PIK - Potsdam Institute for
Climate Impact Research
Member of the Leibniz Association

SEI - Stockholm
Environment Institute

A Formal Framework of Vulnerability
Deliverable to the ADAM Project

Sarah Wolf, Cezar Ionescu, Daniel Lincke,
Sandy Bisaro, Jochen Hinkel, Diana Reckien

FAVAIA Working Paper 6

May 3rd, 2007

FAVAIA
Formal Approaches to Vulnerability Assessment that Informs Adaptation
www.pik-potsdam.de/favaia

This text has been submitted as a deliverable to the project ADAM - Adaptation and Mitigation Strategies: Supporting European Climate Policy. It documents work in progress on the formal framework of vulnerability developed by the FAVAIA group at the Potsdam Institute for Climate Impact Research.

Contents

Executive summary

The notion of “vulnerability” is not well-defined in the climate change scientific community. Recent research has identified a need for formalisation, which would support accurate communication and the elimination of misunderstandings that result from the use of ambiguous terminology. Moreover, a formal framework of vulnerability is a prerequisite for computational approaches to its assessment.

This paper presents the current version of the formal framework of vulnerability that is being developed in the ADAM Workpackage A1 at PIK. Based on a grammatical and semantical analysis of the everyday usage of the term, vulnerability is seen to be a relative concept, in the sense that accurate statements about vulnerability are possible only if one clearly specifies the *entity* that is vulnerable, the *stimulus* to which it is vulnerable and the *preference criteria* to evaluate the outcome of the interaction between the entity and the stimulus.

These three *primitives* are translated into formal language, using mathematical concepts. A *dynamical system* represents the entity, the *exogenous input* to the dynamical system plays the role of the stimulus, and the preference criteria are represented by a *partial strict order.* A dynamical system has two elements: first, a state space, that is, a set of possible ”states”, where each state fully describes a situation the entity can find itself in. The states contain all information relevant to the analysis, and depending on the case the type of the state space may differ: the system description can be given e.g. in deterministic, non-deterministic or stochastic form. The second element of the dynamical system is a transition function which describes the evolution of the entity: given the current state and the exogenous input, it tells what the next state is going to be. Thus, the system evolution depends on the current situation of the entity and on the input which represents the stimulus that the entity is confronted with. The preference criteria, capturing the notion of ”worse” inherent in the concept vulnerability, are represented by a partial strict order on the set of states. This means that different situations of the entity can be compared (the entity may be ”worse off” in some state than in some other state), however, the partiality of the strict order accounts for the fact that some situations may not be comparable by previously established criteria.

Given this mathematical translation of the primitives, complex concepts such as ”vulnerability” or ”hazard” are defined upon them, reproducing the structure of statements made in natural language. Two descriptions of how the formal framework does this are given in this paper: they differ in the amount and complexity of the mathematical notation used and in the point of view from which the topic is approached.

The first part (Chapter 2) starts out from statements as ”vulnerability consists in a potential worsening of the entity’s situation due to the interaction with a stimulus” and builds up formal definitions in several steps with increasing complexity both in the system description and in the statements or concepts to be formalised. For example, a simple entity that cannot act is defined to be ”simply vulnerable” to an input, if due to that input it transits to a state that is worse than its previous state was. Since there are cases in which this definition may be too simple, and in order to account for typical vulnerability statements in the climate change community, other ”variations” of vulnerability are defined and related to the first definition. Refinements for the simple system include comparison with the transition under a reference input (a ”baseline scenario”) instead of the previous state, comparison of different starting states or entities and an extension of the time horizon.

Next, a more complex entity is considered: two possible mathematical descriptions of an entity that can act are presented along with some advantages and disadvantages they bring about in describing the field of vulnerability. To capture the long-term interaction between the entity and its environment that is important in the climate change context, systems in co-evolution are introduced, where the environment is also modelled as a system. Finally, a way of considering multiple agents in the formal framework is displayed. In this process, further mathematical elements are added as needed.

The second part (Chapter 3), building on the experience from the prior framework development as described in the first part, starts out from the mathematical design. As it makes sense to include the stimulus in a more complex system description, the mathematical objects of study have reduced to two: the system and the preference criteria. The second part describes the system in a more general and more ordered manner with a special focus on the socio-ecological system. Then the idea of ”deterioration” of the system that emerges from the preferences is discussed. Vulnerability is defined as potential deterioration in a socio-ecological system, and the definitions from the first part are shown to be ”variations” on this general concept.

As the formal framework is also supposed to support computational work in vulnerability assessments, first tools that are developed alongside with the formal framework are presented (Chapter 4). The basic implementation of systems and of preference criteria is provided, implementations of the complex concepts will follow. The separation of the two basic elements (system evolution and preferences) allows for flexibility: one part can be exchanged without having to rebuild the whole model.

Chapter 1

Introduction

The notion of vulnerability is of interest in various scientific contexts such as climate change, natural hazards, food security and risk management in general. However, the conceptualisations of vulnerability differ greatly between different scientific disciplines and also within disciplines. In the field of climate change, a prominent definition of vulnerability is given in the glossary of the IPCC report (McCarthy et al., 2001):

> “**Vulnerability:** The degree to which a system is susceptible to, or unable to cope with, adverse effects of climate change, including climate variability and extremes. Vulnerability is a function of the character, magnitude, and rate of climate variation to which a system is exposed, its sensitivity, and its adaptive capacity.”

There are however various other definitions and conceptualisations of vulnerability in the climate change community, some of which are inconsistent with one another. The same holds for related concepts such as “sensitivity”, “adaptive capacity”, “resilience”, and many more. Brooks observes a ”sometimes bewildering array of terms” in the literature on vulnerability and adaptation and states that “[t]he relationships between these terms are often unclear, and the same term may have different meanings when used in different contexts and by different authors.” (Brooks, 2003, p. 2).

Conceptual work in this field goes back some twenty years (cf. for example the article by Kates (1985) on the Interaction of Climate and Society), and diverse approaches have been used to clarify concepts (Füssel and Klein (2006) for example analyse the “evolution of conceptual thinking” in climate change vulnerability assessments; O’Brien et al. (2006) discuss ”how two interpretations of vulnerability in the climate change literature are manifestations of different discourses and framings of the climate change problem”; Adger (2006) ”reviews research traditions of vulnerability to environmental change”; Gallopín (2006) uses ”a systemic perspective to identify and analyze the conceptual relations among vulnerability, resilience, and adaptive capacity within socio-ecological systems"; and several "frameworks" related to vulnerability have been proposed e.g. by Jones (2001), Brooks (2003), Turner II et al. (2003) and Luers (2005)).

Still, in the rather recent special issue of Global Environmental Change on "Resilience, vulnerability and adaptation" the editors "experienced a Tower of Babel in hearing the diverse definitions made of core concepts" (Janssen and Ostrom, 2006, p. 237) and state that "efforts should be made to develop clear (and hopefully, mutually compatible) specifications of the concepts for use in abstract and field studies of ecological and social systems." (Janssen and Ostrom, 2006, p. 238). Brooks states a prerequisite for solving the problem: researchers from different backgrounds "must develop a common language so that vulnerability and adaptation research can move forward in a way that integrates these different traditions in a coherent yet flexible fashion, allowing researchers to assess vulnerability and the potential for adaptation in a wide variety of different contexts, and in a manner that is transparent to their colleagues." (Brooks, 2003, p. 2)

The "formal framework of vulnerability to climate change" developed by Ionescu et al. (2006) proposes mathematics as the language of choice in that it defines vulnerability and related concepts in general mathematical terms. As Polya (2004) says: "Mathematical notation appears as a sort of language, *une langue bien faite*, a language well adapted to its purpose, concise and precise, with rules which, unlike the rules of ordinary grammar, suffer no exception." In not making exceptions, mathematical language can see to coherence and compatibility (as desired above) more easily than natural language, which may always contain residual ambiguities. Discussing the "desirability of formalisation in science", Suppes (1968) lists *explicitness, standardisation, generality* and *objectivity* among the advantages of formalising a scientific theory. Explicitness evidently addresses the wish for "clear specifications of the concepts" and that assessments be made "in a transparent manner" quoted above, while standardisation, generality and objectivity are a natural desiderata in a field with "a greater focus on the comparative analysis of case studies"(Janssen and Ostrom, 2006, p. 237).

Applied to vulnerability assessment, formalisation can help ensuring that analyses are carried out in a systematic and consistent fashion because the translation into formal language requires all assumptions to be made explicit. By providing the means to relate different interpretations of the same terms via a common translation into mathematics, clarity of communication can be improved. Finally, a formal model is a precondition for computational approaches to vulnerability assessment and can help take advantage of relevant methods in applied mathematics such as e.g. system theory.

This paper portrays the current state of the formal framework of vulnerability to climate change that is being developed by the FAVAIA-group[1] at the Potsdam Institute for Climate Impact Research (PIK) as part of the ADAM project.

It is based on the first version of the formal framework presented in (Ionescu et al., 2006) and is organised in the following way: Chapter 2 outlines the basic mathematical framework. The mathematics used has (hopefully) been kept at a level that patient non-mathematicians can understand. Chapter 3 spells out the mathematics used in more detail, and presents design and implementation of the formal framework in a more abstract (category theoretic) setting. Chapter 4 provides a software framework for the modelling of vulnerability problems.

Chapter 2

The basic formal framework

This chapter roughly follows the structure of (Ionescu et al., 2006) in presenting first a grammatical and semantical analysis of the term vulnerability and then the formalisation based on this. In the detailed description of the framework, changes in the order of presentation have been made so as to provide a clearer picture.

2.1 Grammatical and semantical analysis

"Towards a formal framework of vulnerability to climate change" (i.e. (Ionescu et al., 2006)) starts out with a grammatical and semantical analysis of the term *vulnerability* as used in everyday language, respectively as described in the entry "vulnerable" in the Oxford Dictionary of English (Soanes and Stevenson, 2005):

> exposed to the possibility of being attacked or harmed, either physically or emotionally: [. . .] | Small fish are vulnerable to predators.

From the grammatical investigation vulnerability is seen to be a relative concept: it is vulnerability *of* something *to* something, where the first "something" is any *entity* and the second one is the *stimulus* that the entity is confronted with. The semantical analysis further identifies a notion of negativity (conveyed by the terms "attacked or harmed") and of potentiality (expressed in "possibility of") as ingredients to vulnerability. The notion of "worse" corresponds to some *preference criteria*.[2] In conclusion one can sum up the structure of vulnerability as follows: vulnerability consists in a potential worsening of the entity's situation due to the interaction with a stimulus.

Having identified the *primitives* "entity", "stimulus" and "preference criteria" as building blocks of vulnerability statements, Ionescu et al. translate these into mathematical primitives. Within the mathematical framework complex concepts are then defined on the primitives yielding exact and explicit definitions, first of all of the concept of vulnerability in its everyday meaning.

Considering it "likely that the technical usage represents a refinement of the everyday one" Ionescu et al. (2006, p. 4) then refine the definition in several variations to account for different types of vulnerability statements.

However, before entering the mathematical framework some further considerations on vulnerability statements are made. Proceeding from simpler to more complex cases, two examples that occur throughout the paper are a system of small fish, vulnerable to predators (an entity that cannot take any action or learn) and a motorcyclist (an entity that can act) confronted with an oil spill on a winding mountain road. The motorcyclist example illustrates some challenges of formalising vulnerability: one has to account for

- comparisons: "We would normally say that a second motorcyclist who drives more slowly or more carefully is less vulnerable to the oil spill."

- different time scales: "The situation may be considerably more complex if we expand the time horizon. What about a third motorcyclist, who has heard about the oil spill on the road and has been able to prepare for it by buying new tires and improving his driving skills? Can his condition be meaningfully compared to that of the first two, who are confronted with an immediate hazard?"

- information and resources: "What about a fourth motorcyclist, who has been informed but has no money to buy new tires?" (Ionescu et al., 2006, p. 5)

Finally, vulnerability to climate change and in particular the IPCC definition quoted above together with the concepts *exposure, sensitivity* and *adaptive capacity* are illustrated by means of examples. A new aspect introduced in the context of vulnerability to climate change is for example "the ability of the vulnerable entity to act proactively to avoid future hazards (by mitigating climate change or by enhancing adaptive capacity)"(Ionescu et al., 2006, p. 5).

Having thus set the stage, Ionescu et al. tackle the mathematical part of the formalisation. The following figure displays a schematisation of the formalisation process.

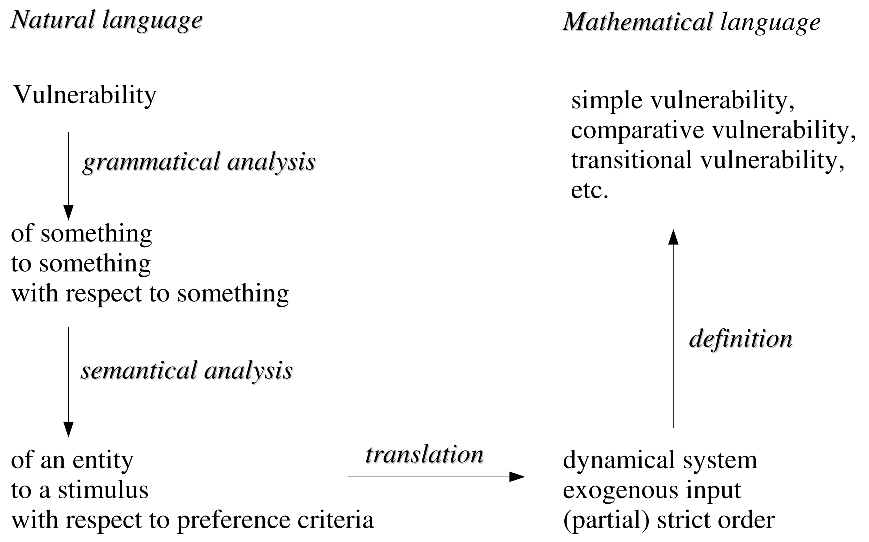

Figure 2.1: Graphical representation of the formalisation of vulnerability. The left column was explained above, the right column constitutes the argument of the following sections.

2.2 Translation: mathematical primitives

The basic mathematical framework of vulnerability translates the three primitives into mathematical ”primitives” on which complex concepts as vulnerability, adaptive capacity and so on can then be defined. Note that for simplicity the following two sections treat an entity that does not have the possibility to act, this feature will be introduced only in Section 2.5. The mathematical notation used in this chapter is rather basic, a table explaining the symbols used can be found at the end of this chapter on page 24.

The *entity* is represented as a discrete dynamical system consisting in a set of states X and a transition function f. The state is supposed to describe the current situation the entity finds itself in and therefore should contain all the information that is relevant to the analysis. This is the first point where the formal framework can help to produce clarity: all assumptions made about the entity should be made explicit here.

The entity's evolution is captured by the iterations of the system from one state to another. These iterations are described by the transition function f, which is applied to the current state of the system x (an element of X; $x \in X$) and an exogenous input e from the set of inputs E, which represents the current *stimulus*. In formulæ:

$$f : X \times E \to X, \tag{2.1}$$

Given the current state and stimulus, the transition function tells which element of X will be the next state of the system: $f(x, e) \in X$. This describes the simplest possible system, the discrete deterministic system; cases where the transition function instead of an element of X yields more complex states (e.g. sets of elements or probability distributions on elements) and the general description of more complex systems in category theoretical terms can be found in Sections 3.1 and 3.3.

Here, as in (Ionescu et al., 2006), the discrete deterministic system is used for simplicity of presentation. Often, the transition function will be impossible to determine in real world assessments. However, if one uses models, their behaviour will play the role of the transition function and even in non-computational assessments the formal concept of the transition function may be helpful in structuring research systematically e.g. by asking questions like which part of the evolution of the system one is interested in or what type of transition function would be adapted to the problem (deterministic, stochastic, . . .).

The *preference criteria* are represented as a partial strict order $\prec$ on the set of states X, where $x \prec y$ means that the system in state x is considered to be "worse off" compared to the system in state y (we will also say that x "is worse than" y). A partial strict order is an anti-reflexive and transitive binary relation, that is, it fulfills the two conditions that

- no element is worse than itself (anti-reflexivity) and
- if x is worse than y and y is worse than z, it follows that x is worse than z (transitivity, in formulæ: $(x \prec y) \wedge (y \prec z) \Rightarrow x \prec z$).

Partiality of the strict order means that for some pairs (x, y) neither $x \prec y$ nor $y \prec x$ may hold. In this case x and y are said to be *non-comparable.* A partial strict order is a very general mathematical model to represent preference criteria. Particularly, it is more general than "preference relations" as used in economics. How it relates to these will become clear below.

The possible partiality of the order accounts for the fact that in vulnerability assessment problems of "non-comparable" situations easily arise, which cannot be solved within a mathematical framework as e.g. value judgements may be necessary. If, for example, related to a disaster the question of how to compare situations that involve people affected or killed and damage done in terms of money arises, this question will have to be answered by agreement, not by mathematical criteria. Further aspects of the interpretation of the preference criteria used here shall become apparent from the definition of vulnerability.

2.3 Definitions of vulnerability

Proposing a variety of "definitions of vulnerability" may at first sight seem counterproductive when one aim of the formal framework is to produce relief in the confusion which partly stems from the fact that a multitude of definitions of the term are in use. However, it accounts precisely for the different structures of some rather generic vulnerability statements. To avoid confusion the different definitions are specified by qualifying adjectives. Further, it will be shown (for one example case) how one definition can be expressed as a special instance of another one, thus illustrating how the different definitions are variations of one general concept.

The list of definitions following here is not supposed to be comprehensive (and neither is that in (Ionescu et al., 2006)). Rather, users of the framework are invited to creatively add definitions that capture the structure of vulnerability statements suitable to their cases.

Based on the grammatical and semantical analysis a first simple statement to be formalised is that "the entity can be said to be vulnerable to a stimulus if due to that stimulus it makes a transition that leaves it "worse off" than it was before" (Ionescu et al., 2006). The corresponding definition is:

Definition 1 (simple vulnerability)**.** A system f in state x is vulnerable to an exogenous input e with respect to the partial strict order $\prec$ if $f(x, e) \prec x$.

Given this definition, a remark on preferences seems appropriate: the notion of worse is used to assert vulnerability, meaning that being worse off is interpreted as a noticeable deterioration (cf. Gallopín (2006): "a system would not be called vulnerable if the effect of the perturbation is limited to the generation of trivial and ephemeral changes"). The preference criteria should reflect this: preference criteria can be rather weak as long as they provide some distinction between 'good' and 'bad' states. Other than this they need not tell much about comparison between states, especially about "similar" states. In the general setting of a partial strict order, detailed comparisons of similar states are of course possible; they may, however, lead to the counterintuitive result that a system is vulnerable when a very slight worsening of the state is observed. An example here may be expressing preferences in terms of the money the entity owns, with the intuitive order that less money is worse. This preference criterion assesses any amount of money to be worse than this amount of money plus, say, one cent. However, saying that a system is vulnerable because of a loss of one cent will in most cases be a nonsensical conclusion.

Thus, mathematically speaking, preference criteria will probably be quite straightforward in that one would expect them to distinguish between a number of "bad states" that are to be avoided (e.g. not sustainable use of resources) and states that are seen as "acceptable" or "good". An example is to consider a threshold between functioning and not functioning of a system: a system would be considered vulnerable if there is the possibility that it transits to a "non functioning" state, or to "disaster". This does not mean, however, that preferences are actually a simple ingredient in vulnerability assessment, on the contrary, it may be very complicated or impossible to come up with a partial strict order to use in a certain case: trade-offs between different interests (or the interests of different people, groups, and so on) may be involved, and many preferences may be implicit. Eliciting preferences from stakeholders may constitute the major part of a vulnerability assessment in some cases.

Simple vulnerability captures most elements from the grammatical and semantical analysis (entity, stimulus, transition and worsening), however, it misses the small but important qualifier "potential" that was associated to the deterioration in the statement "vulnerability consists in a potential worsening of the entity's situation" in Section 2.1. In most cases, this will not be a problem, however, one can easily construct an example where the (as its name says) simple vulnerability definition can be seen to be too simple. The problem stems from the fact that in order to speak about a "potential" deterioration, the case of "non deterioration" must also be possible[3]. In the special case of a system that will be worse off no matter what transition it makes, this possibility is lacking. Consider the example of a terminally ill patient whose condition is deteriorating day by day. Simple vulnerability would assert him to be vulnerable to, say, sunshine, while in natural language one would say that he is not sensitive and not vulnerable to it.

To rule out that such a counterintuitive result is produced by the formal framework, we can establish that simple vulnerability may only be used when non-deterioration is possible. If this cannot be guaranteed, the problem can be solved by using a new definition of vulnerability, which needs as a further element a *reference input* $\tilde{e} \in E$.

Definition 2 (comparative vulnerability)**.** A system f in state x is vulnerable to $e \in E$ compared to $\tilde{e} \in E$ if $f(x, e) \prec f(x, \tilde{e})$.

This comparison guarantees that non-deterioration is possible, because there always is at least one input, namely $\tilde{e}$, that does not cause the system to be worse off, due to the anti-symmetry of the strict order $\prec$.

Fortunately, this reference input also has a natural interpretation in the field of vulnerability to climate change: often "baseline" or "reference" scenarios are used which play the role of this reference input, for example, when the evolution of a system with climate change is compared to the evolution without climate change.

As promised above, it shall now be shown how a more complex definition of vulnerability reduces to the simple one: the comparative vulnerability of a system can be computed as the simple vulnerability of a related system, as follows.

Assume that we have a programme which is capable of assessing simple vulnerability of systems, and assume that the reference input $\tilde{e}$ is given. The system we are going to use as an input to our program is going to have as states *pairs* of states of the initial system:

$$\begin{aligned} F : (X \times X) \ \times E \ &\rightarrow \ (X \times X) \\ F((x_1, x_2), e) \ &= \ (f(x_1, e), f(x_2, \tilde{e})) \end{aligned}$$

Consider the case in which both elements of the pair are identical: (x, x). The next state, as given by F, is going to consist of the pair $(f(x, e), f(x, \tilde{e}))$. The first element of the pair is the next state as given by the initial system f under input e, the second element of the pair is the next state as given by f under the reference input $\tilde{e}$:

$$F((x, x), e) \ = \ (f(x, e), f(x, \tilde{e}))$$

In order to use the definition of simple vulnerability for the system F, we have to have a strict order on the elements of $X \times X$, because they are the states of F. Let us denote this order by $\prec_F$: the subscript F will serve to distinguish it from the order on X used to compare the states of the initial system f. We are going to define $\prec_F$ by defining what a "bad" pair is, thus providing the above mentioned distinction between 'good' and 'bad' states. The set of all bad pairs will be denoted by B. The definition of the system F suggests at once that a pair is bad if the first element is worse than the second (that is, if the transition according to input e is worse than the one given by the reference scenario). We can therefore take the set B as being

$$B \ = \ \{(x, x') \mid x \prec x'\}$$

A pair of identical elements is never "bad", because the first element cannot be worse than the second if they are both identical. Moreover, we have that

$$F(x, x) \in B \ \equiv \ f(x, e) \prec f(x, \tilde{e})$$

The state $F(x, x)$ is "bad" only if the system f in state x is vulnerable to e compared to $\tilde{e}$. Thus, using the definition of $\prec_F$ resulting from B as defined above, we have that

$$F(x, x) \prec_F (x, x) \ \equiv \ f(x, e) \prec f(x, \tilde{e})$$

and thus, we have proven the following

Proposition 1. The system f in state x is vulnerable to e compared to $\tilde{e}$ iff the system F in state (x, x) is simply vulnerable to e with respect to $\prec_F$. □

Therefore, comparative vulnerability can be seen as a variation on simple vulnerability, one which can fit more naturally the cases in which we have a reference scenario. In what follows, we shall see some more variations on the theme of vulnerability. In all cases, the vulnerability statements can be formulated as simple vulnerability of a related system.[4]

With comparative vulnerability we have formalised a statement on vulnerability involving a comparison, as was desired in Section 2.1. Depending on the focus of research, the changing element in the comparison might not be the stimulus, but rather different initial situations of the entity, or different entities may have to be compared in their evolution when confronted with the same stimulus. The following variations on comparative vulnerability[5] formalise these cases considering a fixed stimulus e. For example, if one is trying to steer a system into the best starting position for some future transition, states-comparative vulnerability may be useful:

Definition 3 (states-comparative vulnerability)**.** A system f in state x is more vulnerable to the exogenous input e than in state x' if $f(x, e) \prec f(x', e)$.

Also, the comparison of different systems may be of interest:

Definition 4 (systems-comparative vulnerability)**.** A system f in state x is more vulnerable to an exogenous input e than a system f' in the same state if $f(x, e) \prec f'(x, e)$.

In a way, these definitions can be said to describe the ”dimensions” of possible comparisons, inputs, states or transition functions (or combinations of these) can change and be compared using the partial strict order on X.

Further refinements of vulnerability definitions can be made when more complex types of preference criteria are used. So far, we have mostly compared single states (whether to a previous state or to another state that was reached by a different transition), as the partial strict order was defined on states. Only in the special system that was needed to reduce comparative vulnerability to simple vulnerability has an order on pairs occured. Partial strict orders can also be defined on more complex elements, such as trajectories of states. An order on pairs for example allows to compare changes (from one state to another). In the field of vulnerability to climate change a preference criterion may for example be that changes in mean temperature are small. This preference cannot be expressed by just looking at two different possible states x_1 and x_2: one has to compute the changes that occurred from the previous state x to these two cases and therefore needs the information on what the previous mean temperature was. Such a preference has to be defined on pairs of type (x, x_1) and (x, x_2), it is still denoted by $\prec$, though. Since it is the transition that makes the difference, the corresponding definition takes the name of transitional vulnerability[6]:

Definition 5 (transitional vulnerability)**.** A system f in state x is vulnerable to $e \in E$ compared to $\tilde{e} \in E$ if the transition under e is worse than the one under $\tilde{e}$:

$$(x, f(x, e)) \prec (x, f(x, \tilde{e})). \tag{2.2}$$

2.4 Dynamical extensions

In the above definitions we have considered only one transition of the system, a natural and useful generalisation is to define vulnerability for a system with a longer evolution. Therefore, in this section we consider given a sequence of exogenous inputs $[e_1, e_2, \ldots, e_n]$ and compute the corresponding n transitions of the system as follows.

$$\begin{aligned} x_0 &= x \\ x_1 &= f(x_0, e_1) \\ x_2 &= f(x_1, e_2) \\ &\ldots \\ x_n &= f(x_{n-1}, e_n). \end{aligned} \tag{2.3}$$

The whole trajectory will be denoted by $[x, x_1, x_2, \ldots, x_n]$. Further, given a sequence of reference inputs $[\tilde{e}_0, \tilde{e}_1, \ldots, \tilde{e}_n]$, the corresponding reference evolution of the system is computed analogously according to (2.3) and denoted by $[\tilde{x}, \tilde{x}_1, \ldots, \tilde{x}_n]$. The following definitions generalise those from the one-step case:

Definition 6 (simple n-step vulnerability)**.** A system f in state x_0 is vulnerable to the sequence of exogenous inputs $[e_1, \ldots, e_n]$ with respect to $\prec$ if $x_n \prec x_0$.

Definition 7 (comparative n-step vulnerability)**.** A system f in state x is vulnerable to a sequence $[e_1, \ldots, e_n]$ compared to $[\tilde{e}_1, \ldots, \tilde{e}_n]$ if $x_n \prec \tilde{x}_n$.

Definition 8 (transitional n-step vulnerability)**.** A system f in state x is vulnerable to the sequence of exogenous inputs $[e_1, \ldots, e_n]$ compared to $[\tilde{e}_1, \ldots, \tilde{e}_n]$ if $[x, x_1, \ldots, x_n] \prec [x, \tilde{x}_1, \ldots, \tilde{x}_n]$.

As in the one-step case, the definition of transitional vulnerability requires that preference criteria be defined on a more complex type of element than just states: here, $\prec$ compares entire sequences of states. This may be more desirable than just considering end-points for the interpretation of "worse", as occurs in the first two definitions here, which thus ignore any worsening of the state before the end-point is reached.

We close this section with two remarks: strict orders on complex elements can be assumed given from the beginning (e.g. one might have elicited preferences over the evolution of some system for the next ten years from stakeholders). Otherwise, when provided with a strict order on states only, one may want to extend this to a strict order on trajectories. There are many possibilities to do this and one might want to satisfy some conditions that assure compatibility of the complex preferences with the original ones. This is a topic for further development of the framework.

The system that was considered up to now is too simple to describe systems that involve humans, because there is no representation of action. Two possible formalisations of action in the given context shall be introduced in the following. First, we describe the system with action as input, which is more intuitive, but leads to difficulties when trying to formalise concepts like coping and adaptation. Then, a sketch of the "action-function system" is given. There, some concepts such as coping, adaptation and mitigation can be formalised rather naturally, however, this part of the framework is still under construction.

2.5 Action as an input

As already hinted at, action can be represented by a further input to the system. This input will be called action input and denoted by a set U from which actions u can be chosen. The system thus takes the form:

$$f : X \times E \times U \to X \tag{2.4}$$

and therefore the next state $f(x, e, u)$ depends on the value of $u \in U$, i.e. on the action chosen. In most applications the set U will contain a "do nothing" action. The transition function f will, in general, be *partial*: not all actions are possible with every combination of state and stimulus. In this system the inputs e and u are simultaneous, which may be counterintuitive: often an action will be chosen only when a system is confronted with a certain stimulus e. Thus the stimulus is interpreted as arriving before the action is chosen, and which actions are available may depend on the stimulus. However, this dependence is accounted for by the partiality of f, in that for "incompatible" triples of state, stimulus and action, f is simply not defined. Advantages of this system are that it also works for the converse situation where the set of possible stimuli depends on the action that is being taken, and that it is the simplest system of this kind.

The introduction of action now allows to analyse the concepts of "hazard" and "potential impact". An "impact" could already have been defined for the system without action from Section 2.3:

Definition 9 (impact)**.** If a system f in state x is vulnerable to an input e, then the (deteriorated next) state $f(x, e)$ is called an impact.

Note that any previously defined type of vulnerability can be used in this definition. Depending on the type that is chosen, one could speak of a "simple impact", a "comparative impact" and so on. However, for a "hazard" and a "potential impact" to be defined, one must dispose of the notion of potentiality in the transition that the system makes due to the input. An input would not be called a hazard in natural language, if the deterioration caused by it was predetermined to take place. For the system that is able to act different possibilities of transition arise from different actions (supposing in the following that the set of actions contains at least two elements which lead to different next states), which means that a notion of potentiality is now given. This however also makes the definition of a hazard complicated: a stimulus may lead to a worsening of the situation if some action is taken and not do any harm under a different action. Stating that ”a hazard is, intuitively, an exogenous input that has the potential to make the situation of the system worse and thus cause potential impacts.” Ionescu et al. (2006, p. 14) solve the problem by defining a (non intuitive) auxiliary construction upon which they then base the definition of a hazard, as follows.

Definition 10 (relative hazard)**.** An exogenous input $e \in E$ is a relative hazard for a system f in state x relative to an action $u \in U$ if $f(x, e, u) \prec x$.

Definition 11 (hazard, potential impact)**.** An exogenous input $e \in E$ is a hazard for a system f in state x if $\exists\ u \in U : f(x, e, u) \prec x$. In this case, $f(x, e, u)$ is called a potential impact.

That is, a relative hazard is defined to be a stimulus that leads to a worsening when occuring in combination with a certain action, therefore it is a ”hazard relative to this action”. Then, a stimulus is defined to be a hazard if there is the possibility of choosing an action that makes this stimulus dangerous.

From these complications it is obvious that a hazard as well as a potential impact depend both on the stimulus and on the action the system takes since the next state of the system with action input (and thus of course a potential worsening of it) depends both on the stimulus and on the action. Although complicated, this reflects the real world usage of these terms: that there is an important interaction between stimulus and action chosen to construct a hazard can be seen in examples where the same stimulus affects some systems negatively while others are not affected due to their actions.

Let us remark that care is needed when dealing with these concepts: when looking at a fixed stimulus and two or more actions in the configuration that the stimulus does not do any harm if combined with some actions while it worsens the situation under other actions, one easily tends to see the hazard in the action and to forget about the stimulus. In this case one should keep in mind that the same holds for actions: an action may lead to a worsening of the situation when taken while a certain stimulus is present and not under another stimulus. It is really the interaction of stimulus and action which constructs the hazard.

This interaction other than possibly causing confusion can pose the interesting and highly political question of attribution. For example in questions of adaptation funding the question ”how much” of an impact or a hazard occurs ”due to” the stimulus climate change seems to become highly relevant. Far from solving this problem, hopefully the formal framework can help emphasising this structural aspect of it, and in special cases, such a formal analysis might lead to concrete results.

For further use, Ionescu et al. also define an *unavoidable hazard*:

Definition 12 (unavoidable hazard)**.** An exogenous input e is an unavoidable hazard for a system f in state x if $\forall\ u \in U : f(x, e, u) \prec x$.

Note that for simplicity these (and following) definitions are based on simple vulnerability, alternatively a "comparative hazard" could be defined employing comparative vulnerability and so on.[7]

For the system containing action one can now make statements about vulnerability in which different actions are compared, thus adding a "dimension", and probably the most important one in questions on adaptation to climate change, to the previously mentioned comparisons.

Definition 13 (action-comparative vulnerability)**.** A system f in state x is more vulnerable to the exogenous input e when taking the action u_1 than when taking the action u_2 if $f(x, e, u_1) \prec f(x, e, u_2)$.

In adaptation questions, the special case where an adaptation action u is compared to a "business as usual" action $\tilde{u}$ does not exactly resemble the situation with the baseline scenario in Definition 2: the order of the comparison will in most cases be the other way around, as one wants to assess whether under an adaptation measure the entity is less vulnerable than under business as usual, that means, the transition under the reference action would appear to the left of the $\prec$ sign. To use the above definition, one can turn around the question "is the system with adaptation less vulnerable?" into "is the system without adaptation more vulnerable?".

Ionescu et al. address the original question by defining an *effective action* as one that does not leave the system worse off:

Definition 14 (simple effective action)**.** An action u is effective for a system f in state x subjected to an exogenous input e if *not* $(f(x, e, u) \prec x)$.

This is the definition that corresponds to simple vulnerability, an *action-comparative effective action* is analogously defined as an action that makes the system not end up worse off than it would have been under the transition using the baseline action $\tilde{u}$:

Definition 15 (action-comparative effective action)**.** An action u is "action-comparison-effective" for a system f in state x subjected to an exogenous input e if *not* $(f(x, e, u) \prec f(x, e, \tilde{u}))$.

This concept may be more useful in the context of climate change, where it may not be possible to avoid all the impact of a stimulus, but an adaptation may make the system better off than it would have been without adapting.[8] However, in this formulation, the information may be weaker than desired: one cannot necessarily assert an actual improvement resulting from an action with this concept. An asserted non-deterioration might always be due to non-comparability of the two states, i.e. *not* $(f(x, e, u) \prec x)$ does not necessarily mean $(x \prec f(x, e, u))$.

Based on the notion of effective actions, Ionescu et al. propose preliminary definitions of adaptation and adaptive capacity, which will be refined in the ongoing development of the framework. In the following form they address coping rather than adapting, e.g. because the time horizon involved is of just one future time step:

- Adaptation is choosing an action $u \in U$ such that we have *not* $(f(x, e, u) \prec x)$, i.e. choosing an effective action.
- The adaptive capacity of a system f in state x subjected to an exogenous input e is the set of its effective actions.

Moreover, one might want to consider the quality of the actions available to the system, not just their number, and define adaptive capacity as a measure of this quality. In order to do this, actions are required to have "qualities" that can be measured and compared.

Further specifications could be considered in the development of the formal framework of vulnerability. For example the idea of differentiating potential from actual adaptive capacity that has arisen in recent research would be an interesting point to formalise. The definition of adaptive capacity as a set seems to fit the notion of potential adaptive capacity quite well: the set lists all possible effective actions, and thus represents a potential capacity of the entity. For actual adaptive capacity, that should e.g. reflect the likelihood of successfully exploiting adaptation options, some information on which action will be chosen from a set of possibilities will be necessary. The question of the choice of an action is important in the following dynamical extensions of the action-input system.

As before, we consider a system that makes more than one iteration. The system with action input needs a starting state x, a sequence of exogenous inputs $[e_1, e_2, \ldots, e_n]$ and a sequence of action inputs $[u_1, u_2, \ldots, u_n]$ in order to compute n transitions $x_1 = f(x, e_1, u_1), x_2 = f(x_1, e_2, u_2)$ and so on. While before we assumed that a sequence of exogenous inputs was given, it does not seem appropriate to assume the same for the sequence of actions, since it should be the entity that chooses the action at each time step, depending on the state and stimulus it is confronted with. To model this choice, we can assume that the entity uses a "policy function" $\phi : X \times E \to U$ to determine an action from a current state and stimulus.[9] Finding out this policy function could be the subject of decision making research. Before going into detail on what questions could be addressed with regard to the policy function, it makes more sense to first introduce the slightly more complex example where the exogenous input is not just given but generated by a system of its own that interacts with our previous system[10] in the following section.

On the other hand, without assuming a mechanism that chooses an action at each step, one can also consider the non-deterministic system that results from playing through all the different alternatives. Seeing the effects of different actions at different time steps by simulations may be a worthy undertaking to support decision making.

2.6 Co-evolution of system and environment

The above assumption that the sequence of exogenous inputs is given does not reflect any interaction between the system and the inputs. In order to account for this interaction, the *environment* that provides the exogenous inputs to the system is itself modelled as a dynamical system, $h : X \times E \times U \to E$, so that the next input from the environment $h(x, e, u)$ depends on the state of the system and on the action chosen by the entity.

Given an initial state of the system, x_0, an initial input from the environment, e_0 and a policy ϕ the system iterates as follows:

$$\begin{aligned}
&u_0 = \phi(x_0, e_0), \quad x_1 = f(x_0, e_0, u_0), \quad e_1 = h(x_0, e_0, u_0)\\
&u_1 = \phi(x_1, e_1), \quad x_2 = f(x_1, e_1, u_1), \quad e_2 = h(x_1, e_1, u_1)\\
&\dots\\
&u_{n-1} = \phi(x_{n-1}, e_{n-1}), \quad x_n = f(x_{n-1}, e_{n-1}, u_{n-1}), \quad e_n = h(x_{n-1}, e_{n-1}, u_{n-1}).
\end{aligned} \tag{2.5}$$

As the policy function is the ingredient to this iteration that an entity has in hand, many problems will concern finding a policy function that makes the system iterate in the "desired" way, where "desired" is to be defined on a case by case basis. Ionescu et al. formulate some general goals one might want to achieve by the choice of a policy:

- **Optimisation:** Choose a policy ϕ such that the actions taken drive the system along an optimal trajectory.

- **Mitigation**: Choose a policy ϕ such that for all $k \in \{1, ..., n\}$ no $e_{k+1} = h(x_k, e_k, u_k)$ is an unavoidable hazard.

- **Maintaining adaptive capacity**: Choose a policy ϕ such that for all $k \in \{1, ..., n\}$ there exists at least one effective u_k.

In this formulation, mitigation and maintaining adaptive capacity are mathematically the same thing, since a hazard e is unavoidable if and only if there are no effective actions against e. This problem is addressed by the action-function system below (cf. Section 2.8).

2.7 Multiple agents

For completeness of the overview over the basic formal framework presented in (Ionescu et al., 2006), let us briefly refer to the treatment of multiple agents. Ionescu and colleagues propose two possible ways of dealing with interacting systems.

For simplicity they consider two systems

$$\begin{aligned} &f_1 : X_1 \times E \times X_2 \times U_1 \to X_1, \\ &f_2 : X_2 \times E \times X_1 \times U_2 \to X_2 \end{aligned} \tag{2.6}$$

which interact with the environment and with each other:

$$\begin{aligned} x_{1,k+1} &= f_1(x_{1,k}, e_k, x_{2,k}, u_{1,k}), \\ x_{2,k+1} &= f_2(x_{2,k}, e_k, x_{1,k}, u_{2,k}). \end{aligned} \tag{2.7}$$

(Partial) strict orders $\prec^1$ and $\prec^2$ on X_1 and X_2 respectively are assumed to be given. A vulnerability assessment can be carried out for the combined system or for each system independently. In the first case one needs to define preference criteria for the combined system.

$$f_{1,2} : X_{1,2} \times E \times U_{1,2} \to X_{1,2}, \tag{2.8}$$

where

$$\begin{aligned} X_{1,2} &= X_1 \times X_2, \\ U_{1,2} &= U_1 \times U_2, \\ f_{1,2}((x_1, x_2), e, (u_1, u_2)) &= (f_1(x_1, e, x_2, u_1), f_2(x_2, e, x_1, u_2)). \end{aligned} \tag{2.9}$$

This means choosing a (partial) strict order on the set $X_{1,2}$, which would combine the two (partial) strict orders, $\prec^1$ and $\prec^2$. As there is no unique way of doing this - and in the special case of total strict orders there even exists a theorem, the "Impossibility Theorem" by Kenneth Arrow, which states that under some rationality conditions, no rule to aggregate any given set of preferences can exist (cf. for example (Sen, 1979)) - combining preference criteria will have to be done on a case by case basis. Tools that are being developed with the formal framework may help to check consistency with the given preferences.

In the second case, in order to assess the vulnerability of each system independently, the other system can be viewed as contributing to the exogenous input. Taking the case of the first system, one would simply consider the environment as including the second system:

$$f_1' : X_1 \times E' \times U_1 \to X_1, \tag{2.10}$$

where

$$\begin{aligned} E' &= E \times X_2, \\ x_{1,k+1} &= f_1'(x_{1,k}, (e_k, x_{2,k}), u_{1,k}). \end{aligned} \tag{2.11}$$

Then the problems of optimisation, mitigation and maintaining adaptive capacity can be addressed with respect to the extended environment. Multi-scale analysis becomes important in this case, because the environment will contain a part f_2, which operates at the same scale as the system f_1, and another part, given by the evolution of e, which typically takes place at a much slower pace.

Having encountered several problems in the formalisation that represents actions as input to the system, a different approach to the system with action will be sketched in the following section.

2.8 Action as a function

A second possibility to formalise action is to consider action as a function that, composed with (that is, applied after) the transition function of the system without action, can change the transitions of the system. This part of the formal framework is still under construction and is based on some rather mathematical considerations that are presented in Chapter 3. Therefore this section just presents some ideas on how this different representation of action might become useful in describing concepts like coping, adaptation and mitigation.

Let us return to the system without action input and consider it in co-evolution with the environment, that is, we have a set X of states of the system, a set E of inputs or states of the environment and two transition functions $f : X \times E \to X$, of the system, and $h : X \times E \to E$, the transition function of the environment. Given an initial state x_0 and an initial input e_0

these systems interact as follows:

$$\begin{aligned}
&x_1 = f(x_0, e_0), \quad e_1 = h(x_0, e_0)\\
&\dots\\
&x_n = f(x_{n-1}, e_{n-1}), \quad e_n = h(x_{n-1}, e_{n-1}).
\end{aligned}$$

This can be denoted more concisely in the form of the combined system (already seen in the multiple agents case) which will be named F:

$$\begin{aligned}
&F : X \times E \to X \times E, \text{ where}\\
&F(x, e) = (f(x, e), h(x, e)), \text{ also written as}\\
&F = \langle f, h \rangle \qquad (\text{"}f \text{ paired with } h\text{"})
\end{aligned}$$

Action is now represented by a function $g : X \times E \to X \times E$ which is composed with F to constitute the transition function of the system with action $g \circ F$. If we choose the identity function for g, i.e. $g(x, e) = (x, e)$, this represents the "business as usual action"; the transition then consists in the change that occurs due to F alone.

g can also be described as a pair of functions

$$\begin{aligned}
g =& \langle g_1, g_2 \rangle \text{ with}\\
&g_1 : X \times E \to X \text{ and } g_2 : X \times E \to E
\end{aligned}$$

The preference criteria remain the same as before: assume a (partial) strict order on X is given. Vulnerability is defined in terms of the combined system in Section 3.5 (a slightly different notation is used).

The two components of g have rather natural interpretations in the context of climate change: g_1 acts on the state space X and leaves the environment E untouched. g_1 can therefore be interpreted as "adaptive" component of g, while g_2, leaving the state X untouched and modifying only the environment E is the "mitigative" component of g.

It now turns out that since the preferences are defined only on X, for one-step problems only the first component of g is relevant: in the case where

$$f(x, e) \prec x,$$

the system would be simply vulnerable, and one would thus try act in a way that

$$f(x, e) \prec x, \text{ but } g_1(f(x, e), h(x, e)) \not\prec x$$

Any action taken on the environment would at the earliest influence the situation of the entity after one more time step, which seems to fit the idea of inertia in mitigation quite nicely.

More importantly, by modifying the type of the action function, one can now capture a difference in "coping-action functions" and "adaptation-action functions".

Let us introduce as a separate component to the system a set Y of states of the action function, that is $g : (X \times E) \times Y \to (X \times E) \times Y$. This state is not manipulated by $F : X \times E \to X \times E$, and can thus be interpreted e.g. as containing information about action. The whole system will then make transitions from $(X \times E) \times Y$ to $(X \times E) \times Y$ and therefore one has to "extend" F to formally "do nothing" to Y:

$$\begin{aligned}
&F_{ext} : (X \times E) \times Y \to (X \times E) \times Y \\
&F_{ext} = F \times id_Y \text{ ("} F \text{ cross } id_Y\text{") which means that} \\
&F_{ext}((x, e), y) = (F(x, e), id_Y(y)) = (f(x, e), h(x, e), y).
\end{aligned}$$

System iteration is given by the transition function $g \circ F_{ext}$, where g has three components of the following form:

$$g = \langle\langle g_1, g_2\rangle, g_3\rangle, \text{ where}$$

$$g_1 : (X_1 \times X_2) \times Y \to X_1 \tag{2.12}$$

$$g_2 : (X_1 \times X_2) \times Y \to X_2 \tag{2.13}$$

$$g_3 : (X_1 \times X_2) \times Y \to Y$$

This system description can capture a distinction between coping and adaptation actions: coping corresponds to the first system described in this section (which can be described as a special of the second system) because a "stateless" g only acts upon the current state and stimulus, fitting the notion of ad-hoc-action involved in the term "coping". Using the "stateful" function g from the second system to describe adaptation action, information (such as past experience, future predictions, plans, ...) may be stored in the state Y and is not wiped out or altered by application of F.

Let us close the chapter on the basic formal framework of vulnerability to climate change with some remarks on ...

2.9 Scope and usefulness of the framework, outlook

The considerations in the previous sections have pointed out that there is not one single answer to the question "what is the right definition of vulnerability?" Rather, the definition of vulnerability that might be most useful is to be decided on a case by case basis. Having this formal framework at hand will hopefully make the decision easier in that it points out the structure of the different variations. So, while of course far from providing an answer to the questions asked in a vulnerability assessment, the framework might be helpful in working out what exactly the questions are in the assessment.

As a final example to this regard let us take a look at the following schematized situation: for analysing the effectiveness of an adaptation measure one could ask:

- whether the system is vulnerable without implementing the adaptation (simple vulnerability for the system using a "do nothing" action)
- whether the adaptation has any effect (action-comparative vulnerability: is the system without adaptation vulnerable compared to the system with the adaptation?)
- whether the adaptation is sufficient (simple vulnerability for the system using the adaptation as the action)

In particular, while the system with the adaptation might be "better off" than without it, it might still be in an undesirable state, which means there would be the need for a better adaptation.

It has thus become clear that the formal framework of vulnerability presented here is not to be confused with a "framework for assessing vulnerability" which could be a set of guidelines or lists of questions to be applied in a concrete assessment. This formal framework has been developed to translate (still strongly simplified) vulnerability statements into mathematics to better understand their structure by making it explicit. The common misunderstanding that the use of mathematics requires quantitative input should thus have been dispelled here. Neither does the formalisation have to lead to quantitative results: actually, one does not even have to go all the way to the mathematical formulation in the formalisation process. Making explicit what the entity under consideration, the stimulus (or stimuli) and the preferences are may already be useful to produce clarity. Especially in the case where several entities, each with a different set of preferences, interact, misunderstandings that result e.g. from an unnoticed change in perspective (from the combined system to one subsystem or vice versa) can be avoided if at each stage the point of focus is made explicit. A more systematic assessment is thus supported.

Hopefully, the formal framework will not be perceived as being prescriptive, and as trying to throw sand in the eyes of those unfamiliar with mathematical notation. Rather, we wish that researchers make creative use of it where it can be serviceable to their (case) studies.

In the case where computational components (such as models) are to be used in a vulnerability assessment, the interface between the highly complicated real world and the dull precision of the computer will anyway have to be a mathematical model. Here, the formal framework and the computational tools implementing it can provide the basic architecture to be filled with different content from case to case. The next chapter illustrates how the "vulnerability engineer" can make use of the formal framework.

There, the strict separation of the system evolution and the preference criteria proves to be an advantage of the framework: the decomposition into parts allows for flexibility in assessments. For example, updating preferences according to results from stakeholder dialogues can be done without having to change the part of the work that concerns the system evolution.

Ionescu et al. further have a section of preliminary applications in their paper that relates the formal framework to the IPCC conceptualisation of vulnerability and to two vulnerability assessments, ATEAM and DINAS-COAST. The findings of this section include that "the three determinants of vulnerability as identified by the IPCC correspond only in part with the three primitives of our formal framework"(Ionescu et al., 2006, p. 26). A formalisation of adaptive capacity that better fits the usage in the climate change community is one of the points where the basic framework shows scope for refinement. Also, other concepts, e.g. sensitivity, will have to be integrated into the framework.

A (rather abstract and mathematical) generalisation to more complex systems has been made; a way that makes it possible to deal with e.g. non-determinism, stochasticity and fuzziness is presented in the following chapter. Other refinements, such as continuous time, multiple scales and many more are the goals of further development of the framework.

As Ionescu et al. (2006, p. 27) state"[t]he analytical framework must be informed by the large body of results available from past case studies and by the needs of ongoing vulnerability assessments and the users of their results." To see to this, the development of the formal framework closely interacts with the analytical review and meta-analysis of impacts and vulnerability that is undertaken also in Workpackage A1 of the ADAM project. On the one hand it provides a general, standardised and objective (to speak with Suppes, cf. page 2) conceptual basis that can serve as a guideline for the meta-analysis. On the other hand, the meta-analysis can test the formal framework and strengthen its further development by grounding the framework in the synthesis results from case studies.

Symbol	Meaning
$x \in X$	x is an element of the set X
$f : X \to Y$	f is a function defined on the set X and taking values into the set Y
$X \times Y$	The set of ordered pairs having as first element an element of X and as second element an element of Y these pairs are written (x, y)
$f(x) = y$	The value of f for the element x is y
$x \prec y$	x is worse than y
$x \wedge y$	Logical and, *i.e.*, x is true and y is true
$x \Rightarrow y$	from x follows y
$B = \{x \mid \ldots\}$	B is the set of all x such that $\ldots$ holds
$x \equiv y$	x is equivalent to y (x is true if and only if y is true)
iff	if and only if
$\exists\ x : ...$	There exists at least one x such that ...
$\forall\ x : ...$	For all x such that ...
$\langle f, g \rangle$	f paired with g; for functions of the types $f : X \times Y \to X$ and $g : X \times Y \to Y$ $\langle f, g \rangle(x, y) = (f(x, y), g(x, y))$
$x \not\prec y$	x is not worse than y
$f \times g$	f cross g; for functions of the types $f : X \to X$ and $g : Y \to Y$ $(f \times g)(x, y) = (f(x), g(y))$

Table 2.1: Mathematical symbols and their meanings.

Chapter 3

The formal framework: technical details

Suppose a vulnerability assessment has been commissioned. According to our analysis, the first task that faces the "vulnerability engineer" is to identify the system representing the (potentially) vulnerable entity. This identification will have as a result a model of the entity. Since our aim is to design components for computational vulnerability assessment, we concentrate here on models which are expressed as executable programs. The identification of the entity includes that of the sources of the potential hazards, and their systemic, computational modeling. It will often be the case that these different systems have different types, some deterministic, some stochastic, etc., and act on different time scales. Moreover, the systems themselves will generally be composed out of subsystems. A framework of vulnerability must allow the vulnerability engineer to put together complex systems from simpler ones, to integrate different types of systems and to take advantage of existing models.

The next task is to find out, in dialogue with the stakeholders, the preference relation with respect to which vulnerability is to be judged. In our experience, the stakeholders will wish to express different preferences, and be able to assess the vulnerability with respect to each of them. Any architecture for vulnerability assessment must take into account the changing nature of the preference relations, and must allow the vulnerability engineer to quickly substitute one for another, to test that they satisfy certain consistency conditions, and to suggest possible alternatives.

If the entity under consideration is found to be vulnerable, the vulnerability engineer must analyse, together with the stakeholders, the possible alternatives for adaptation and mitigation. This will in general involve testing alternative components of the system which represent the results of different policy choices. Again, a framework of vulnerability must provide ways to replace some components of a system with others of a possibly different type and to test the consistency of the result.

The results of the investigation undertaken in the previous chapter provide a starting point for the design and implementation of such a framework. We start by formulating the general definition of a system which we will use in the sequel. In order to define vulnerability, we first clarify the more general notion of *deterioration*. We then define vulnerability as arising from potential deterioration in the context of a *socio-ecological system*. Finally, we revisit the definitions from the previous chapter and show they are special cases of the one given here.

The definitions given here are operational: they are, in fact, Haskell function definitions, and this chapter is a literate Haskell program, and therefore serves, at the same time, as an implementation of the components for vulnerability assessment derived from the theoretical framework for vulnerability which we develop.

3.1 Dynamical systems

The usual way of defining a system in the fields of Engineering is in terms of the actions of a monoid on a set. A standard example is offered by Denker (2005, p. 4) (our translation):

Definition 1 (Denker)**.** Let T be a semigroup with unit element e and X a non-empty set. The tuple (X, T) is called a *dynamical system* if there exists an associative map

$$\begin{array}{lcl} X \times T & \longrightarrow & X \\ (x, t) & \longrightarrow & tx \end{array}$$

for which the unit e acts as identity, that is, when the following two conditions hold:

$$(t_1 x, t_2) \longrightarrow t_2(t_1 x) = (t_2 t_1)x \text{ and } ex = x$$

The set T is interpreted as *time*, typical values for it are $\mathbb{N}$ for discrete dynamical systems, and $\mathbb{R}$, for continuous ones. The set X is called *the state space*, an element $x \in X$ is called a state of the system. Different types of system are distinguished by the existence of certain structures on X. Thus, in order to define linear systems, X must be a vector space, for stochastic systems we must have a probability space on X, for non-deterministic systems X must be a powerset, and so on.

In the past decade, a different definition for “dynamical system” has emerged within the Computing Science community:

Definition 2 (Rutten, Jacobs)**.** Let **C** be a category (usually **SET**) and $F : \mathbf{C} \longrightarrow \mathbf{C}$ be an endo-functor. A (general) *dynamical system* is a co-algebra of F, i. e. a function of type

$$f \ : \ X \longrightarrow FX$$

for some set X.

In this framework, different types of systems are distinguished by the different functors F and the underlying category. If F is P, the powerset functor, we have the type of non-deterministic systems, if the underlying category **C** is **Vect**, we have for $F = Id$ a linear deterministic system, and for $F = P$ a linear non-deterministic one, and so on.

This definition of a system seems to describe better the cases we encounter in practice. For example, in this framework, a stochastic system is described by a conditional probability distribution, which is indeed always the case when one is talking about "stochastic systems". Of course, the *iteration* of such a system requires the definition of a way of "applying" a function defined on X to a probability distribution over X, but this is just what the conditional probability formula gives us. We would not refer to a function $\phi \ : \ \text{Prob } X \longrightarrow \text{Prob } X$ as a stochastic system if it could not be described as the application of a conditional probability.

The co-algebraic point of view seems very well suited for discrete systems. The iterations of such systems are expressed in terms of morphisms to the final co-algebra, resulting in potentially infinite lists of succesive states (see (Rutten, 2000)). It is, however, not immediately obvious how to describe continuous-time dynamical systems in this framework.

The solution we adopt here is to generalise the standard definition of a dynamical system in order to take into account the co-algebraic point of view. In order to do this, we first note that the standard definition is equivalent to the somewhat simpler following form:

Definition 3 (equivalent to 1)**.** Let $(T, +, 0)$ be a monoid and X an arbitrary set. A dynamical system is a morphism from $(T, +, 0)$ to $([X \to X], \cdot, \text{id})$, i.e. a function

$$\phi \ : \ T \longrightarrow [X \to X]$$

such that

$$\begin{array}{lcl} \phi \ 0 & = & \text{id} \\ \phi \ (t_1 + t_2) & = & \phi \ t_1 \cdot \phi \ t_2 \end{array}$$

If $T = \mathbb{N}$, then, denoting $f = \phi \ 1$, we have

$$\phi\ n = f^n$$

The function f gives the iterations of the discrete dynamical system ϕ. In such a situation, Denker writes (X, f) for the system instead of (X, T).

If we take the co-algebraic perspective, f is a deterministic dynamical system, being a coalgebra of the functor Id. For other functors, such as Prob or P, the discrete system would take the form $f\ :\ X \to FX$ and the iterations would use a different operator instead of composition, but the structure would remain "the same". It is this similarity which is captured, and in a certain sense justifies, the definition which we adopt for "dynamical system".

Definition 4 (Dynamical System)**.** Let $(T, +, 0)$ be a monoid, $\mathbf{C}$ a category and $F\ :\ \mathbf{C} \longrightarrow \mathbf{C}$ an endo-functor, X an object of $\mathbf{C}$, $(\text{Hom}\ (X, FX), \diamond, \text{unit})$ a monoid. A *dynamical system* is a monoid morphism

$$\phi\ :\ T \longrightarrow \text{Hom}\ (X, FX)$$

Remarks.

1. ϕ is an arrow in **SET**, even when $\mathbf{C}$ is a different category. That is why we have written $\text{Hom}(X, FX)$ instead of the usual $[X \to FX]$. Of course, when $\mathbf{C} = \mathbf{SET}$, ϕ is a curried function of two arguments:

$$\phi\ :\ T \longrightarrow X \longrightarrow FX$$

2. If $T = \mathbb{N}$, then we have, as in the standard case

$$\begin{array}{lcl} \phi\ n & = & f^n \\ \text{where} & & \\ f & = & \phi\ 1 \\ f^0 & = & \text{unit} \\ f^{n+1} & = & f^n \diamond f \end{array}$$

   Then f is called the *transition function* of the system ϕ. A frequent abuse of terminology is to refer to both ϕ and f as "the system".

3. If F is a monad, then $\text{Hom}(X, FX)$ is a monoid with respect to the Kleisli composition. All the types of systems we consider, deterministic, non-deterministic, stochastic and fuzzy, are defined in terms of monadic functors F.

A point of terminology arises from the fact that, for $F \neq Id$, it is not clear what the state space should be: X, or FX? In particular, when looking at stochastic systems represented by Markov chains, one finds the term "state" used sometimes to denote the elements of X, sometimes the probability distributions over these elements. In order to avoid this confusion, we chose to call the elements of X *simple states* and the elements of FX *complex states*.

In the following, we shall only consider the case where F is a monad over **SET**, with $\diamond$ and unit being defined as in Kleisli composition. If $\phi \; : \; T \longrightarrow X \longrightarrow MX$ is a monadic system, then

$$\phi \; t \; x$$

is the complex state reached after t starting from the simple state x, and

$$mx \rhd (\phi \; t)$$

is the complex state reached after t starting from the complex state mx.

With these considerations, the Haskell definition of a dynamical system becomes:

$$\textbf{newtype} \; (\mathit{Monoid} \; t, \mathit{Monad} \; m) \Rightarrow \mathit{DynSys} \; t \; m \; x = \mathit{DynSys} \; (t \to x \to m \; x)$$
$$\mathit{simpleApply} \; (\mathit{DynSys} \; \phi) \; t \; x = \phi \; t \; x$$
$$\mathit{complexApply} \; (\mathit{DynSys} \; \phi) \; t \; mx = mx \rhd (\phi \; t)$$

3.2 Deterioration

All the definitions of vulnerability given in the previous chapter have in common the idea that a state of a system may get worse after a certain evolution of the system. This evolution has been described as being influenced by exogenous inputs, to which the vulnerability can be ascribed. We shall try here to represent a simpler, more general notion than *vulnerability*, namely that of (potential) *deterioration*.

Consider a monadic system $\phi :: \mathit{DynSys} \; t \; m \; x$ and a complex state $mx :: m \; x$. Statements of deterioration of the state mx must be made relative to a partial strict order on $m \; x$ which represents some notion of "worse than". Deterioration, just like vulnerability, is a relative notion.

The simplest way to express partial strict orders in Haskell is as boolean valued functions:

$$\textbf{type} \; \mathit{StrictOrd} \; a = a \to a \to \mathit{Bool}$$

Assuming that we have such a partial strict order on the set of complex states of ϕ

$$\mathit{worse} :: \mathit{StrictOrd} \; (m \; x)$$

we can ascertain whether the state mx suffers a deterioration after a given time t by computing the value of

$$(\mathit{complexApply}\ \phi\ t\ \mathit{mx})\ `\mathit{worse}`\ \mathit{mx}$$

(we have used the Haskell idiom of writing curried binary functions in infix notation by putting them in inverted ticks). If the result is *True*, then the state reached after the time t is worse than the initial state: therefore the state has deteriorated. If the result is *False*, the resulting state is either not comparable to the initial one, or is at least as good, therefore we cannot say that a deterioration has occurred.

We can summarise these considerations in an operational way as a Haskell function:

Definition 5 (Deterioration)**.** Given a system ϕ, a strict order *worse*, a complex state mx, we have that the state mx suffers a (potential) *deterioration* after time t if the predicate *deterioration* ϕ *worse* t mx is true, where

$$\begin{aligned}
&\mathit{deterioration} :: \mathit{DynSys}\ t\ m\ x \to \\
&\quad \mathit{StrictOrd}\ (m\ x) \to t \to m\ x \to \mathit{Bool}
\end{aligned}$$

$$\begin{aligned}
&\mathit{deterioration}\ \phi\ \mathit{worse}\ t\ \mathit{mx} = \\
&\quad (\mathit{complexApply}\ \phi\ t\ \mathit{mx})\ `\mathit{worse}`\ \mathit{mx}
\end{aligned}$$

This definition is not just operational, but also very general. It applies to systems which are deterministic, non-deterministic, stochastic, etc., with dynamics which can be continuous or discrete. This is a generality that we would like to bring to the formalisation of vulnerability started in the previous chapter. Accordingly, in the next section we begin investigating the connections between deterioration and vulnerability.

3.3 Deterioration for systems with input

The investigation of vulnerability done in the previous chapter has started from the assumption that vulnerability is relative to an exogenous input. Accordingly, all the definitions considered for various types of vulnerability have then been given in the context of a deterministic system with input. It seems natural that if we want to use the operational definition of “deterioration” given above, we can start by first asking ourselves if a system with input is, in fact, a system according to our definition 4.

Let us first review the form of the transition function for the deterministic system with input used in the previous chapter:

$$f : S \ \times \ E \rightarrow S$$

If this transition function were to fit our definition of a system, it should, however, fit the form

$$f : S \rightarrow F \ S$$

with Hom $(S, F \ S)$ a monoid with respect to a suitable operation.

Our transition function does fit this form if we curry it:

$$f : S \rightarrow (E \rightarrow S)$$

because the functor $F \ S = E \rightarrow S$ is a monad. Thus the simple deterministic system with input seems to fit our definition of a dynamic system, being, in fact, a monadic system.

The reason for the cautious qualifier "seems" is due to the fact that we have not checked that the transitions given by the monadic bind operator for the $(E \rightarrow)$ monad are the ones we take for granted in the case of the deterministic system with input. The latter are, intuitively speaking, of the following form: starting with a given $s_0 \in S$, we apply f and obtain a function $E \longrightarrow S$. We then choose an element $e_0 \in E$, and use it to obtain a new element $s_1 \in S$. With this new element we start again, applying f and choosing an $e_1 \in E$ with which we obtain $s_2 \in S$, and so on.

If we compute two transitions of the monadic system given by f we have

$$\begin{aligned}
& f^2 \ s_0 \\
= & \{ \textit{definition transition function} \} \\
& (f \diamond f) \ s_0 \\
= & \{ \textit{definition} \diamond \textit{for the } (E \rightarrow) \textit{ monad} \} \\
& h \\
& \textbf{where} \\
& h \ e = f \ (f \ s_0 \ e) \ e
\end{aligned}$$

The result is a function h which produces a value in S for every element $e \in E$. We have no way of using, as in our intuitive description, two elements of E, one for each transition. In the monadic system, once we pick an element as input for the first transition, the same element will be used for all subsequent transitions as well.

As a way out of this quandary, we can try to change the type of the input given to the transition function, so that one element is "enough". After all, in the definitions in the previous chapter we have often assumed that a number of inputs, $e_1, e_2, \ldots, e_n \in E$ were given, so why not modify the transition function so that instead of just elements of E it takes lists of values of type E, or, perhaps more general, entire trajectories of the input, that is, functions of type $\mathbb{N} \longrightarrow E$. The type of the transition function, which we denote by g, would thus have the form:

$$g : S \to (\mathbb{N} \to E) \to S$$

If we now compute two transitions made with g under a trajectory $es : \mathbb{N} \to E$, we obtain

$$g\ (g\ s_0\ es)\ es$$

and we would like to have

$$g\ s_0\ es = f\ s_0\ (es\ 0) = s_1$$

and

$$g\ s_1\ es = f\ s_1\ (es\ 1)$$

This is, in general, not possible: the information about which element of $\mathbb{N}$ to use in selecting an input from the trajectory is simply not available to g.

Once again we find that, if we want to make the monadic transitions agree with the intuitive description we have, we must change the transition function of the system. This time, the change involves the state space: S must be extended to contain information about the time at which the state is observed, in order to enable us to select the appropriate element of the input trajectory. We denote the new transition function with h, and we have

$$\begin{aligned}&h : (S\ \times\ \mathbb{N}) \to (\mathbb{N} \to E) \to (S\ \times\ \mathbb{N})\\ &h\ (s, n)\ es = (f\ s\ (es\ n), n + 1)\end{aligned}$$

The transition given by h computes, given an $s \in S$ at time n, the next state at time $n + 1$. We can now start with $(s, 0)$ and we can compute the result after two transitions as

$$\begin{aligned}&\ \ h\ (h\ (s_0, 0)\ es)\ es\\ =&\ \{\textit{definition } h\}\\ &\ \ h\ (f\ s_0\ (es\ 0), 0 + 1)\ es\\ =&\ \{f\ s_0\ (es\ 0) = s_1, \textit{defined above}\}\\ &\ \ h\ (s_1, 1)\ es\\ =&\ \{\textit{definition } h\}\\ &\ \ (f\ s_1\ (es\ 1), 1 + 1)\\ =&\ \{f\ s_1\ (es\ 1) = s_2\ \textit{defined above}\}\\ &\ \ (s_2, 2)\end{aligned}$$

This is the result which we wanted, and it was made possible by the explicit introduction of the time step as an element of the state.

Before going on to the question of deterioration in the context of this system, we make a couple of remarks.

Remarks

1. This new description of the system is equivalent to the old one. Indeed, we have seen that from the initial transition function f we can obtain the h for the monadic system, but we can also obtain a classical transition function f from h by defining:

   $$f\ s\ e = \mathit{fst}\ (h\ (s, 0)\ (\mathit{const}\ e))$$

   that is, by evaluating h at an arbitrary point in time with an input trajectory that is constant everywhere equal to e and taking the first element of the resulting pair.

2. It is easy to generalise from deterministic to other types of systems, because, for any monad M, we have that the functor $F\ S = (\mathbb{N} \to E) \to M\ S$ is a monad. The general transition function will have the type

   $$h : (S\ \times\ \mathbb{N}) \to (\mathbb{N} \to E) \to M\ (S\ \times\ \mathbb{N})$$

Having seen that the system with input considered in the previous chapter can indeed be written as a monadic dynamical system, we can now analyse the question of deterioration for such a system. First, whether we can apply Definition 5 depends on our ability to define a suitable strict order

$$\mathit{worse} :: \mathit{StrictOrder}\ ((\mathbb{N} \to E) \to (S\ \times\ \mathbb{N}))$$

Let us consider, for example, the case of simple n-step vulnerability. There, we would like to compare the initial state to the one obtained after n steps of transition with the system. The statement we would like to obtain is of the form: "state $(s_0, 0)$ is n-step vulnerable to the input sequence $e_0, e_1, ..., e_n$ with respect to the strict order $\prec$ iff the value computed by

$$\mathit{deterioration}\ \phi\ \mathit{worse}\ n\ (\mathit{return}\ (s_0, 0))$$

is *True*", where ϕ is defined by the transition function h. We have taken into account that time has been made an explicit component of the state, and that $\mathit{return}\ (s_0, 0)$ seems a reasonable complex state on which to base the deterioration assessment.

The result of $\mathit{complexApply}\ \phi\ n\ (\mathit{return}\ (s_0, 0))$ is a complex state which, given an input trajectory es can be reduced to the simple state (s_n, n). This suggests defining a strict order which compares the two complex states by applying both to a trajectory es whose initial segment of length n is the input sequence $e_0, e_1, ..., e_n$, that is $\mathit{es}\ i = e_i$, $\forall i \leqslant n$.

$$\mathit{sf}_1\ \text{`}\mathit{worse}\text{`}\ \mathit{sf}_2 = (\mathit{sf}_1\ \mathit{es})\ \prec\ (\mathit{sf}_2\ \mathit{es})$$

It is easy to see that if $\prec$ is a strict order, then *worse* is also one, and that

$$\textit{deterioration}\ \phi\ \textit{worse}\ n\ (\textit{return}\ (s_0, 0)) = s_n\ \prec\ s_0$$

We have, therefore, translated n-step simple vulnerability in a statement about the (potential) deterioration of a system. Similar translations are possible for all the types of vulnerability defined in the previous chapter.

However, the price paid for this translation is somewhat daunting. We have had to modify our transition function by currying it, changing the input set to the set of all possible input trajectories, extending the state space with time stamps for each state, and we have had to use a complicated strict order parameterised on input trajectories.

Even more importantly, the translation is in a certain sense "brittle". In the previous chapter, we have described in the section 2.6 a situation where the exogenous inputs e are given not just as a function of the time step n, as is the case in the definition of simple n-step vulnerability, but as a function of their previous value and of the previous value of the state s of the system representing the vulnerable entity (we leave aside for the moment the action variable $u \in U$). The exogenous inputs are therefore given by a transition function of a system representing the environment which has the form

$$\textit{es}' : S\ \times\ E \to E$$

instead of

$$\textit{es} : \mathbb{N} \to E$$

There is now no way of representing the transitions of the co-evolving system by a transition function $S \to F\ S$ for a monadic F as we did before. If we try, say for instance $F\ S = ((S\ \times\ E) \to E) \to S$ we run into the problem that F is not a monad (in general, $(\to A)$ is not a monad, and because of the interaction between the two transition functions we have the argument to the functor on the left side of $\to$).

On the other hand, the composite system is, trivially, a system with state space $S\ \times\ E$ of the monad Id, with the transition function

$$\begin{aligned}&\textit{compsys} : S\ \times\ E \to S\ \times\ E\\&\textit{compsys} = \textit{pair}\ (f, \textit{es}')\end{aligned}$$

Iterations of this system are trivial to compute. Moreover, if we consider the extension made above to the state space S by the inclusion of a time stamp, we can express the case of exogenous inputs depending on time as a special case of co-evolving inputs:

$$\begin{aligned}&\textit{es}' : (S\ \times\ \mathbb{N})\ \times\ E \to E\\&\textit{es}'\ ((s, n), e) = \textit{es}\ (n + 1)\end{aligned}$$

which expresses that the evolution of the exogenous inputs depends only on time: if the current input is e and the time step is n, the next input is going to be $es\ (n+1)$.

The transition function of the system describing the vulnerable entity has to also be defined on the extended state:

$$f' : (S\ \times\ \mathbb{N})\ \times\ E \rightarrow S$$
$$f'\ ((s,n),e) = f\ (s,e)$$

This extension is also trivial: we just drop the time stamp and use the old transition function.

The composite system starts with $((s_0,0),e_0)$ and then gives in the familiar way the sequence of states of the system under the action of the sequence of inputs:

$$\mathit{compsys} : (S\ \times\ \mathbb{N})\ \times\ E \rightarrow (S\ \times\ \mathbb{N})\ \times\ E$$
$$\mathit{compsys}\ ((s,n),e) = ((f\ (s,e), n+1), es\ (n+1))$$
$$\mathit{compsys}^n\ ((s_0,0),e_0) = ((s_n,n),e_n)$$

A deterioration of the composite system can be ascertained and related to simple n-step vulnerability if we can define a suitable strict order on the set $(S\ \times\ \mathbb{N})\ \times\ E$ in terms of the strict order $\prec$ on S. This is, again, trivial: take

$$\mathit{worse}' :: \mathit{StrictOrder}\ (S\ \times\ \mathbb{N})\ \times\ E$$
$$((s_1,n),e_1)\ `\mathit{worse}'`\ ((s_2,m),e_2) = s_1\ \prec\ s_2$$

Then we have the expected result that the initial system in state s_0 is n-step vulnerable to the sequence es with respect to $\prec$ iff the composite system *compsys* starting in $((s_0,0),e_0)$ suffers, after n steps, a deterioration with respect to worse'.

Again we have obtained a way of relating deterioration and vulnerability, this time by considering a combined system made by the entity and the environment. We have again had to extend the state space of the system by the explicit incorporation of time, but we have only made trivial changes to the transition function and to the strict order considered, and the resulting combined system is of the simple *Id* monad type, instead of the complex $((\mathbb{N} \rightarrow E) \rightarrow)$ monad. Most importantly, this alternative allows us to move very smoothly to the case in which the exogenous inputs depend on the state of the system representing the vulnerable entity, which, in realistic applications, is almost always the case.

3.4 The socio-ecological system

The moral of the previous section is that in order to relate vulnerability to deterioration it is best to consider the combined entity-environment system. We have given mathematical reasons for this conclusion: the resulting definitions are simpler and more general, and we can hope that they will lead to simpler and more general computational tools that we might use to assess vulnerability in given situations.

A closer look reveals that two factors have led decisively to this conclusion:

1. First, the analysis of the common usage of "vulnerability", which has led us to consider the three major determinants of vulnerability statements, to wit: "who" is vulnerable, "to what" it is vulnerable and "with respect to" which criteria it is vulnerable.

2. Second, the decision to model the potentially vulnerable entity as a dynamical system.

Neither of these factors is particularly controversial: one could say that they lie in the nature of vulnerability. It is not surprising, therefore, that the conclusion they imply has been reached before within the climate change community, in the specific context of the systems involved in assessing vulnerability to climate change. For instance, Gallopín (2006)) has argued that "the notion of vulnerability can only be studied in the context of the socio-ecological system").

In this section, we consider mathematical representations of the socio-ecological system, and use them for a better understanding of (Gallopín, 2006).

The definition of a socio-ecological system is given there as "a system that includes societal (human) and ecological (biophysical) subsystems in mutual interaction". Therefore, first of all, the socio-ecological system is a system. Considering the discrete case, we can describe this system by a monadic transition function:

$$ses :: (Monad\ M) \Rightarrow X \to M\ X$$

where X is the state space of the system. We interpret the inclusion of the two subsystems to mean that the states of the socio-ecological system contain information about the states of the societal and ecological systems. In other words, we assume the existence of two functions which extract from a state of the state space X the corresponding states of the two subsystems:

$$\begin{aligned} &ex_S :: X \to S \\ &ex_E :: X \to E \end{aligned}$$

where S and E are respectively the sets of states of the societal and ecological system.

For example, taking the *compsys* as the transition function *ses*, we have

$$\begin{aligned} &ex_S :: (S\ \times\ \mathbb{N})\ \times\ E \to S \\ &ex_S = fst \cdot fst \end{aligned}$$

$ex_E :: (S \times \mathbb{N}) \times E \to E$
$ex_E = snd$

The sentence “SESs are non-decomposable systems” ((Gallopín, 2006)) can seem surprising because, by definition, socio-ecological systems seem to be made of at least two components. However, the mathematical interpretation is perfectly clear: in general there exist no transition functions

$f_S :: S \to M\ S$
$f_E :: E \to M\ E$

such that either of the following diagrams commute:

$$\begin{tikzcd}
f & :: & X \arrow[r] \arrow[d, "ex_S"'] & M\ X \arrow[d, "M\ ex_S"] \\
f_S & :: & S \arrow[r] & M\ S
\end{tikzcd}$$

$$\begin{tikzcd}
f & :: & X \arrow[r] \arrow[d, "ex_E"'] & M\ X \arrow[d, "M\ ex_E"] \\
f_E & :: & E \arrow[r] & M\ E
\end{tikzcd}$$

The evolution of the states of the societal subsystem cannot be studied in isolation from the evolution of the states of the socio-ecological system.

In that case, why talk of “subsystems” at all? The reason is that, as Gallopin states, “it is always possible to single out certain components for study, and this strategy has provided important understanding of the components, as has been traditionally done with great success by social and natural scientists”. In other words, historical and pragmatic factors often lead to the study of systems having transition functions of the form

$f_S :: (Monad\ M_S) \Rightarrow S \to M_S\ S$
$f_E :: (Monad\ M_E) \Rightarrow E \to M_E\ E$

and the transition function of the socio-ecological system is made by combining the two functions via an operation of the type

$ses = combine\ (f_S, f_E)$
$combine :: (S \to M_S\ S) \times (E \to M_E\ E) \to X \to M\ X$

Such operations will be studied and implemented in the further framework development, but we can already see that in general *combine* will not be injective, that is, we will not be able to recover the original systems from the combined one.

3.5 Vulnerability

We can now define vulnerability as potential deterioration in a socio-ecological system. Consider a dynamical system $\phi : T \rightarrow X \rightarrow M\ X$, a function $ex_S : X \rightarrow S$ and a strict order $worse_S$ on $M\ S$.

We remind that, given a strict order $\prec_A$ on a set A and a function $f : A \rightarrow B$, we obtain a strict order $\prec_B$ on the set B by taking $b\_1 \ \prec_B\ b\_2 \equiv (f\ b\_1)\ \prec_A\ (f\ b\_2)$. Therefore, since we have a strict order on $M\ S$, we can obtain a strict order on $M\ X$ via a function $M\ X \rightarrow M\ S$: the natural choice for such a function is $M\ ex_S$. With a strict order on $M\ X$, we can assess whether a given state $mx :: M\ X$ suffers a deterioration after a given time t. Since this deterioration is expressed only in terms of the components extracted via ex_S, we think of this component as having suffered the deterioration, and say that the initial state of this component was vulnerable *to the context* mx. This is the idea summarised by the following definition:

Definition 6 (Vulnerability)**.** With the elements defined above, we say that a state $ms :: M\ S$ is *vulnerable* over a time t to the context $mx :: M\ X$ if

1. $ms = (M\ ex_S)\ mx$

2. mx suffers a deterioration over time t with respect to the strict order induced by $worse_S$, that is

$$\begin{array}{l} vulnerable\ \phi\ ms\ mx\ worse_S\ t = \\ \quad deterioration\ \phi\ worse_X\ t\ mx \\ \quad \textbf{where} \\ \quad mx1\ `worse_X`\ mx2 = ((M\ ex_S)\ mx1)\ `worse_S`\ ((M\ ex_S)\ mx2) \end{array}$$

In the case in which the elements of type $M\ X$ can be obtained by the combination of elements of type $M\ S$ and $M\ E$, that is, we have a function $comb : M\ S\ \times\ M\ E \rightarrow M\ X$, we can define the vulnerability of $ms :: M\ S$ to $me :: M\ E$ as follows:

Definition 7 (Vulnerability to a hazard)**.** A state $ms :: M\ S$ is *vulnerable* over time t to a hazard $me :: M\ E$ if ms is vulnerable over time t to the context $comb\ (ms, me)$.

Computationally:

$$\begin{array}{l} vulnerableTo\ \phi\ ms\ me \\ \quad vulnerable\ \phi\ ms\ (comb\ (ms, me)) \end{array}$$

We can use the functions *vulnerable* and *vulnerableTo* to express all the versions of vulnerability put forward in the second chapter of the document. Let us consider, for example, the "transitional n-step" vulnerability (Definition 8).

In order to compute this type of vulnerability, we have to be given a system $f : X \times E \to X$, a function $e : \mathbb{N} \to E$ representing the inputs to the system at succesive points in time, a function $\tilde{e} : \mathbb{N} \to E$ representing the baseline scenario, a starting state x, and a strict order $\prec$ on the elements of $[X]$ (trajectories of elements of X). In order to compute the transitional n-step vulnerability of f using the computational version *vulnerable*, we have to define a related system, which computes the transitions both according to the given input and to the standard scenarios, and which keeps the histories of these transitions (the trajectories) in the state:

```
$transF : ([X] \times [X]) \times \mathbb{N} \to ([X] \times [X]) \times \mathbb{N}$
$transF\ ((xs, ys), n) = ((xs + [f\ (x, e1)], ys + [f\ (y, e2)]), n + 1)$
  where
  $e1 = e\ n$   -- representing the actual input at step n
  $e2 = \tilde{e}\ n$   -- representing the baseline input at step n
```

We need to compare pairs of trajectories, a similar situation to the one we faced when reducing comparative vulnerability to simple vulnerability. Accordingly, we take

```
$(xs1, xs2)$ `worse` $(ys1, ys2) = \neg\ (ys1 \prec ys2) \wedge$
  $xs1 \prec xs2$
```

The comparison we have defined does not use the time step, which plays the role of input for the system *transF*, therefore we can take as extraction function:

$$ex_S = fst$$

Finally, we have: in the conditions above, the n-step vulnerability is expressed by

$$vulnerable\ transF\ ([x], [x])\ (([x], [x]), 0)\ worse\ n$$

Chapter 4

Software Components for Vulnerability Engineering

4.1 General aspects

Let us now consider possible software components for vulnerability assessments. According to our analysis in the previous chapter the first component we need is an object which allows us to model dynamical systems. As also stated before we have to deal with systems of different types which operate on different datatypes. Moreover these systems have to be combined by adequate methods. A software implementation for vulnerability engineering should therefore provide three features: first of all it has to implement our formal framework with its definitions. Secondly, it must be very modular and highly polymorphic. Last but not least it should allow the integration of existing models to avoid unnecessary modelling work.

A first decision to be taken is the choice of a programming language. We decided to use C++, which is one of the most popular programming languages for the development of industrial applications as well as in science and research. It is a general-purpose, high-level programming language, which also has low-level facilities. In contrast to many other programming languages C++ is a multi-paradigm language, supporting procedural programming, data abstraction, object-oriented programming and generic programming.

Despite its complexity and its multitude of features C++ is quite easy to learn, so that we assume that the ”vulnerability engineer”, even if he is not a programmer by education, will have no problems to get familiarised with the use of our FAVAIA-library.

4.2 Functions

As seen in the presentation of the formal framework and in the previous chapter, one of the preconditions for our vulnerability modelling is a mechanism to handle functions in an easy and intuitive way. In the usual (imperative) programming languages like C++ such a mechanism does not exist a priori. Therefore we had to invent something useful. C++ provides a powerful mechanism for implementing polymorphism and type-computations - the C++ - templates. As both features are important for the handling of functions, we decided to use template metaprogramming for our software library.

With this approach, every function which is intended to be used has to be provided as a template-metafunction. The template for the successor function could look like this:

```
template<typename T>
struct succ {
    typedef T source_type;
    typedef T target_type;

    static target_type apply(source_type t) {
        return t+1;
        }
    };
```

The template has to provide three things: the definition of the types of source/target of the function, and a method *apply* which executes the application of the function. Note that the types for source/target might be fixed and therefore independent of T. For instance a function string_to_double would have fixed types. To be consistent with polymorphic functions it is necessary to express these kinds of functions also as template-metafunctions.

To use these templates as objects at runtime, we provide a class “Function” which is a template class itself and expects two template parameters: the template-metafunction and the type to instantiate it. The class “Function” then defines source and target types again (derived from the instantiated template-metafunction) and an application-operator (which basically calls the underlying apply-method).

```
template <template <typename> class FUNCTION, typename TYPE=void>
class Function {
    ...
    };
```

These functions can now be used as objects. By some additional effort we have implemented operations for composing functions. The following example shows some of the possibilities (Note that the composition works also for polymorphic functions):

```
...
Function<succ,double> SUCC;

cout << SUCC(0) << endl;
// Output: 1

cout << (SUCC following SUCC following SUCC)(1) << endl;
// Output: 4
...
```

4.3 Dynamical systems

The first step taken by the "vulnerability engineer" is the identification of the underlying systems, which represent, among others, the vulnerable entity. For the concrete modeling we need an object representing a dynamical system. As seen in Section 3.1 our definition of a dynamical system uses the concept of a monad. We therefore think that it is a good idea to implement some category theory (including functors and monads). Our class for a Monad looks like this:

```
template <class FUNCTOR,
          template <typename> class RETURN,
          template <typename> class BIND>
class Monad {

  public:
    typedef Function<RETURN, typename FUNCTOR::source_type> return_type;
    return_type Return() {return _return;}

    template <typename F_TYPE>
    Function<BIND,F_TYPE> Bind()
        {...}

  private:
    return_type _return;
    };
```

A dynamical system can now contain a Monad as a private structure and we can immediately come up with the definition of a dynamical system:

```
template <class FUNCTOR,
          template <typename> class RETURN,
          template <typename> class BIND,
          template <typename> class TRANSITION>
class DiscreteSystem : public Monad<FUNCTOR, RETURN, BIND> {

  public:
    typedef typename FUNCTOR::source_type  S;
    typedef typename FUNCTOR::target_type CS;
    typedef Function<TRANSITION, S> transition_function_type;
    ...
    CS Apply(S state) {...}
    vector<CS> Iterate(S initial, int steps) {...}
    ...
  private:
    Monad<FUNCTOR, RETURN, BIND> M;
    };
```

The discrete system basically provides two methods: 'Apply' for applying the system once and 'Iterate' for applying the system n times, delivering a collection of states which were passed during iteration. These methods directly access the bind- respectively return-function of the underlying monad as we show here for 'Apply':

```
CS Apply(S state) {
    return (M.Bind() (
                  Product<CS, transition_function_type>
                     ( (M.Return() (state)), f)
                  )
           );
    }
```

The definition of a special system requires some efforts: the "vulnerability engineer" has to define three different template-metafunctions and a functor before he can define his system.

Therefore a few standard systems are predefined: deterministic, nondeterministic, stochastic and fuzzy systems. As an example let us have a look at the stochastic system:

```
template<typename T>
struct stochastic_bind {
    typedef Product<typename T::target_type, T> source_type;
    typedef typename T::target_type target_type;

    static target_type apply(source_type s) {
         target_type ret, arg=s.outl();
         for (typename target_type::iterator it=arg.begin(); it!=arg.end(); ++it)
             {ret += s.outr()(it->first) * (it->second);}
         return ret;
         }
    };

template <typename STATE_TYPE, template <class> class TRANSITION>
class StochasticDiscreteSystem :
    public DiscreteSystem<typename STOCHASTIC_FUNCTOR<STATE_TYPE>::FUNCTOR,
                          wrap_into_probability,
                          stochastic_bind,
                          TRANSITION> { ... }
```

The number of template-parameters for the instantiation of the system has decreased from four to two: only the type of the state and the transition-metafunction is now needed.

4.4 Deterioration

An essential feature of our vulnerability formalisation is the possibility to express that the state of the system may get worse after its evolution. We therefore need implementation of our notion of getting worse, which is that of a partial strict order. As the handling of partial functions in a programme might lead to difficulties, we modell a new datatype for such partial functions which we call partial_bool:

```
typedef enum { UNKNOWN, TRUE, FALSE } partial_bool;
```

The idea behind this is that our new datatype models three results of a computation: true, false or unknown. If required, we can build up an algebra around this new datatype by

implementing operations like *and, or* respectively *not* for partial_bool.
By the use of this datatype our implementation of a partial order looks basically like this:

```
template <typename TYPE>
class PartialOrder {

  public:
    typedef TYPE complex_state_type;
    ...
    partial_bool operator() (complex_state_type s1,complex_state_type s2)
                            {...}

  };
```

The partial order is defined in a polymorphic way and provides an application-operator for comparing two states returning a partial_bool value. Having a partial order implemented as well as a dynamical system, the computation of deterioration can be done in a straightforward manner:

```
template<class SYSTEM, class PARTIAL_ORDER>
partial_bool deterioration(SYSTEM system,
                           PARTIAL_ORDER partial_order,
                           typename SYSTEM::S state,
                           int steps) {

    typename SYSTEM::CS start_state = system.Return() (state);
    vector<typename SYSTEM::CS> states = system.Iterate(state,steps);
    typename SYSTEM::CS final_state = last_element(states);
    return partial_order(start_state,final_state);

    }
```

4.5 Outlook

The software-package described so far is the current state of affairs in terms of software for the implementation of our formalization. It should be noted that a few things are not done yet:

- the systems implemented right now do not support different time-scales

- the operations for combining systems (in the sense of the SES) are under development.

These tasks will be considered in the upcoming months. An extensive test will be performed by using existing models in the SES-setting and computing vulnerabilities with respect to different preferences. The FAVAIA-Software-Library will be available for downloading at the FAVAIA-website (http://www.pik-potsdam.de/research/research-domains/transdisciplinary-concepts-and-methods/favaia/index_html)
soon.

Notes

[1] http://www.pik-potsdam.de/research/research-domains/transdisciplinary-concepts-and-methods
/favaia/index_html

[2] The notion of potentiality will be addressed later.

[3] In most cases a "not worse" transition will be possible, and one can therefore often just use simple vulnerability.

[4] To spell this out for each case would be rather technical. In Section 3.5 this is shown for "transitional n-step vulnerability" the most complex definition of vulnerability in this chapter (see Definition 8). This is sufficient, because all other definitions, being simpler, are "contained" in this one. E.g. choosing $n = 1$ all results of the n-step case hold obviously for the one-step case. Also, the comparative case is included in the transitional one, because one can interpret a preference order on states as a preference order on sequences: it compares sequences by comparing one special element, in this case the last one.

[5] For distinction comparative vulnerability above should be named "stimulus-comparative" vulnerability or something of the like. We stick to the term comparative as it was used in (Ionescu et al., 2006).

[6] Wanting to be pedantic, one could define this as "transitional stimulus-comparative vulnerability" and similarly define "transitional states-comparative vulnerability" for the case where e is fixed and the two transitions from two different starting states are compared and so on.

[7] An illustration of how definitions based on simple and comparative vulnerability differ can be found below for the concept of an *effective action* from (Ionescu et al., 2006), since the distinction seems most useful in this case.

[8] Note that for this assertion one needs to check two conditions, namely that the action is "action-comparison-effective" and at the same time it is not simply effective.

[9] The term policy function is used in control theory and fits climate change terminology rather nicely.

[10] A very interesting result in the more mathematical analysis of this system interaction is, that the latter case, which looks more complicated in this description, is actually mathematically simpler than the "simple" case where the exogenous input is given. Cf. Section 3.3.

References

Adger, W. N. (2006). Vulnerability, *Global Environmental Change* **16**: 268–281.

Brooks, N. (2003). Vulnerability, risk and adaptation: A conceptual framework, *Tyndall Center Working Paper* **38**.

Denker, M. (2005). *Einführung in die Analysis dynamischer Systeme*, Springer.

Füssel, H.-M. and Klein, R. (2006). Climate change vulnerability assessments: an evolution of conceptual thinking, *Climatic Change* **75**(3). In press.

Gallopín, G. (2006). Linkages between vulnerability, resilience, and adaptive capacity, *Global Environmental Change* **16**(3): 293–303.

Ionescu, C., Klein, R., Hinkel, J., Kavi Kumar, K. and Klein, R. (2006). Towards a formal framework of vulnerability to climate change, *Environmental Modelling and Assessment* . Submitted.

Janssen, M. and Ostrom, E. (2006). Resilience, vulnerability and adaptation: A cross-cutting theme of the international human dimensions programme on global environmental change, *Global Environmental Change* **16**(3): 237–239. Editorial.

Jones, R. (2001). An environmental risk assessment/management framework for climate change impact assessments, *Natural Hazards* **23**(2-3): 197–230.

Kates, R. (1985). The Interaction of Climate and Society, *in* R. Kates, J. Ausubel and M. Berberian (eds), *Climate Impact Assessment: Studies of the Interaction of Climate and Society*, SCOPE Report 27, John Wiley and Sons, Chichester, UK.

Luers, A. (2005). The surface of vulnerability: an analytical framework for examining environmental change, *Global Environmental Change* **15**(3).

McCarthy, J., Canziani, O., Leary, N., Dokken, D. and White, K. (eds) (2001). *Climate Change 2001: Impacts, Adaptation and Vulnerability*, Contribution of Working Group II to the Third Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, Cambridge.

O'Brien, K., Eriksen, S., Nygaard, L. and Schjolden, A. (2006). Beyond semantics: Why conceptualizations of vulnerability matter in climate change discourses, *Climate Policy* . Accepted.

Polya, G. (2004). *How to solve it : a new aspect of mathematical method*, Princeton science library, Princeton University Press. previous edition 1954.

Rutten, J. (2000). Universal coalgebra: a theory of systems, *Theoretical Computer Science* **249**(1): 3–80. url: citeseer.ist.psu.edu/rutten96universal.html.

Sen, A. K. (1979). *Collective Choice and Social Welfare*, Advanced Textbooks in Economics, Elsevier, Amsterdam.

Soanes, C. and Stevenson, A. (eds) (2005). *Oxford Dictionary of English*, second edition (revised) edn, Oxford University Press.

Suppes, P. (1968). The desirability of formalization in science, *The Journal of Philosophy* **65**(20): 651–664.

Turner II, B., Kasperson, R., Matson, P., McCarthy, J., Corell, R., Christensen, L., Eckley, N., Kasperson, J., Luers, A., Martello, M., Polsky, C., Pulsipher, A. and Schiller, A. (2003). A framework for vulnerability analysis in sustainability science, *Proceedings of the National Academy of Sciences of the United States of America* **100**(14): 8074–8079.