, and non-functional when they do not affect the client's interest.

The *target* system need not be chosen by the client. More likely, the client seeks to improve the performance of a larger system: the *system of interest*. The engineer will analyse this system, diagnose which subsystems constrain performance most, and then propose interventions that will improve the performance of these specific subsystems. Based on the engineer's findings, the client typically chooses or prioritises the proposed interventions for these target systems. This may involve trade-offs for the client, as interventions may also affect the performance of processes outside their target system. Such *contingent processes* may also be of interest to the client: directly because they are functional processes as well or indirectly because, although external to the client's *system of interest*, they constitute functional processes for third parties. The impacts – positive or negative – of interventions on processes outside the scope of the client's *system of interest* are called *externalities*. Even when the client is indifferent to the affected parties, systems engineering ethics dictate that engineers should identify and factor in such *externalities* as well.

The humour of the joke of the cardiologist and the mechanic lies in its suggestion of the painful image of a mechanic foolishly inserting his hand into a spinning jumble of interlocking steel parts. For the mechanic, evidently, the target process itself (the engine running) physically prohibits performing the intervention. For the cardiac surgeon, this need not apply. To place an aortic valve, she can even opt for an intervention "with the engine running", as for a minimally invasive transcatheter procedure, the heart need not be arrested. But for an open-heart procedure, she needs to solve the problem of creating suitable conditions for implementing the artefact (anesthetised patient, open chest, arrested heart) while also maintaining adequate performance of *contingent processes* (blood oxygenation and circulation) during the intervention.

Interventions will be more challenging to the extent that they affect or are affected by processes in the target system or elsewhere in the *system of interest*. When the road surface of a motorway in a busy metropolitan area has to be renovated, or a dam is to be constructed in a river, such interventions aimed at furthering the interest of the client (people needing transport, flood protection, irrigation, and hydropower) need to be planned and performed as meticulously as open-heart surgery, or they may actually harm these interests. In both examples, the target process (flowing traffic or water) impedes the intervention, but cannot be stopped (unlike the running car engine). To perform the intervention, the flow must be diverted for some time (similar to the patient's blood circulation). This diversion typically requires additional artefacts, notably temporary structures that deviate the flow from the working area and can be moved over time as the implementation process proceeds step by step.

This highlights that an intervention is itself a process, typically comprising a set of smaller interventions. In addition to placing new layers of tarmac or concrete,

renovating an intersection will, for example, also comprise placing barriers and road signs to deviate the traffic flow, making formwork for the concrete, and making schedules for workers. Barriers and formwork (physical structures), work schedules (institutional structures), and road signs (both physical and institutional) are again artefacts. In this case, barriers and road signs are *transient* artificial structures that are placed within the target system (the intersection) to reshape the target process for a period of time so that it permits performing the intervention. This type of transient artefact, designed to enable the intervention while preserving adequate performance of *contingent processes*, may temporarily lower the measure of performance of the target process (slower traffic flow). The formwork and work schedules are also transient artefacts, but these are designed to enhance the performance of processes that actually implement the new road surface (the *primary* artefact).

The idea of *planned* intervention entails that in addition to the artefact that will enhance the target process, the engineer also designs another artificial structure: the implementation plan. This plan should shape the process of implementing the primary artefact (step by step) according to its design, such that the contingencies and *externalities* of this intervention are minimal (or at least acceptable).

Being a structure designed to shape a process to improve its measure of performance (implementation efficiency), an implementation plan is itself an artefact. Being a prescriptive procedure for human action, an implementation plan is an *institutional* artefact. This highlights that planned intervention takes place within a context of social norms and formal rules (Ostrom 2005). These institutional structures are an intrinsic part of engineering systems.

To become functional, a primary artefact and its implementation plan must *both* be designed in conformance with their "contextual" institutional artefacts. For open-heart surgery, these would include, for example, the ISO 5840 standard for cardiovascular implants and the professional standards and guidelines that shape the cardiac surgeon's clinical practice.

Likewise, a primary artefact and its implementation plan are *both* susceptible to flux, albeit on a difference timescale. Both artefacts should remain functional during their "lifetime", but the "lifetime" for an implementation plan (the time required for surgery and recovery) is typically much shorter than for the primary artefact (5–10 years for tissue valves, much longer for mechanical valves). Also, different types of flux will affect the performance of the two artefacts differently (the implementation process would be greatly disturbed if the patient wakes up and starts moving, whereas after recovery the valve will be insensitive to such movement).

Although interventions constitute a major source of flux in engineering systems, flux evidently also results from a wide variety of natural processes: corrosion, infection, insolation, precipitation, sedimentation, sea-level rise, and demographic developments are but a few examples. All these processes may cause artefacts to become non-functional or even dysfunctional. Designing in engineering systems in flux therefore entails (1) devising a primary artefact that, once implemented, will improve a particular measure of performance of the *system of interest* even when conditions change and (2) devising an implementation plan (plus the transient

artefacts it requires, plus – recursively – their implementation plans) that will ensure adequate performance of the implementation process and *contingent processes*.

The next section reviews five categories of strategies that engineers may adopt to achieve this. These categories are not mutually exclusive. Strategies for designing the primary artefact such that it will function even when conditions change (design for robustness, flexibility, and/or evolution) may be combined with strategies for keeping conditions stable (mitigate flux). Moreover, since implementation plans can be seen as institutional artefacts, the strategies for implementation planning typically reflect strategies from the other four categories. Their recursive application is pervasive and entrenched in systems engineering thinking and practice. The systems engineering methodologies reviewed in Section "Systems Engineering Methodologies: Strategies for Managing Flux" are keen examples of institutional artefacts designed to enhance the performance of intricately nested processes of design and implementation of likewise complex primary artefacts.

Strategies for Designing Artefacts in Systems That Are in Flux

Although the specific measures of performance will vary widely, depending on target system and client, some characteristics of artefacts, such as quality, safety, usability, operability, reliability, and maintainability, are considered to be generally desirable. Some of these "ilities" as De Weck (2011) calls them relate specifically to flux: *robustness* and *flexibility*.

Robustness is the ability of an artefact to function as intended in a wide range of conditions. In other words, a robust artefact is insensitive even to significant changes in its context (e.g., earthquake-resistant buildings) or in the target process it shapes (e.g., power cables that can withstand loads up to several times their nominal capacity). Robustness differs from resilience in that a robust artefact will continue to function even under extreme conditions, whereas a resilient artefact may fail to function but still retain the ability to quickly resume its functioning once conditions have normalised again (e.g., an installation that automatically reboots after a power failure).

Flexibility is the ability of an artefact to respond to a need for different functions. What this entails depends on the phase in the artefact's lifecycle. For the design phase, i.e., when the artefact exists only on the drawing board, flexibility refers to the relative ease with which the conceived artefact can be changed to (also) perform a new function or be connected with other artefacts. A flexible design affords a wider range of interventions. This type of flexibility is called adaptability when it is easy to change the design so that the artefact will perform its original function in a context that sets very different conditions, extensibility when it is easy to change the design such that the artefact can perform new functions in addition to its original function, and evolvability when the design has such generic properties that, over a longer time, it affords successive changes such that new "generations" of artefacts can perform radically different functions. The term agility applies when a design can be adapted or extended in a very short time.

For the operation phase, i.e., when the artefact has been realised and implemented within its target system, flexibility is the ability of the artefact to perform multiple functions, i.e., shape *other* target processes in ways that *also* enhance the performance of the *system of interest*. A smartphone is in this sense very flexible since it affords talking to someone while simultaneously taking a picture and checking one's e-mail or calendar or playing a game. Artefacts that can only perform one function at a time can still be flexible in that their design affords that their structure is changed into different configurations such that it can perform different functions. The Swiss army knife is the iconic example of this type of flexibility, which De Weck (2011) calls reconfigurability. The ability of artefacts to easily adjust to the need to expand its capacity for performing its function is called *scalability.*

De Weck's thorough semantic analysis of the "ilities" of designs and artefacts affords a categorisation of intervention strategies that system engineers may adopt to cope with flux.

Design for Robustness

This category comprises design strategies that anticipate on exogenous change in conditions while assuming that the functions and the client needs they stem from are stable. Although the literature on methodologies for robust design pertains mainly to industrial products (Arvidsson and Gremyr 2008; Christensen et al. 2012), their basic principles – awareness of variation and insensitivity to "noise" throughout all "lifecycle phases" of the artefact – are generic. This applies even more to the design principles that Knoll and Vogel (2009) propose for civil engineering artefacts:

- Focus on loads. Structures must be strong enough to withstand high loads. Identify all functional processes as well as non-functional processes that put strain on structures. Establish the error of estimate for the magnitude of loads.
- Foresee and prevent interior flaws. Identify structural properties that are critical. Challenge why the design makes them strong enough. Make failure/breakdown mechanisms explicit. Pay special attention to structures that are sensitive to error during implementation.
- *Consider* structural hierarchy. Focus on primary structures, i.e., those bearing the main load of the processes they shape. Identify cascading failure mechanisms, i.e., how failure of substructures may cause adjacent structures to fail ("domino effect") and/or cause high loads on structures higher in the hierarchy (escalation).
- Foresee external causes. Identify processes and events that may cause exceptional loads on structures. Gauge the extent of such loads and formulate "maximum credible events". Consider the effects of such events when they occur simultaneously ("worst case" scenarios).

Practicing these principles will reveal which system components are critical, and this will prompt designers to consider alternative strategies for making these components less prone to failure. Two common strategies for achieving this are

over-dimensioning (designing structures to withstand loads well beyond their original specifications, possibly even beyond those foreseen in the "worst case" scenario) and *redundancy* (duplicating system components such that their function remains fulfilled in case a component fails). Design strategies to prevent, or at least contain, cascades of failing structures include periodically adding strong elements ("zipper stoppers") among clusters of brittle elements to stop the progression of the failure, and placing structures designed to fail ("sacrificial structures") in order to protect the rest of the structural system from excessive loads (e.g., fuses, circuit breakers, pressure valves). Still, artefacts that have been designed for robustness may become brittle over time, not only because loads grow to exceed their planned capacity but also as its structures are altered such that the assumptions that were true at design time no longer hold.

Design for Flexibility

This category comprises design strategies that anticipate change in the functions and/or capacity of the artefact in response to changing client needs. Cardin (2014) has synthesised a wide range of such strategies (design methods, procedures) in an action-oriented framework that distinguishes five design activities that aim specifically at identifying and utilising opportunities for making a design more flexible:

1. *Create a baseline design.* This design should still be conceptual, so focus on design concepts that address high-level functional requirements. Consider existing designs, but ignore their detailed functional specifications, load estimates, and constraints that may have been provided by the client. The set of design concepts ("design architecture") must be specific enough (e.g., a detailed sketch or physical prototype) to allow consideration of uncertainty and flexibility in activities 2 and 3.
2. *Recognise uncertainties.* Identify uncertain factors that will affect the performance of the artefact in any phase of its lifecycle. Consider endogenous factors (related directly to the artefact, and the organisations involved in its design and construction) as well as exogenous factors (related to users, markets, politics and culture). Model the identified uncertainties such that their consequences can be assessed in activity 4.
3. *Generate flexibility concepts.* Distinguish between flexibility of the design and flexibility of the artefact. Develop design concepts as combinations of a *strategy*, i.e., the process by which the artefact will adapt in response to future events uncertainty, and an *enabler*, i.e., the structural elements in the design that afford this adaptation and how it is managed.
4. *Explore the design space.* Develop quantitative procedures to evaluate the lifecycle performance of a design. Assess which flexibility concepts provide better lifecycle performance relative to the baseline design. Use this assessment to select high-potential enablers, and formulate decision rules for when to apply the associated strategy.

5. *Manage the process.* This applies to activities 1 through 4 in the design process but also to processes of implementation, operation and decommissioning of the artefact. For the design process, process management entails motivating stakeholders (client, corporate management, designers, market analysts) to think in terms of "flux and flex", stimulating creativity as well as rigorous methods for evaluation under uncertainty. For the operation and decommissioning, it entails knowing the designed-in flexibilities and monitoring triggering conditions for exercising them.

Flexibility enablers can be found by analysing the baseline design to identify design variables that are most sensitive to changes in client needs or that when changed will cause need for more changes. Reconsidering the structural hierarchy of the baseline design and the interfaces between subsystems in the baseline design is also a good heuristic for localising flexibility enablers.

Adaptability may be increased by adding "real options" (De Neufville et al. 2006), i.e., investments that are not of immediate value but will permit (or greatly reduce the cost of) modification or expansion sometime in the future. The enabler for such options can be (a combination of) over-dimensioned structures (e.g., the main arteries in a network, or the foundation of a building) that permit upscaling or extension, reconfigurable structures that permit adaptation to different market demands (e.g., office buildings that can be converted into apartment buildings), and modular structures that permit efficient decommissioning and reuse of components (e.g., vehicles designed for disassembly).

Design for Evolution

Where design for robustness and design for flexibility can be seen as *hedging* strategies that aim to mitigate the consequences of flux for planned intervention, design for evolution can be considered as a *shaping* strategy (Dewar 2002) as it aims to harness contextual processes of change as part of the intervention. Such strategies can be particularly effective when the contextual dynamics are well understood, affording adequate prediction of the evolution of a functional artefact. A small-scale example of this design strategy is "tissue engineering", where a degradable scaffolding structure is placed in a human body to shape cell growth processes to form new bone, skin, or heart valves (Neuenschwander and Hoerstrup 2004). On a much larger scale, "Building with Nature" projects (Van Slobbe et al. 2013; De Vriend et al. 2015) harness slow natural hydro-morphological processes to form structures that mitigate erosion and flood risk.

For large-scale engineering interventions such as infrastructure development and city planning, evolution of the artefact mainly depends on social processes. Human agency makes the circular causation in the development of urban areas and infrastructures even more complex (and hence less predictable) than the feedback mechanisms in natural processes (Gifford 1995). The interactions between actors (planners, designers, contractors, operators, users) cause emergence of patterns

perceived by these same actors, and this (re-)interpretation of the system causes them to alter their interactions, giving rise to new patterns, and so on (Holtz et al. 2015; Portugali 2000, 2008). When designing for evolution, city planners and infrastructure engineers may seek to enhance their capacity for prediction through modeling (cf. section "Modeling for Design and Evaluation of Interventions in Dynamic Systems") but more often will rely on design for flexibility approaches or on incremental approaches based on pilot projects (Vreugdenhil et al. 2010).

In projects embedded in "open source" product development communities (Bonvoisin et al. 2017; Scacchi et al. 2006), the design process itself is evolutionary because it implements the Darwinist principle of evolution through mechanisms of variety and selection. When new requirements emerge, these are broken down into modular tasks and communicated to let community members decide what to work on. Bottom-up integration may rely on a core team of senior community members who, being most knowledgeable and skilled, assess the quality of a contribution before its integration. Alternatively, the integration strategy may also be to permit contributors to integrate their work as they see fit and rely on other members to improve it or replace it by a better contribution. Both strategies reflect that evolutionary design approaches balance capacity for centrally planned and coordinated change (to accommodate the complexity of the task) with the capacity for decentral and incremental change (to accommodate changing client needs).

Mitigate Flux

This category of strategies for dealing with flux fundamentally differs from the previous three in that the strategies aim at reducing or containing the variability in the context of the intervention, rather than at making the artefact insensitive or adaptive to contextual change. Mitigating flux can also be seen as a *shaping* strategy (Dewar 2002) but – quite unlike design for evolution – one that aims to maintain the status quo. Groynes and breakwaters are examples of physical structures designed specifically to protect coasts and riverbanks by mitigating water flows that would otherwise cause erosion. Likewise, shock absorbers can be used to protect more sensitive substructures against abrupt movements.

On a project level, flux mitigation strategies may, for example, seek to limit "scope creep" due to changing client preferences by anchoring specifications and procedures for scope control in contracts (Collyer and Warren 2009). To stabilise the industry sector they are part of, corporate actors use institutional artefacts such as patents, licensing contracts, and standards. Holgersson et al. (2018) demonstrate how (coalitions of) corporate firms in the mobile telecommunications sector used these intellectual property strategies to preserve their dominant position, and how interventions of this type by newcomers can first disrupt and then reform. An apparently paradoxical finding is that when disruption leads to a shift from soft institutions (implicit contracting and gentlemen's agreements that rely on social norms) to hard institutions (formal rules embedded in patents and licensing contracts

and enforced through litigation), dynamics increase and stability decreases. When "patent wars" increase the transaction costs (North 1990; Williamson 2000) in an industry sector to the level where they impair new product development, the sector will design formal institutions that increase stability, such as technological standards coupled with the obligation for all firms to license standard-essential patents at fair, reasonable, and non-discriminatory terms.

The study by Holgersson et al. (2018) shows that coalitions of firms can use institutions both to mitigate flux and to stimulate flux. When striving to maintain a monopolistic position, they will design propriety standards and develop restrictive patent licensing strategies; when aiming to stimulate other firms to adopt and extend their technologies, they will use liberal licensing strategies and promote open standards. This reflects that institutional design (Alexander 2005; Koppenjan and Groenewegen 2005) within engineering systems may focus on institutional structures that provide a relatively stable context for processes of systems engineering but also on strategies that stimulate technological innovation. Systems engineering methods as strategies for managing flux will be reviewed in the next section. Strategies for inducing flux, for example, to stimulate innovation, are beyond the scope of this chapter.

Design of Implementation Plans

An implementation plan should shape the process of performing a planned intervention in an engineering system in flux such that (1) it is effective, i.e., implements a functioning artefact; (2) it has limited negative impact on the performance of the target process and *contingent processes*; and (3) it delivers on time and within budget.

The first two requirements relate to flux in the sense of intervening "with the engine running". When these requirements are not critical, engineers are likely to take the approach of the mechanic repairing an engine because this is more efficient. This interruption strategy means halt the target process, typically using transient structures to isolate the target system from the larger *system of interest*; then implement the artefact; then reconnect the target system; and finally, restart the target process. But when the intervention must be performed without interrupting the target process, this typically requires some form of redundancy. Depending on the target system situation, one of the following strategies can be adopted:

- Augmentation strategy. Create in situ the additional structures that will enhance performance of the target process. Test, and then deploy these new structures by connecting them with the larger *system of interest*. This strategy is feasible when the target system is sparse in the sense that it provides ample space for implementing additional structures while keeping the current structures intact and functioning. Typical examples are adding new servers to a data centre or expansion of networked infrastructures (rail, road, cables, pipelines) when additional lines can be built along new trajectories or in parallel to existing ones, and

their connection to nodes on either end can be a controlled and virtually instantaneous operation.

- Substitution strategy. Use redundant capacity of existing structures in the *system of interest* to keep up the performance of the target process, or implement new transient structures that can achieve this for the duration of the intervention. Then perform the planned intervention in the target system using an interruption strategy. Then when the (now enhanced) target process has been resumed, remove the transient structures. This strategy is feasible if the *system of interest* can temporarily provide the required additional capacity or space. In a meshed transport network, traffic can be rerouted. The hard shoulder of a motorway can be used to compensate for the traffic capacity that is lost while reconstructing the pavement of a lane. A heart-lung machine affords open-heart surgery because it can substitute the circulation and blood oxygenation functions of these organs. Reservoir engineers will use redundant capacity when geological conditions allow diverting a river away from the build site via an adjacent valley or create such capacity by digging tunnels.
- Piecemeal strategy. Reduce the impact of the intervention on performance by splitting the intervention into a series of smaller ones that, because of their limited scope in time and space, are easier to perform with a substitution strategy or have less impact on system performance when performed with an interruption strategy. Piecemeal strategies evidently work well for implementing modular artefacts such as NASA's International Space Station that have been designed such that implemented component modules can function independently from the modules still awaiting their implementation. Another example is the timed implementation of software updates for operating systems of smartphones: rolling out an update in phases, each phase targeting a specific user group controls not only the load on the software servers but also the disruption of the target system.
- Control/mitigation strategy. Condition processes in the context of the target system such that they interfere less with the implementation process and/or are less sensitive to interruption of the target process. Heart surgeons administer medication that will slow down the patient's heart rate to facilitate a minimally invasive procedure. System operators and service providers typically schedule and announce maintenance windows so that users can anticipate and shift critical processes to other moments. System engineers smoothen transitions to new technologies by announcing deprecation of standards well in advance but also design artefacts with "forward compatibility" to prolong their operational lifetime.

The part of the implementation plan that structures the "core" intervention – implementing the artefact within the target system – generally reflects the structural hierarchy of the primary artefact, simply because realisation of an artefact entails realisation of its parts. Hence subsystems imply implementation sub-processes. But the implementation planning strategies show that implementation entails additional processes. Some structures in the target system may need to be modified to redirect the target process or to achieve that the primary artefact can be connected to them. In addition, the transient structures needed to implement the primary artefact, or to

mitigate interference with *contingent processes*, must also be implemented (and eventually removed). Designing these additional processes – recursively – as planned interventions (which implies also considering and resolving their impact on *contingent processes*) will eventually produce a complete set of implementation processes.

Given this set, project planning methods like PERT/CPM (Moder et al. 1983) are useful to improve implementation performance in terms of time and budget. The project planning term for the decomposition of a process into sub-processes is activity breakdown structure. The bottom layer of this breakdown defines the "atomic" sub-processes (*activities*). Planning adds the fourth dimension: time. The hierarchical relation of an activity breakdown structure does not determine the precedence relation between the activities; it merely defines them as "pieces of the puzzle". Planners establish the precedence relation by checking for each activity X which other activities *must* have been completed before X can be performed. Larger substructures must typically be implemented before their smaller substructures can be connected to them. The resulting precedence graph allows planners to plan activities in parallel and apply the critical path method (CPM) to minimise overall project time.

Implementation plans are institutional artefacts and hence must themselves be "implemented" within existing institutions, both formal (contracts, permits, labour laws, safety regulations) and informal (common social routines and professional practices). Ideally, they should be compatible with the plans for other interventions, but the image of workers breaking up a newly paved street for lack of coordination between the roads department and the water and sewer department is – alas! – all too familiar. A rigorous project plan with an elaborate activity breakdown structure optimised for efficiency may lack resilience. Just like physical artefacts, an implementation plan should preferably be robust and flexible. In fact, each of the four categories of design strategies reviewed earlier in this section will help design implementation plans that can cope with flux. Pilot projects serve as "sacrificial structures". Forward compatibility can be seen as a "real option". Adding slack resources to critical steps in an implementation plan is a form of "institutional overdimensioning" to prevent "cascading failure" of the entire plan.

Systems Engineering Methodologies: Strategies for Managing Flux

Systems engineering methodologies (e.g., Sage and Rouse 2009; Walden et al. 2015) can be seen as institutional structures that have been designed by engineers to shape the processes of designing and performing interventions in engineering systems to enhance their efficiency, i.e., the ratio of the functionality of the artefact over the resources used (time and budget). These methodologies reflect the recursive application of "design thinking" not only to a primary artefact and the artefacts that shape its implementation process (the implementation plan and transient artefacts) but also to a third category of artefacts: those that shape the processes of designing the

primary artefact and all other artefacts needed for its effective implementation, operation, and eventual decommissioning. Such "methodological" artefacts typically "codify" best practices as formal procedures and standards that, when enforced, shape the decision processes of engineers as they diagnose and decompose the *system of interest* and conceive, test, and evaluate interventions in identified target systems.

Interventions hinge on changing structures by (re)placing artefacts in selected target systems within the *system of interest* so that the overall performance of the *system of interest* improves. Systems engineering methodologies therefore focus on the primary artefact. They commonly structure the systems engineering process in phases that follow the "lifecycle" of this artefact. Although the number and names of phases vary per publication, they typically follow this pattern:

1. *Inception:* a process of growing awareness of needs that the *system of interest* does not fulfil (unsatisfactory system performance).
2. *Design:* a process of identifying the target system within the *system of interest*, specifying its functions and requirements by operationalising their measures of performance, conceiving and assessing alternative options for improving performance (global design of artefact), and detailing the preferred option (detailed design of artefact and its implementation plan).
3. *Implementation:* a process of realising (in the literal sense of "making real") the design produced in the previous phase, i.e., constructing and deploying the artefact within the *system of interest* as planned.
4. *Operation and maintenance:* a process of keeping the artefact functional so that it shapes processes within the *system of interest* as intended and intact so that it continues to do so.
5. *Decommissioning:* a process of dismantling and/or removing the artefact from the *system of interest* so that it no longer shapes processes within this system.

Systems engineering methodologies focus most strongly on the design phase, as in this phase the processes in the subsequent phases should be anticipated and structured by the design. Although authors emphasise the iterative nature of the design phase, the methodologies aim for closure. They prescribe structures for decision-making processes (Parnell et al. 2011) that generally follow the (bounded) rational *intelligence-design-choice* pattern (Simon 1981) that involves divergence and convergence, but the end product is a design that consolidates the many choices made during the decision process in a design that specifies the artefact in such detail that it can be realised and implemented.

The graphical representation of the V-model of systems engineering (Forsberg and Mooz 1991) in Fig. 1 highlights this decision focus by emphasising the stage gate decision points. Using the terminology of sections "Engineering Systems in Flux: Some Terminology" and "Strategies for Designing Artefacts in Systems that Are in Flux", the first point, at the end of the *Needs Assessment* and *Concept Selection* processes (Phase 1), corresponds to the selection – after analysis and diagnosis of the *system of interest* – of the target process and the "baseline design"

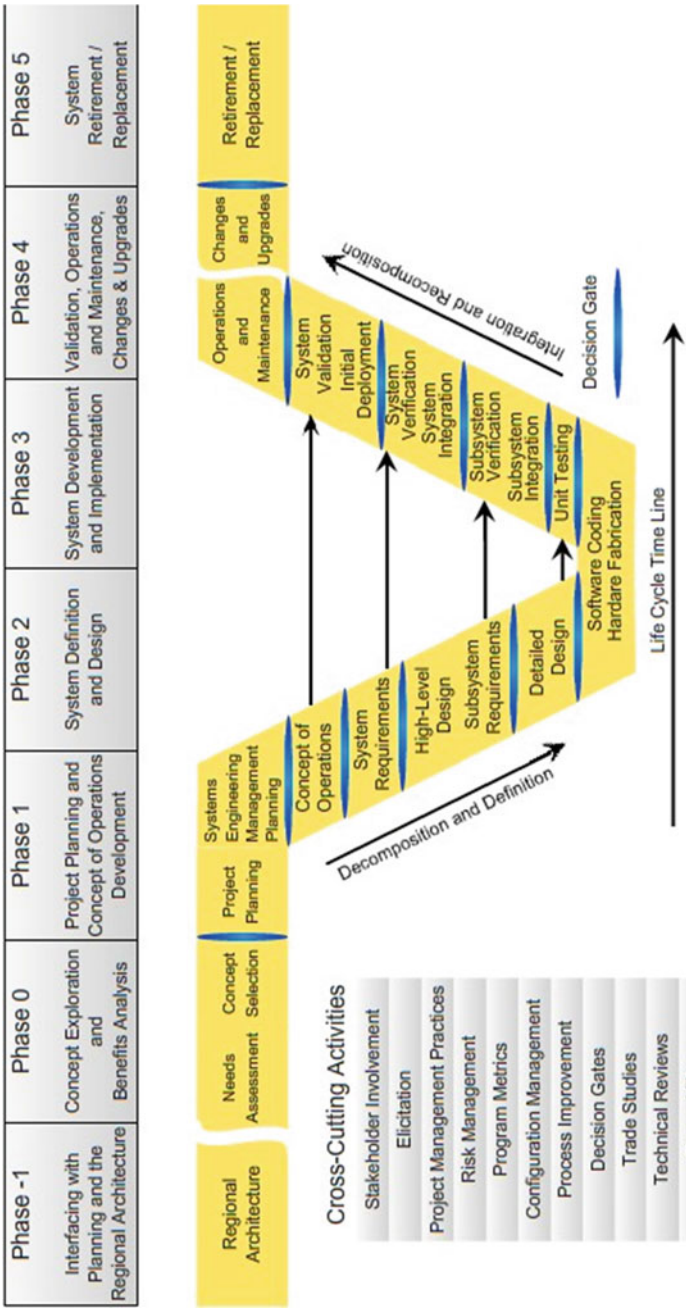

Fig. 1 V-model of the systems engineering process (FHA 2009)

of the artefact. The stage gate decision points at the end of each sub-process in Phase 2 concern the breakdown of the overall intervention into smaller ones, each targeting specific sub-processes with specific substructures that can be designed more or less independently. Likewise, those in Phase 3 mark the closure of steps in the implementation plan. The idea of such stage gate decisions is that at those points in time specific design and implementation choices are "frozen" to provide a stable structure for subsequent decision processes.

The decomposition of the design task typically follows the structural hierarchy of the artefact. Moving along the downslope of the V, the client needs are translated to main functions and requirements, which prompt decomposition into subsystems. Detailing the functions and requirements for these subsystems prompts further decomposition down to the elementary level (bottom of the V), where a system element is an artefact that can be bought "off the shelf" or can be fabricated to specifications. Moving along the upslope of the V, elements are assembled first into units, which are assembled into components, which are integrated further into subsystems until the completed artefact is ready to be deployed.

The diagram in Fig. 1 is limited in that it simplifies the crucial mechanisms of decomposition and integration as two arrows, whereas these mechanisms mean that each "box" in the V comprises a multitude of concurrent design processes and implementation processes, each dealing with one particular subsystem, component, unit, or element of the complex artefact that is being designed and implemented within the target system. Likewise, the horizontal arrows represent a multitude of concurrent processes of validation and testing.

These testing and validation processes at all levels (the horizontal arrows) may reveal unsatisfactory performance (possibly due to evolving needs). This then may call for changes in the design that challenge earlier made choices, not only regarding the tested element, unit, component, or subsystem but possibly also regarding their connected parts.

That changes to the design of one part can call for redesign of other parts highlights that the decision processes of concurrently designing engineers are contingent to the extent that the target processes of the artefacts they are designing are contingent. When such contingencies exist, engineers must coordinate their decision processes so as to ensure that in structural and functional properties, one artefact will not impair those of other artefacts and likewise that their implementation plans do not interfere. Or phrased positively, engineers must coordinate to achieve synergy.

The complexity of a systems engineering project thus has two related dimensions: (1) the multiplicity of structural connections between parts and (2) the dynamic interaction between concurrent design and implementation processes as they progress (Whitty and Maylor 2009). Hence, a strategy common to most systems engineering methodologies is to decompose the artefact so that the resulting hierarchy of substructures minimises the number of their connections and interactions. This reduces the contingencies between processes in the target system, and this will reduce the sensitivity of designs of substructures to changes in the design of other substructures.

Koppenjan et al. (2011) point out that management of large engineering projects requires a capacity for rigorous planning and control as well as a capacity for flexible adaptation to changing conditions and that this leads to contradictory requirements for systems engineering methods. From a *predict-and-control* perspective, project management should focus on front-end analysis to produce precise definitions of project scope, tasks, schedules, and budgets that should be managed tightly through hierarchy and standardised information exchange. From a *prepare-and-commit* perspective, project management should define scope and tasks by setting global terms of reference, accepting that client needs and context will change, and focus on creating horizontal structures for cooperation and learning in the networks of client, team managers, contractors and technology providers.

Although systems engineering methodologies recognise the need for balance between the rigor and adaptiveness, the predict-and-control perspective tends to dominate over the prepare-and-commit perspective. This may be because this perspective is reflected and reinforced by systems engineering standards, such as ISO/IEC 15288 (systems engineering – systems lifecycle processes), which emphasize project management while providing limited coverage of early-stage activities of conceptualising the problem and considering alternative solutions (Kasser 2010). Interestingly, the review of systems engineering standards by Lowell (2009) shows that standards have been developed for specific aspects (quality, reliability, maintainability, producibility, safety; configuration management, parts management, environmental management), but not for (design for) the "ilities" associated with flux.

Meanwhile, the need to respond to contextual changes has led to the development of systems engineering methods that aim to enhance flexibility by speeding up the pace of the design and realisation phases. Examples are rapid prototyping (for software systems *RAD – Rapid Application Development*), the *Dynamic System Development Method*, the *Agile Software Process*, and *SCRUM*. These methods typically reduce the development time by combining lightweight project management, modular process structures, and incremental product delivery based on evolutionary development through many rapid iterations. Such iterative processes permit adaptation to flux but may hamper integration when engineering more complex systems.

Whether rigorous or adaptive, systems engineering methodologies can be seen as strategies for *managing* the flux that is inherent to large-scale systems engineering projects. This flux can be endogenous (design decisions and/or insights from validation and testing that change conditions for contingent design processes) as well as exogenous (changes in the context of the target system and/or changes in client needs and preferences). The management strategies are similar to those discussed in section "Strategies for Designing Artefacts in Systems that Are in Flux". The stage gate decision points are institutional structures that function as "zipper stoppers" that should prevent "cascading failure" of a design, i.e., invalidation of the design of an entire subsystem when only one element or unit fails a test. Adaptable designs will reduce the risk of such failure or at least the time

needed for redesign. Reconsidering the structural hierarchy (subsystem-unit-element) and the interfaces between subsystems may enhance flexibility. Flux mitigating strategies to reduce the need for redesign of system elements and units include enforcing standards and forward and backward compatibility of design and using client contracts to reduce "scope creep".

All systems engineering methodologies have in common that they provide a generic structure or "architecture" that supports coordination of the multitude of concurrent design processes performed by a host of engineering professionals. Coordination of processes requires functional institutions. To improve the engineering practice, public authorities ("top-down") as well as professional societies ("bottom-up") seek to set standards for artefacts and their measures of performance, and protocols for their implementation. Koppenjan and Groenewegen (2005) offer several reasons why this is difficult. Firstly, most institutions that are not mere "rules on paper" but effectively shape social processes as "rules in use" (Ostrom 2005) are the result of informal and incremental processes. It is by such slow processes that institutions gain their legitimacy to constrain social interactions. Unless well embedded in "rules in use", new rules lack this legitimacy, will not become institutionalised, and hence remain ineffective. These properties also explain why institutions (design strategies, systems engineering methodologies, best practices, modeling approaches, standards, policies) that have performed well in one engineering system cannot simply be "transplanted" to other engineering systems (De Jong 2004). Secondly, to fulfil their crucial role as suppliers of stability and predictability, institutions *should* be difficult to change. Being the "rules of the game" (Williamson 2000), they determine the chances for winning or losing, and players will attempt to change them to their own advantage. For this reason, purposefully designed institutional artefacts are typically designed for robustness so that it is not easy to adapt them.

In sum, attempts to create or change institutions can (and often should) be planned similar to (and often as part of) engineering interventions that focus on technical artefacts. By consequence, the capacity for planned intervention in engineering systems in flux depends on the capacity for institutional design.

Modeling for Design and Evaluation of Interventions in Dynamic Systems

Modeling is deeply embedded in engineering practice. Systems engineers use models for a wide range of purposes: analysis of the *system of interest*, design problem definition, conceptual design, requirements specification, testing, implementation planning and risk analysis, and many more. Overviews of modeling techniques and their application can be found in systems engineering handbooks (e.g., Parnell et al. 2011; Sage and Rouse 2009; Walden et al. 2015). The two types of application of computer-based models reviewed in this section relate more specifically to the strategies for design of interventions in engineering systems in flux reviewed in the preceding sections.

Exploratory Modeling

When it comes to modeling in support of design of robust, adaptive interventions that *satisfice* objectives and constraints over a wide range of futures, *exploratory modeling and analysis* (EMA) (Bankes 1993; Marchau et al. 2019) is the present state-of-the-art. Although the EMA terminology reflects that this approach was originally developed to support analysis and design of policies, EMA can be applied to any type of planned intervention. The general concept of (institutional) artefact as defined in section "Engineering Systems in Flux: Some Terminology" is virtually equivalent to the concept of policy as it is used in EMA. By extension, this also applies to implementation plans.

Similar to design strategies for flexibility and robust adaptive implementation plans (cf. section "Strategies for Designing Artefacts in Systems that Are in Flux"), the idea of adaptive policymaking is to plan in advance for policy changes that may be needed in response to future events. An adaptive policy prepares for additional actions (e.g., to seize opportunities or to cope with more stringent constraints) and defines variables ("signposts") that should be monitored to see whether success conditions for the policy are still met or that adaptation is needed. Adaptability may be increased by adding "real options" that afford changes at relatively low cost. Ex ante analysis of opportunities and threats and timing and sequence of policy options produces a "roadmap" into the future. Using the graphical language of a metro map (see Fig. 2), such maps show for each option when (under some class of scenarios) it no longer meets the policy objectives (Haasnoot et al. 2013). These "adaptation tipping points" indicate the need for additional action and can be represented as crossroads that branch to options that are still feasible.

Keeping options open will reduce sensitivity to uncertain assumptions but comes at the cost of lower efficiency. Deep uncertainty prohibits appraisal of this trade-off using traditional expected utility methods (Lempert and Schlesinger 2000). Alternatively, the value of adaptability can be assessed using simulation models to explore *potential* system behaviors. Such "exploratory modeling" helps in specifying appropriate conditions for adapting a policy, by identifying actions and conditions that produce satisfactory results across a large ensemble of scenarios.

Figure 3 outlines the basic idea. The approach assumes that the analyst has a computational model that can simulate the dynamic behavior of the system. Given a

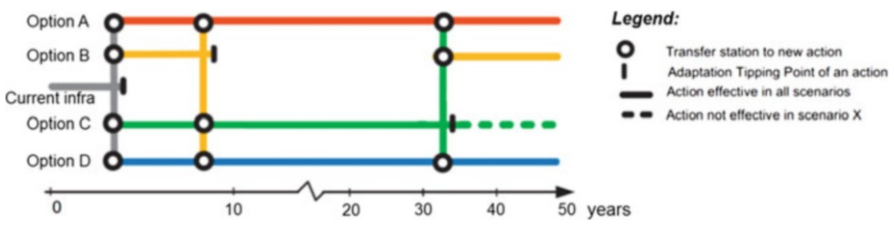

Fig. 2 Example of an adaptation pathways map. (Adapted from http://www.delta-alliance.org/toolboxoverview/dynamicadaptivepolicypathways)

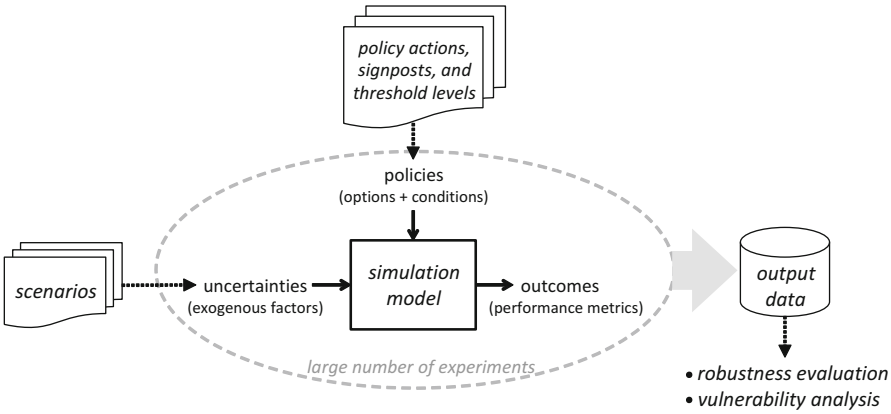

Fig. 3 Exploratory modeling

scenario (assumptions about uncertain exogenous factors in the future) and a policy (a set of policy options and conditional rules specifying when an option is applied) as input, the simulation produces outcomes (performance metrics that reflect how well the policy performed under the given scenario) as output. Repeating this experiment for a variety of policies and a wide range of scenarios (potentially many thousands) generates a large set of output data. These experiments and analyses can be performed efficiently using open source software tools (Kwakkel 2016). These tools support the generation and efficient (parallel) execution of computational experiments using existing simulation models and the visualisation and analysis of their results (identifying key uncertainties, assessing the efficacy of policy options, and iteratively improving the robustness of policies through vulnerability analysis).

Robustness evaluation searches for the policy that performed the best across all of the scenarios, while vulnerability analysis seeks to identify the scenarios in which a particular policy performs poorly, so that policymakers can think of actions that will protect the policy from failing. Robustness can be evaluated using "regret" as measure, where regret is defined as the difference between (a) the performance of a policy in a given scenario and (b) the performance of the best policy in that scenario. The examples in Fig. 4 illustrate (for only a small, two-dimensional scenario space) the regret matrix for three alternative policies, demonstrating that the adaptive policy C is much more robust than the static policies A and B.

Exploratory modeling can support "design for robustness" as well as "design for flexibility" strategies as reviewed in section "Strategies for Designing Artefacts in Systems that Are in Flux" because it provides well-defined quantitative procedures to evaluate the lifecycle performance of a design. Recent developments involving the use of algorithms for multi-objective robust optimisation (Hamarat et al. 2014; Beh et al. 2017) will afford using EMA also for more directed search for interventions that will be effective in an uncertain dynamic context.

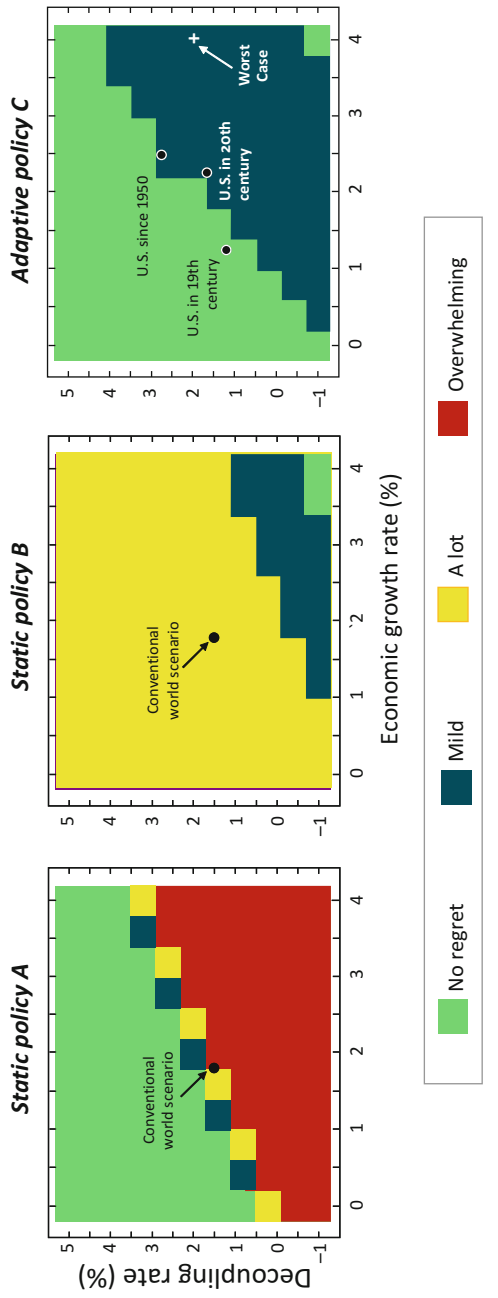

Fig. 4 Robustness evaluation of 3 policies under 121 scenarios. Adapted from Lempert (2004)

Participatory Modeling

Engineering systems comprise technical-physical elements as well as cognitive social actors who are capable of acting and reacting with strategies to the patterns they help create. This adds a layer of complexity not experienced in the natural sciences (Lansing 2003). Agent-based models can capture some of the non-linear effects of socio-technical systems that would otherwise be out of reach. However, computational models have a fundamental limitation because they lack the capacity of humans for "double loop learning", i.e., for reinterpreting their environment, reframing their problems, and developing novel strategies (Argyris 1976).

Recent advances in computation power, visualisation, and human-computer interaction provide new possibilities to make humans (typically representatives of the client) an integral part of an advanced simulation-game model (Mayer 2009; Meijer 2012). Part of the complexity of the system can be modeled and simulated in the computer, while significant dimensions of strategic actor behavior and learning are captured in a social-interactive game. Because they can reveal reinterpretation and alternative uses of artefacts (both technical and institutional), simulations with models of this type afford more realistic ex ante evaluation of engineering interventions. Moreover, directly involving users in modeling activities in early stages of the systems engineering process can improve elicitation of design requirements as well as enrich the set of design concepts.

Similar to exploratory modeling, participatory modeling has its roots in policy development (Pahl-Wostl 2002; Barreteau 2003), but the approach is gaining terrain in support of systems engineering activities (Daniell 2012; Nolte and Herrmann 2016). Where the application of exploratory modeling is mainly limited by the availability of computational resources, the main challenge for participatory modeling lies in organising and managing the process.

Conclusion

Engineering systems are in perpetual flux. While performing countless functions – day to day, minute to minute, or even on millisecond scale – that provide food, shelter, transport, and telecommunication and permit trade and social interaction, these systems evolve over the years as engineers seek to better meet human needs using new technologies. Designing and performing interventions in such intricate and dynamic systems is in many ways similar to trying to fix an engine while it is running. Although at first glance such endeavour would seem absurd, it need not be, provided that the engineers know what they are doing. After removing the right cover plates, a leaking fuel line can be patched. If the engine has more than one cylinder, a sparkplug can be replaced without stopping it, especially when the engine was designed to run on a variable number of cylinders and allow for controlled disabling. And if the engineers know how to keep the larger system of which the engine is part functioning reasonably well without propulsion, they can still opt to

shut it down for some time. Engineers fixing a twin-engine airplane in mid-flight would definitely be spectacular, but voyage repairs on ships at sea are not uncommon. The key to success is knowledge, skill, and a sound plan.

Understanding what such a plan entails and how it can be devised has been the focus of this chapter. When planning an intervention in an engineering system, engineers cope with flux by aiming for robustness, flexibility, and evolvability of their designs while seeking to mitigate flux in the immediate context of their intervention. Each aim calls for particular strategies. These strategies have formed the silver thread for this chapter, as they can be applied to all aspects of design: the artefact (the object that engineers intend to introduce or modify by their intervention), the implementation plan (the organisation in time and space of the intervention and the required resources), as well as the systems engineering methodology (the organisation of the design process through procedures and standards).

Planning entails anticipating future conditions that result from planned actions as well as exogenous changes. Today's massive computational resources allow engineers to test the robustness and flexibility of artefacts as well as their implementation plans by simulating their performance under a vast range of scenarios. Datamining of the simulation results can reveal vulnerabilities that can be remedied by, for example, introducing reconfigurable components or preparing for alternative adaptation paths. To overcome the limitations of computer models as means for anticipating human behavior and social response, engineers have started to directly involve future users and other stakeholders in their simulations. Large-scale simulations based on participatory modeling and serious games may soon become mainstream in systems engineering projects.

That processes of design, implementation, and use of artefacts are entwined is inherent to engineering practice and goes back to prehistoric times. What has changed is the scale and the interconnectivity and hence the complexity and flux of engineering systems. As these continue to grow, so will the challenge of gathering knowledge, acquiring skill, and devising a sound plan so as to make successful interventions. The engineering principles and strategies reviewed in this chapter provide guidance on how to meet this challenge.

Cross-References

- ▶ Choosing Effective Means: Awareness of Bias in the Selection of Methods and Tools
- ▶ Engineering Systems Interventions in Practice: Cases from Healthcare and Transport
- ▶ Evaluating Engineering Systems Interventions
- ▶ Flexibility and Real Options in Engineering Systems Design
- ▶ Risk, Uncertainty, and Ignorance in Engineering Systems Design
- ▶ The Evolution of Complex Engineering Systems

References

Alexander ER (2005) Institutional transformation and planning: from institutionalization theory to institutional design. Plan Theory 4(3):209–223

Anonymous (2019) The joke of the cardiologist and the mechanic. https://www.reddit.com/r/Jokes/comments/6rc9yx/cardiologist_and_the_mechanic/. Accessed 15 June 2020

Argyris C (1976) Single-loop and double-loop models in research on decision making. Adm Sci Quart 21(3):363–375. https://doi.org/10.2307/2391848

Arvidsson M, Gremyr I (2008) Principles of robust design methodology. Qual Reliab Eng Int 24: 23–35

Bankes SC (1993) Exploratory modeling for policy analysis. Oper Res 4(3):435–449

Barreteau O (2003) The joint use of role-playing games and models regarding negotiation processes: characterization of associations. J Artif Soc Soc Simul 6(2):3. https://www.jasss.org/6/2/3.html

Beh EHY, Zheng F, Dandy GC, Maier HR, Kapelan Z (2017) Robust optimization of water infrastructure planning under deep uncertainty using metamodels. Environ Model Softw 93: 92–105

Bonvoisin J, Thomas L, Mies R, Gros C, Stark R, Samuel K, Jochem R, Boujut J-F (2017) Current state of practices in open source product development. In: Maier A, Škec S, Kim H, Kokkolaras M, Oehmen J, Fadel G, Salustri F, Van der Loos M (eds) DS 87–2 proceedings of the 21st international conference on engineering design (ICED 17) Vol 2: design processes, design organisation and management, Vancouver, Canada, 21–25 August 2017. Design Society, Glasgow, pp 111–120

Cardin MA (2014) Enabling flexibility in engineering systems: a taxonomy of procedures and a design framework. J Mech Des 136(1):011005

Christensen ME, Howard TJ, Rasmussen JJ (2012) The Foundation for Robust Design: enabling robustness through kinematic design and design clarity. In: Marjanovic D, Storga M, Pavkovic N, Bojcetic N (eds) DS 70: proceedings of DESIGN 2012, the 12th international design conference, Dubrovnik, Croatia. Design Society, Glasgow, pp 817–826

Collyer S, Warren CMJ (2009) Project management approaches for dynamic environments. Int J Proj Manag 27(4):355–364

Daniell KA (2012) Co-engineering and participatory water management: organisational challenges for water governance. Cambridge University Press, Cambridge, UK

De Jong M (2004) The pitfalls of family resemblance: why transferring planning institutions between 'similar countries' is delicate business. Eur Plan Stud 12(7):1055–1068. https://doi.org/10.1080/0965431042000267902

De Neufville R, Scholtes S, Wang T (2006) Real options by spreadsheet: parking garage case example. J Infrastruct Syst 12(2):107–111

De Vriend HJ, Van Koningsveld M, Aarninkhof SGJ, De Vries MB, Baptista MJ (2015) Sustainable hydraulic engineering through building with nature. J Hydro Environ Res 9(2):159–171

De Weck OL (2011) Life-cycle properties of engineering systems: the Ilities. Chapter 4. In: De Weck OL, Roos D, Magee CL (eds) Engineering systems: meeting human needs in a complex technological world. MIT Press, Cambridge, MA

De Weck OL, Roos D, Magee CL (2011) Engineering systems: meeting human needs in a complex technological world. MIT Press, Cambridge, MA

Dewar JA (2002) Assumption-based planning: a tool for reducing avoidable surprises. Cambridge University Press, Cambridge, UK

FHA (2009) Systems engineering guidebook for intelligent transportation systems, version 3.0. Federal Highway Administration, United States Department of Transportation, Washington, DC

Forsberg K, Mooz H (1991) The relationship of system engineering to the project cycle. In: Proceedings INCOSE international symposium, Chattanooga, TN, 21–23 October 1991. Wiley, Hoboken, pp 57–65

Gifford J (1995) Complexity, adaptability and flexibility in infrastructure and regional development: insights and implications for policy analysis and planning. In: Batten DF, Karlsson C (eds) Infrastructure and the complexity of economic development. Advances in spatial science. Springer, Berlin, pp 169–186

Haasnoot M, Kwakkel JH, Walker WE, Ter Maat J (2013) Dynamic adaptive policy pathways: a method for crafting robust decisions for a deeply uncertain world. Glob Environ Chang 23(2): 485–498

Hamarat C, Kwakkel JH, Pruyt E, Loonen ET (2014) An exploratory approach for adaptive policymaking by using multi-objective robust optimization. Simul Model Pract Theory 46:25–39

Holgersson M, Granstrand O, Bogers M (2018) The evolution of intellectual property strategy in innovation ecosystems: uncovering complementary and substitute appropriability regimes. Long Range Plan 51(2):303–319

Kasser JE (2010) Seven systems engineering myths and the corresponding realities. In: Proceedings of the systems engineering, test evaluation conference (SETE 2010), Adelaide, Australia, 3–6 May 2010. SESA, Barton, pp 1–13

Knoll F, Vogel T (2009) Design for Robustness. Structural Engineering Documents 11. International Association for Bridge and Structural Engineering, Zürich. ISBN: 9783857481208

Koppenjan J, Groenewegen J (2005) Institutional design for complex technological systems. Int J Technol Policy Manage 5(3):240–257

Koppenjan J, Veeneman W, Van der Voort H, Ten Heuvelhof E, Leijten M (2011) Competing management approaches in large engineering projects: the Dutch RandstadRail project. Int J Proj Manag 29:740–750

Kwakkel JH (2016) The exploratory modeling workbench: an open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making. In: Sauvage S, Sánchez-Pérez J-M, Rizzoli A (eds) Proceedings of the 8th international congress on environmental modelling and software, Toulouse

Lansing JS (2003) Complex adaptive systems. Annu Rev Anthropol 32:183–204

Lempert RJ, Schlesinger ME (2000) Robust strategies for abating climate change. Clim Chang 45 (3):387–401

Lowell SC (2009) Standards in systems engineering. In: Sage AP, Rouse WB (eds) Handbook of systems engineering and management, 2nd edn. Wiley, Hoboken, pp 463–478

Marchau VAWJ, Walker WE, Bloemen PJTM, Popper SW (eds) (2019) Decision making under deep uncertainty: from theory to practice. Springer, Cham

Mayer IS (2009) The gaming of policy and the politics of gaming: a review. Simul Gaming 40(6): 825–862

Meijer S (2012) Gaming simulations for railways: lessons learned from modeling six games for the Dutch infrastructure management. In: Perpinya X (ed) Infrastructure design, Signalling and security in railway. InTech Europe, Rijeka, pp 275–294

Moder J, Phillips C, Davis E (1983) Project Management with CPM, PERT and precedence diagramming, 3rd edn. Van Nostrand Reinhold, New York

Neuenschwander S, Hoerstrup SP (2004) Heart valve tissue engineering. Transpl Immunol 12 (3–4):359–365

Nolte A, Herrmann T (2016) Facilitating participation of stakeholders during process analysis and design. In: De Angeli A, Bannon L, Marti P, Bordin S (eds) COOP 2016: proceedings of the 12th international conference on the Design of Cooperative Systems, 23–27 May 2016, Trento, Italy. Springer, Cham, pp 225–241

North DC (1990) Institutions, institutional change and economic performance. Cambridge University Press, Cambridge, UK

Ostrom E (2005) Understanding institutional diversity. Princeton University Press, Princeton

Pahl-Wostl C (2002) Participative and stakeholder-based policy design, evaluation and modeling processes. Integr Assess 3(1):3–14

Parnell GS, Driscoll PJ, Henderson DL (eds) (2011) Decision making in systems engineering and management, 2nd edn. Wiley, Hoboken

Portugali J (2000) Self-organization and the City. Springer, Heidelberg
Portugali J (2008) Learning from paradoxes about prediction and planning in self-organizing cities. Plan Theory 7(3):248–262. https://doi.org/10.1177/1473095208094823
Sage AP, Rouse WB (eds) (2009) Handbook of systems engineering and management, 2nd edn. Wiley, Hoboken
Scacchi W, Feller J, Fitzgerald B, Hissam S, Lakhani K (2006) Understanding free/OpenSource software development processes. Softw Process Improv Pract 11:95–105
Simon HA (1981) The sciences of the artificial, 2nd edn. MIT Press, Cambridge, MA
Van Slobbe E, De Vriend HJ, Aarninkhof S, Lulofs K, De Vries M, Dircke P (2013) Building with nature: in search of resilient storm surge protection strategies. Nat Hazards 66:1461–1480
Vreugdenhil H, Slinger J, Thissen W, Ker Rault P (2010) Pilot projects in water management. Ecol Soc 15(3):13
Walden DD, Roedler GJ, Forsberg KJ, Hamelin RD, Shortell TM (2015) INCOSE systems engineering handbook: a guide for system life cycle processes and activities, 4th edn. Wiley, Hoboken
Whitty SJ, Maylor H (2009) And then came complex project management. Int J Proj Manag 27: 304–310
Williamson OE (2000) The new institutional economics: taking stock, looking ahead. J Econ Lit 38 (3):595–613