

A Linguistic-Based Systematic Approach to Complex System Dynamics and its Application to E-government Introduction in Zanzibar

M.M. Khamis^{1*} and Th.P. van der Weide¹

¹Institute of Computing and Information Sciences, Radboud University,
Toernooiveld 212 6525 EC Nijmegen, the Netherlands

M.Khamis@cs.ru.nl, tvdw@cs.ru.nl

Abstract. System thinking has become an effective strategy when dealing with complex systems. Such systems are characterized by mutual interactions, causality and inter-dependency between system components. A typical example is the cooperation between governmental organizations and stakeholder interaction. The complexity of developing an e-government system suggests a more fundamental approach, where the roles of domain expert and system analyst are clearly separated. The main focus of this article is (1) to propose a linguistically-based systematic approach to the construction of models for the dynamics of complex systems, and (2) to propose extended causal diagrams. Our research methodology is based on Design Science. We start from a conceptual language developed for the application domain at hand and use this to define the dynamic factors. Then, we show how the resulting extended causal diagram is transformed into a framework for System Dynamics. We have demonstrated this approach by using a basic form of an e-government as a running example. Our intention is to use this approach as a basis for a systematic step-wise introduction of e-government in Zanzibar. Besides, this method is useful for modeling any complex system, especially for the description and evaluation of intended policies.

Keywords: Complex systems, conceptual modeling, system dynamics, extended causal diagram, model transformation, e-government introduction.

1 Introduction

E-government has emerged as a means of service delivery improvement, enhancing public sectors by participating in a new knowledge landscape. However, the ability of developing countries to reap the full benefits of e-government is very low and faces many challenges in its implementation.

* Corresponding author

© 2017 M.M. Khamis and Th.P. van der Weide. This is an open access article licensed under the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).

Reference: M.M. Khamis and Th.P. van der Weide, “A Linguistic-based Systematic Approach to Complex System Dynamics and its Application to E-government Introduction in Zanzibar,” *Complex Systems Informatics and Modeling Quarterly*, CSIMQ, Issue no. 11, pp. 85–111, 2017. Available: <https://doi.org/10.7250/csimq.2017-11.05>

The e-government concept has originated in the developed countries. These countries are significantly different from developing countries in many aspects such as economy, infrastructure and culture. E-government implementation is not a process that can simply be inherited from one country to the other. It needs a distinctive study of the case domain to better understand the requirements. Domain understanding creates room to utilize the benefits and to extract the most potential holes that need to be repaired at the earliest opportunity. In that case a system thinking approach [1] is the best approach since it creates a way of unfolding a complex domain making it more understandable. This can be used by policy makers to enhance strategic decision making.

The system thinking approach is defined as a construction of selective abstractions of reality, known as mental models, that help us to derive decisions that inform our actions for the purpose of getting the meaning out of what we are experiencing [2]. Mental models have been used for many years especially in Artificial Intelligence and Psychology. Later, they have become a useful tool in both qualitative and quantitative analysis [3].

According to Richardson ([4], or [5]), *System Dynamics (SD) is a computer-aided approach to policy analysis and design*. It has been introduced as a technique to build simulation models. These models are used to visualize the effects of intended policies. Even if this may have a small impact on policy implementation, it can be helpful for getting a quick impression of the effects of intended policy changes [6].

A drawback is that building a SD application is not an easy task, especially when a complex application domain is involved (see e.g. [7]). It basically consists of 2 steps. The first step builds a causal network that shows the relevant (dynamic) factors and how they influence each other. The second step expands causal networks with a set of recurrence relations that describe the progress of the system each time step. The intention of this article is to describe this building process in a systematic way by separating domain-specific activities (such as modeling and evaluation of the results) from more technical steps.

Building a causal network can be simplified by using semi-natural language techniques to relate the (more abstract) modeling concepts with (more concrete) domain-specific descriptions. The intention then is to first describe the language that is used by experts from that domain as a controlled language. This language allows the domain experts to be involved in the modeling process.

A rigid approach to conceptualize an application domain is Object Role Modeling (ORM). ORM basically aims at building a grammar that describes a simplified (structured) yet semi-natural version of domain language. The structure and meaning of that language is formally defined by ORM. Consequently, the domain description is understandable within the application domain, and understandable by the system builders.

Traditionally, ORM is used to build information systems, since from the information grammar a relational database structure is easily derived. However, the information grammar can also be applied as a first order logic system that can be used to build a formal reasoning system. The information grammar describes the elementary sentences that are valid in the associated application domain. From these sentences other sentences may be formed. Object Role Calculus (ORC) [8] and ORM2 [9] are examples of such generic systems for constructing sentences. In formal terms, the information grammar may be seen as the signature of a first-order logic theory [10], together with a set of axioms, derived from the population rules. The associated logic theory is a basis for formal reasoning about the domain. Proof assistants (such as the Coq Proof Assistant [11]) may help policy makers.

The major goal of this article is to provide a systematic and structured method to analyze inter-dependency in the context of the conceptual scheme, and to build a simulation tool by which government is supported to get a quick and well-founded impression of the effects of their various managerial choices. We use the ORM approach to generate the information grammar in its graphical representation as a conceptual scheme to be used as a basis for reasoning about the

application domain. We follow the SD approach to find relevant dynamic factors and how they influence each other. In causal reasoning such an influence can be positive definite, negative definite or none of these both.

Causal diagrams allow for qualitative reasoning about (the effects) of SD. This reasoning is only effective when loops are involved since they describe the inherent system options for self-reinforcement and self-stabilization. But even then reasoning on the basis of a causal diagram is rather poor. In this article we propose an extension of causal diagrams that makes their expressive power slightly better. We also discuss how the expressive power may be further extended in a semi-quantitative way by extending influences with a strength and a velocity indicator.

We describe causal reasoning itself more systematically based on the selection of target factors that the policy makers want to optimize, and introduce the active diagram for these factors. We give some properties of Extended Causal Diagrams (ECD), and show rules by which an ECD can be transformed into an SD structure. This structure then has to be enriched with quantitative expressions that define how a new system state is calculated from the previous state.

The layout of this article is as follows. The article has 2 complementary story lines. After a literature review in Section 2 we continue the theoretic story line with the introduction of extended causal diagrams in Section 4; we provide its syntax and semantics, provide some modeling guidelines and describe causal reasoning. We continue the theoretic line in Section 6 with the transformation of the ECD into an SD structure.

The other story line uses the e-government example to demonstrate the techniques described in the theoretic story line in this article. In Section 3 we discuss the e-government conceptual scheme that we will be using. Then in Section 5 we focus on a particular managerial goal for e-government, and show how the ECD is systematically derived for this case. Then in Section 7 we use the transformation rules from Section 6 to transform the ECD into an SD structure. We use the Sysdea tool to develop a simulation tool.

We close the article in Section 8 with some conclusions and suggestions for further research.

2 Literature Review

Several methods have been proposed to extract complex application domain in a relatively easy way. These methods are used to represent complex problems in a normative approach. Failure to understand domain problems and incorrect interpretation of the case domain may cause technical failures of the proposed solution. Proper interpretation is important for a relevant modeling design. There are three dimensions to characterize a modeling method: the theoretical, the methodological and the domain method [12].

The domain approach focuses on understanding of the physical system. This dimension connects causality changes to normative knowledge, often related to dynamic phenomena. Domain modeling characterizes the behavior of a mental model in a system thinking context. A term “model” was used in this study to describe a product used to represent a system of interest. The model becomes a tool when it has a salient feature to study the behavior of the actual system. A mental model is defined as the representation of the external system that conceptualizes historical, existing or projected system aspects [13]. Historically, mental models have been used for many years especially in Artificial Intelligence and Psychology. They have become a useful tool to be used in most areas where both qualitative and quantitative analysis are being manipulated [3].

There are many ways to elicit mental models. The causal diagram is the most suitable conceptual diagram to represent dynamic characteristics [14]. But, concept mapping, cognitive mapping and causal mapping are the most popular approaches, especially when making a design for strategic applications [15].

The causal diagram is a modeling technique used to promote understanding of nonlinear behavior of complex systems [16]. This type of diagrams is used to explore the dynamic behavior of

a system [17], but also to extract empirical data using auxiliary experiments and observations [16]. Besides such data science applications, causal diagrams are nowadays also used in modern applications to make the utility of large data sets more potent [18]. Indeed, they can be used as analysis tool without the need for simulation experiments [6]. The impression of causal diagramming shows some limitations to present the entire case domain in a realm of SD [19]. Causal modeling influences variables in a form of feedback loops, but it cannot elucidate the detailed information of the model such as stock, flow and implicit influence. For the case of stock and flow representation, causal diagrams deploy SD techniques to illustrate the details [20]. But, for implicit influence, this has not yet been explored. Besides, the analysis using causal diagrams is not that simple; it requires not only understanding and knowledge of this modeling approach, but also some experience in its application [15].

Concept mapping has been used in many applications to support qualitative reasoning. It is largely used to express and organize knowledge in various levels of complexity. Concept mapping was used to analyse complex situations such as climate change [21]. However, exploiting this approach may cause some restrictions in modeling. Strictly spoken, it cannot separate structure from behavior, and thus puts more focus on influences than effects. But, it is very handy to assess conceptual knowledge especially in embedded technology based environments [22].

Another technique is cognitive mapping. Cognitive mapping is a method to represent personal constructs in a form of note taking. It uses chunks of 10–12 words as nodes to disclose concepts and their relations concerning the problem at hand. This technique is mainly used to improve problem understanding or to explore the opportunity on some certain theory. It structures a problem and analyzes it to make sense of details in a form of assumptions, action possibilities, strategies and goals. However, the cognitive mapping method does not provide much information, thus it is combined with other research methods such as observation and interview to increase the validity of findings. It is mostly used for qualitative data analysis [23].

Causal mapping is a technique that captures expert knowledge about the application domain using linking strategy of what people think and how they act. It represents domain knowledge more descriptively in a way that it articulates many ideas from both individuals and groups. Its way of knowledge representation uses high level of variable abstraction. Thus, causal mapping is applied largely to investigate manager decisions [24].

The main reason of selecting causal diagrams over other methods is the end results of the modeling. The causal diagram puts more emphasis on the dynamic behavior during the modeling, specifically on the use of cause and effect relation. This technique focuses on feedback loops and loop effects of the design. On the other hand, cognitive mapping investigates the constructs that need extra details from interviews or observation to understand a particular knowledge. Using constructs in variable construction reveal descriptive version of the facts which is more or less relevant in qualitative analysis. Again, the causal mapping technique concentrates on the strength of the link relationship that mostly position the construction process to a subsequent analysis techniques such as statistical analysis [25] or Bayesian network methods [26].

However, due to the wide spread use of causal diagrams in policy analysis, they have been used in many applications to overcome policy resistance. For instance, [27] deploys causal diagram modeling to understand public health risk behavior in hospital context including environments, resources, and delivery system. Furthermore, [28] has improved the knowledge of the complex logistics in food supply chain by investigating the behavior in food industry specifically in poultry production and processing. Also, causal diagram is used with other modeling methods such as Object Role Modeling [29] and cognitive mapping [30] to enhance decision making.

These works show various approaches for data collection, system design and integration of the data in the system. A common feature is, however, that the actual work is usually done by a domain expert (e.g. social scientist, environmental scientist) which is then delivered to a computer modeler to integrate the data and ideas in the model that should be developed. This basically one-way

communication may cause hard-to-trace inconsistencies between the various steps in the chain (from theory to meta model, from meta model to data collection, from rough data to interpreted data, from data to model and so on). Moreover, a lack of two-way interaction may result in missed opportunities, e.g. more efficient data collection or more explicit theory.

The quality of conceptual design is influenced by the quality of the conceptual modeling method [31]. A good model should satisfy a number of conditions. According to [32] the most important are: (1) the communication principle, stating that the primary intention of the model is not to model the structure of the application domain, but to model the structure of the communication about the application domain by its participants; (2) the *100% conceptualization principle*, stating that the model covers all the aspects of the user communication, but nothing more, see also [33]; (3) the *redundancy free principle* which requires only elementary concepts to be modeled; (4) the *unification principle* which ensures that all constructs can be populated; (5) the *substitution principle* which states that the elementary sentences can be regenerated; (6) the *generic principle* which requires a balance between generality and specificity; and finally (7) the *minmax principle* stating that the model should be a minimal grammar to accept the sentences provided by the domain expert (also referred to as Ockham's Razor), and yet maximally expressive (by adding abstractions when they are likely to be profitable). Additionally, the quality of the model is highly influenced by the modeling process and its participants. See [34] for an overview of the process and the required competencies of its participants. In [34] this process is discussed from the perspective of *domain expert* and *system analyst* and describes the competencies required for both roles, and argues that these competencies ensure a successful modeling process.

Using a natural-language based approach also helps to validate the constructed model. A formal model of an informally described application domain cannot be proven correct, but only shown to be incorrect. This is called the *falsification principle* (Popper, [35]). A model is seen as plausible when enough efforts have been put to falsify the model. Showing that a model is plausible can be more effective when all stakeholders share a common language that is understandable for them from their point of view. We argued that the ORM approach uses a semi-natural language (also referred to as controlled language [36]) approach for this purpose. The system analyst will detect a model flaw when the domain expert refers to a construct not (yet) present in the current state of the information grammar. The system analyst will also actively try to find inconsistencies by performing semantic checks. A semantic check typically is a question from the system analyst to the domain expert in the form: "*Does the following query have the following result?*". The domain expert will base the answer to the query on domain knowledge while the system analyst uses the formal techniques of the modeling method to calculate the answer.

3 Modeling Process

In this section we describe the conceptual model that will be used to demonstrate a non-trivial application of the Extended Causal Diagrams technique that we will introduce in the next section. This conceptual model was developed in [37] to study the context for successful implementation of e-government in Zanzibar. The model describes the case domain that will be used in Sections 5 and 7 to extract a framework for the causality to (positively) influence e-government implementation to provide better public services.

Figure 1 shows a graphical representation of this model. The model was designed using the ORM modeling approach; see [38] and [39]. The model describes the concepts and their relations as far as they are relevant for implementation strategies in e-government to provide better public services. The model categorizes these concepts into three groups: actors, actions and enabling factors. Citizens, organizations (such as companies) and governmental offices are the actors in this conceptual model. The citizen is anyone who has been registered as an inhabitant of the country considered. Citizens are consumers, but also deliver work force. An organization is any legal entity registered in the country considered. Organizations are the producers and in need of work force.

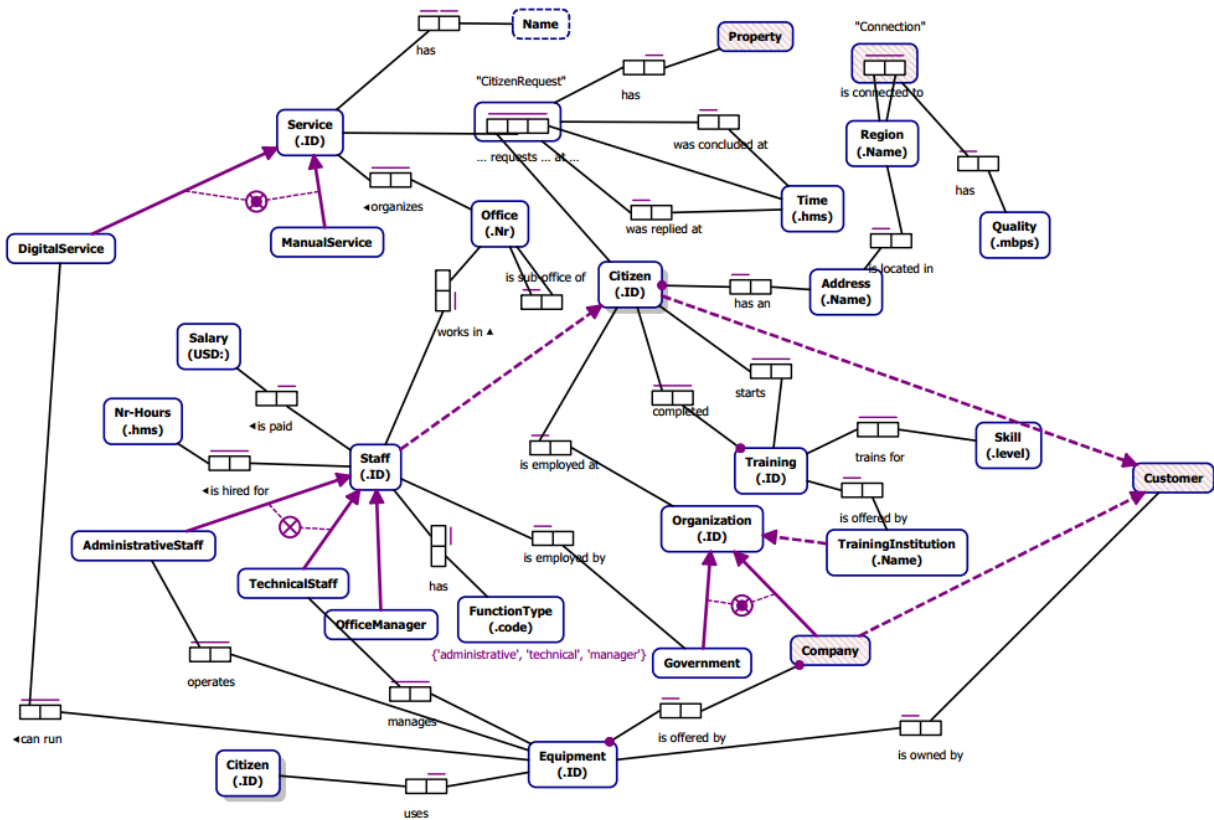


Figure 1. Conceptual design using ORM modeling

A governmental office is uniquely identified and named by its dedication of specific professional activities, supported by relevant staff specialization.

It is the task of the government to enable effective and efficient interactions between its citizens and organizations. Government guarantees a legal, safe and productive environment to align the demand and supply. Services are the main instrument of government to steer and support this process. In our model, a service is any action provided by government to citizens and organizations.

Traditionally, services are delivered manually, where people interact with people at specific locations at specific times. Manual service delivery typically is slow and costly.

Digital service delivery can improve the delivery of services considerably. For interaction, the interactors are not required to be at the same place at the same moment. Also, digital service delivery allows a very fast handling of the required communications [40]. This enables the government to implement its service delivery much more efficiently by at a much lower cost. The result will also help the country to improve its international position, and to participate increasingly in international trade.

In order to use digital service delivery, the citizens need to have access to the electronic highway for digital interaction. Citizens need to be trained to have the appropriate skills, but also they need to have the required tools (computer, internet connection, etc.). For organizations the same applies, but usually they are one step ahead of citizens in this respect.

The government may thus consider a number of strategies for improving acceptance and usage of ICT. In this article, we restrict ourselves to:

1. Improving the infrastructure (power supply, internet connections, etc.). The quality of the infrastructure is a critical success factor for a successful introduction of digital services. However, improving infrastructure is a slow and costly process.
2. Improving the overall skill level of citizens. The government may want to improve skills strategically by a fine tuned course offering (or the stimulation of (new) organizations to offer

these courses). In general, education not only is costly and slow, but probably the most critical success factor. Special courses may be cheap and fast, when they are used to fill a specific skill gap.

In the next section we will introduce Extended Causal Diagrams (ECD) as a method to define causalities that play a role in the context of the options considered. Then, in Section 5, we apply this technique to define the causalities that play a role in the e-government context described in this section. After that, in Section 6, we describe how an ECD is systematically transformed into a SD. This we apply in Section 7 to obtain a SD model for this e-Government case. This model then is prototyped in a concrete SD tool to show some simple simulations.

4 Causal Reasoning

A conceptual model provides a static description of the underlying application domain. A conceptual model may be extended with behavioral aspects. We will follow the approach as developed in [41]. In this approach causal diagrams are used to develop SD models.

A conceptual base normally is the result of a particular focus on the application domain [42]. The conceptual base, in terms of the conceptual model, provides the dimensions that describe the states of the (considered part of the) application domain. Since a conceptual model such as ORM defines a semi-natural domain language, this (quantitative) focus can be most conveniently described in terms of this conceptual language. The major advantage of this approach is that the definitions made can be understood and verified by (non-technical) domain experts.

The proposed conceptual base will consist of those concepts that span the state space for the focus in question. The elements of the conceptual base (also referred to as dimensions) are referred to as substantial variables in this article. Each state of the application domain is represented as a population of the conceptual base. The value of a substantial variable is the number of elements (the size) of its associated population.

Besides the substantial variables, there also will be a dashboard of characteristics defined in terms of the conceptual model. Dashboard variables are typically described in terms of a computation over the underlying conceptual model. They correspond to functions that quantify some aspects of the application domain in terms of the underlying conceptual model. Both the substantial and dashboard variables are the variables of the causal diagrams.

A complete quantitative description of real life applications tends to be very complex, since such systems usually can be indicated as complex dynamic systems (see [43]). Causal reasoning provides a qualitative first order approximation of the dynamics of the considered system. A first order system description is an approximated (linear) description of the system in which only first derivatives play a role. This means that only the (partial) derivate $\frac{\partial B}{\partial A}$, where A and B are dimensions, is considered. This derivate indicates how changes in variable A influence the value of variable B .

4.1 Causal Diagrams

Causal relations describe elementary causal influences between variables (see, e.g. [44]). In causal diagrams, these relation can be binary only. A causal relation between variables A and B indicates that a quantitative change of variable A leads to a quantitative change in variable B . This relations corresponds to the partial derivate $\frac{\partial B}{\partial A}$. Causal relations may be accompanied by a polarity indicator (positive or negative) to qualify the effect of the associated causality (see Figure 2):

1. A positive causality indicates that $\frac{\partial B}{\partial A}$ is positive definite, meaning that $\frac{\partial B}{\partial A} > 0$ for all A . This is denoted as $A \xrightarrow{+} B$, see Figure 2(a). Consequently, (1) an increase in A leads to an increase in B , and (2) a decrease in A leads to a decrease in B .

2. For a negative influence $A \xrightarrow{-} B$ the derivate $\frac{\partial B}{\partial A}$ is negative definite, meaning that $\frac{\partial B}{\partial A} < 0$ for all A , see Figure 2(b). In this case, (1) an increase in A leads to a decrease in B , while (2) a decrease in A leads to an increase in B .
3. The absence of a polarity indicator, see Figure 2(c), indicates that the derivate $\frac{\partial B}{\partial A}$ is neither positive definite nor negative definite, or, $\frac{\partial B}{\partial A} > 0$ for some value of A , and $\frac{\partial B}{\partial A} < 0$ for some other value of A .

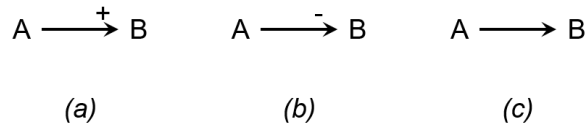


Figure 2. Direct causal relations

Arrows in the causal diagram correspond to elementary causal relations between the corresponding variables and are used to describe dynamic aspects of the application domain. Arrows may be combined into paths, to show non-elementary causal effects. The polarity of a path is obtained from the polarity of its components. If all arrows on the path have a polarity indicator, then the polarity of the path is positive if the number of negative polarities on the path is even, and negative otherwise.

Cycles are most important paths in causal diagrams. A cycle is a closed feedback system. Each variable in a cycle (also) is influenced by itself. There are two special types of causal loops. A *balancing loop* is a cyclic path that has a negative polarity indicator, whereas a *reinforcing loop* is a cycle with a positive polarity indicator [44]. These special kinds of loops allow to reason about influencing the underlying application domain in terms of causal effects in a semi-quantitative way: they give the policy makers opportunities to stimulate growth effectively and to create stabilization.

If a factor in a reinforcing loop is increasing, then, when there are no other influences, this leads to an unstoppable increase of all factors in the loop. For policy makers such loops are most useful to start a self-reinforcing development. Note that the policymaker will want to use another factor to stop the resulting growth explosion at some appropriate moment.

Balancing loops are different. An increase of a factor in such a loop leads to a decrease of that factor by the causal effect from the loop. Balancing loops enable the policymakers to build self-stabilizing systems.

4.2 Extended Causal Diagrams

Usually it is difficult to separate linearity and non-linearity [45]. The first derivative approach of causal diagrams is meant to approximate the behavior of the application domain in a linear mode. Non-linear behavior is added to fine-tune a linear model for a more appropriate description of the application domain behavior. Non-linear behavior is described by higher order derivatives.

The Extended Causal Diagram (ECD) is an extension of the traditional causal diagram as described in the previous subsection. In ECD's we also consider how the influencing of variable A by variable B can change by modifications in variable C . This corresponds to the following higher order derivate:

$$\frac{\partial \frac{\partial A}{\partial B}}{\partial C} = \frac{\partial^2 A}{\partial C \partial B}$$

This causal relation is denoted as $C \xrightarrow{*} [A \xrightarrow{*} B]$, see Figure 3(a). Note that this allows the system analyst to add more application domain semantics into the ECD than would be possible when the situation from Figure 3(a) would be 'flattened' into the scheme in Figure 3(b).

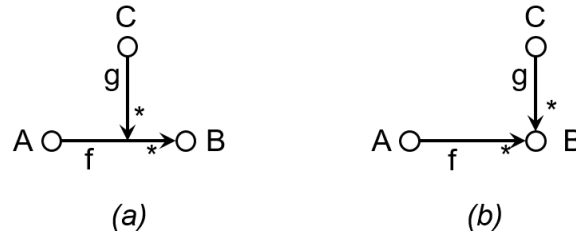


Figure 3. Indirect influence and its flattened version

Formal syntax. The formal syntax for extended causal diagrams assumes the syntactic categories `SubstantialFactor` of substantial variables and `DashboardFactor` of dashboard variables as described in the introduction of this section and extends these to the syntactic category `CompoundFactor`. Using this category, the syntactic category `CausalRelation` is extended. Also in this case the causal relation may have a polarity indicator. The resulting grammar is (see Figure 4 for a graphical overview):

1. `CausalRelation::SubstantialFactor, LeadsTo, SubstantialFactor`

<code>SubstantialFactor, Influences, SubstantialFactor</code>
<code>DashboardFactor, Influences, DashboardFactor</code>
<code>SubstantialFactor, Influences, [, CausalRelation,]</code>
2. `LeadsTo:: \implies | $\overset{+}{\implies}$ | $\overset{-}{\implies}$.`
3. `Influences:: \longrightarrow | $\overset{-}{\longrightarrow}$ | $\overset{+}{\longrightarrow}$.`

We will use the generic notation $A \overset{*}{\rightarrow} B$ to denote either $A \rightarrow B$, $A \overset{+}{\rightarrow} B$ or $A \overset{-}{\rightarrow} B$. Likewise, we use $A \overset{*}{\implies} B$ to denote either `Flow [B \rightarrow](A)`, $A \overset{+}{\implies} B$ or $A \overset{-}{\implies} B$. Note that this definition of causal relations allows for higher order constructs such as for instance: $D \overset{*}{\rightarrow} [C \overset{*}{\rightarrow} [A \overset{*}{\rightarrow} B]]$.

Special cycles. As an example, consider the Extended Causal Diagram in Figure 5. We have a reinforcing loop $A - B - C - D$. The policy maker has the option to influence the reinforcement strength via the factor E . Increasing E enhances the reinforcement effect of the loop, but the opposite effect is obtained by decreasing E . A special case is when the influence by E causes the effect of the causal relation $B \overset{+}{\rightarrow} C$ to become zero.

New kinds of cycles can be defined in ECDs. The general format is $A_n \overset{*}{\rightarrow} [A_{n-1} \rightarrow \dots [A_1 \overset{*}{\rightarrow} A_0] \dots]$. An example of the simplest case $B \overset{-}{\rightarrow} [A \overset{+}{\rightarrow} B]$ is shown in Figure 6(a).

In this case, the causal relation $A \overset{+}{\rightarrow} B$ is balanced by B . For instance, let A correspond to the factor `Population` and B to the factor `Food`. More food leads to an increasing population $\text{Food} \overset{+}{\rightarrow} \text{Population}$, but an increasing population has a decreasing effect on the strength of this causality:

$$\text{Population} \overset{-}{\rightarrow} [\text{Food} \overset{+}{\rightarrow} \text{Population}]$$

Another special case is a situation like $A \overset{-}{\rightarrow} [A \overset{+}{\rightarrow} B]$, see Figure 6(b). In this case, an increase of factor A leads to an increase of factor B . But at the same time, this increase has an effect on this causality. For instance, in an educational situation where courses are offered to the public and citizens register for courses, we have the causal relation that offering more courses will lead to an increase in course registrations. However, offering more courses may also confuse citizens when making courses. This would be described by

$$\text{Training} \overset{-}{\rightarrow} [\text{Training} \overset{+}{\rightarrow} \text{Registration}]$$

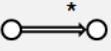
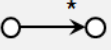
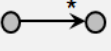
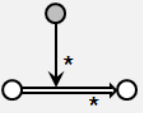
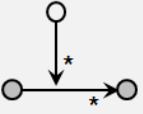
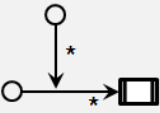
Symbols	○	substantial variable
	●	dashboard variable
	□	causal relation
Causal relations: Base		via state change
		between substantial variables
		between dashboard variables
Step		control flow of instances
		control influence dashboard variables
		control influence of substantial variable

Figure 4. Graphical syntactical overview

Semantics. Next we focus on the semantics of these constructs. First we consider substantial variables. Substantial variables have associated a population of instances. The size of the population of variable A is denoted as $n(A)$. The causal relations $\text{Flow}[B \rightarrow](A)$, $A \xrightarrow{+} B$ and $A \xrightarrow{-} B$ indicate that instances from variable A move to variable B . Consequently, the variables A and B must correspond to type-related concepts in terms of the underlying conceptual model. Suppose we would introduce dashboard variables $n(A)$ and $n(B)$, then the semantics of causal relations $\text{Flow}[B \rightarrow](A)$, $A \xrightarrow{+} B$ and $A \xrightarrow{-} B$ are defined as $n(A) \rightarrow n(B)$, $n(A) \xrightarrow{+} n(B)$ and $n(A) \xrightarrow{-} n(B)$ respectively.

With each causal relation C we associate a derivate $\mu(C)$, defined inductively as follows:

Induction base

- $\mu(A \xrightarrow{*} B) = \frac{\partial n(B)}{\partial n(A)}$ when A and B are substantial variables.
- $\mu(A \xrightarrow{*} B) = \frac{\partial n(B)}{\partial n(A)}$ when A and B are substantial variables.
- $\mu(A \xrightarrow{*} B) = \frac{\partial B}{\partial A}$ when A and B are dashboard variables.

Induction step

- $\mu(A \xrightarrow{*} C) = \frac{\partial \mu(C)}{\partial n(A)}$ when A is a substantial variable and C is a causal relation.

The semantics of causal relations are defined as differential inequalities as follows. Let C be a causal relation, then:

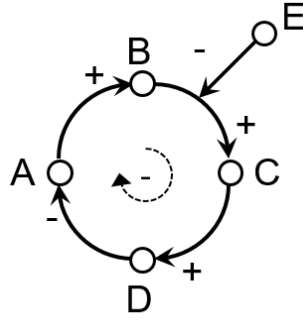


Figure 5. Cycle in ECD

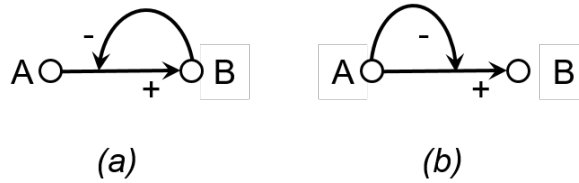


Figure 6. Implicit cycles in ECD

1. if C has a positive polarity indicator, then C expresses the differential inequality $\mu(C) \succ 0$, where $f \succ g$ is a relation between functions that indicates that for all arguments the value of f is larger than the corresponding value of g .
2. if C has a negative polarity indicator, then C expresses the differential inequality $\mu(C) \prec 0$, where $f \prec g$ indicates that the value of f is smaller than the corresponding value of g .
3. if C has no polarity indicator, then $\mu(C)$ can take both positive and negative values. This is denoted as $\mu(C) \approx 0$, where $f \approx g$ means that for some argument the value of f is different from the corresponding value of g .

Summarizing, the semantics of an Extended Causal Diagram $\langle \mathcal{F}, \mathcal{C} \rangle$ consists of a set of variables \mathcal{F} with a differential inequality equation for each causal relation $C \in \mathcal{C}$.

Our feeling is that the extra features of ECD compared to CD better suit the description of logistic growing behavior (see for example [46]). For instance, the ECD consisting of variables x and y with the causal relations $x \xrightarrow{+} y$ and $x \xrightarrow{-} [x \xrightarrow{+} y]$ leads to the system of differential inequalities

$$\begin{aligned} \mu(x \xrightarrow{+} y) \succ 0 &: y'(x) > 0 \\ \mu(x \xrightarrow{-} [x \xrightarrow{+} y]) \prec 0 &: y''(x) < 0 \end{aligned}$$

which can typically describe a plateau function like $y = \frac{1}{1+e^{-x}}$ as is easily verified (see Figure 7). But other solutions are also possible. Suppose, for instance, that the domain for variable x is restricted to $(0, \infty)$, then $y = \ln(x)$ also satisfies this system of differential equations.

An Extended Causal Diagram can be made more explicit by quantifying the differential equations. In Section 7 we will discuss the transformation to a SD diagram as an example.

4.3 Further Extensions

In the traditional approach to causal diagrams, the diagram does not indicate a time fragment of the causal effects. The effect of a causal relation may be immediate (i.e. effective after a very short time) or may be long-term (i.e. effective with a larger delay). For instance, a causal relation that is

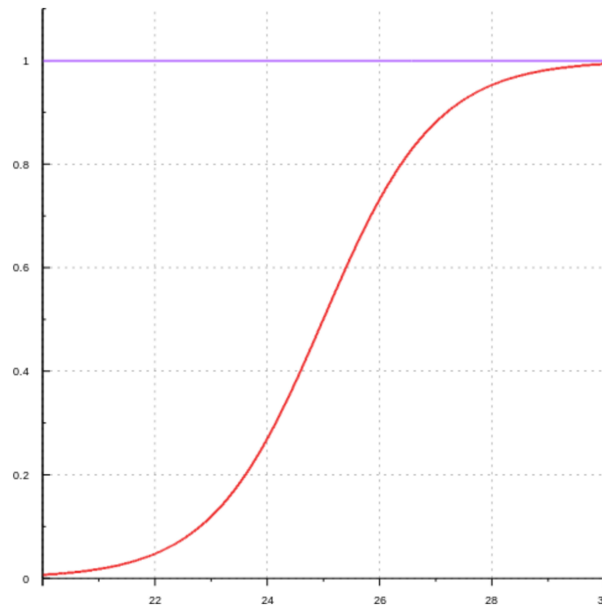


Figure 7. A typical plateau function: $y = \frac{1}{1+e^{-(x-25)}}$

conveyed by traditional publishing on article will not be as fast as a information exchange based on modern information technology. We will not discuss this issue in this article.

In this (traditional) approach to causal diagrams, it is also not possible to describe a varying strength of causal relations. For instance, improving the factor conditions may lead to a stronger effect of the casual relation that describes the relation between the factor material and the factor product. This is what we will add to obtain Extended Causal Diagrams.

We assume the following properties for causal relations. The causal relation $A \rightarrow B$ is called *fast* if the effect of a change in A in changing B happens in a sufficiently short time. The causal relation $A \rightarrow B$ is called *weak* if this effect on B is below some threshold. The notions of *sufficiently short* and *weak* are to be determined by the domain expert.

We extend these definitions to paths as follows:

- A path is called fast if all its constituent causal relations are fast. In all other case the path is not seen as fast.
- A path is called weak if all its constituent causal relations are weak. In all other cases the effect of the path is not seen as weak.

This allows us for the reasoning following steps to estimate the cumulative effect of two cycles P_1 and P_2 having the factor A in common. For instance, if P_1 is reinforcing and fast while P_2 is balancing but not fast, then the accumulated effect is a plateau-effect (see Figure 7): a positive change in A will first lead to reinforcement that later is muted by the balancing effect from P_2 .

4.4 Constructing Guidelines

The construction of an Extended Causal Diagram from a conceptual model for a given focus roughly consists of the following steps:

- **Step 1** – Determine the substantial variables that are necessary and sufficient to span the state space of the focus.
- **Step 2** – Introduce the dashboard variables that characterize properties that are relevant for the focus.
- **Step 3** – Identify the elementary causal relations between the factors.

In the next section we discuss some cases to demonstrate this transformation procedure.

4.5 Causal Reasoning

The extended causal diagram allows policy makers to reason in a semi-quantitative way about the underlying application domain. We discuss a number of aspects of this reasoning. We will see an ECD as a labeled directed graph $G = \langle V, E, \lambda \rangle$, where the nodes (V) are the factors of the ECD, the edges (E) the connections and λ a function assigning the polarity indicator to each edge.

Firstly, the policy makers want to distinguish independent and dependent factors. The other (dependent) factors can be influenced via these elementary factors. So the independent factors provide the managerial basis for the policy makers. Formally, a node without incoming edges is an independent factor. Let $\text{base}(G) \subseteq V$ be the basis of graph G , the set of independent nodes.

Secondly, the policy makers will want to indicate a (set of) factor(s) as target factor(s). Their policy will be to optimize on the target factors. Then the ultimate success of their policies will be measured in terms of these target factors. In our example, the factor `DigitalUser` is chosen as the (single) target variable. Let $T \subseteq V$ be the set of target variables.

We call the restricted ECD diagram G to the target variables T the active diagram, denoted as active (G, T) . This diagram consists of all vertices v such that there is a path from v to some target variable. Only the edges encountered on such paths are part of the active diagram, and the labeling function is the restriction of λ to the active labels.

Thirdly, policy makers will want to detect the balancing and reinforcing loops in the active diagram. These loops provide special opportunities for uncontrolled growth and for stabilisation.

In the next step, the policy makers detect all control paths in active (G, T) . A control path connects an independent factor with a target factor. The polarity of a control path indicates how this path contributes to ultimate goal in terms of the target factors.

After these steps, the policy makers have to find their best options by selecting the most effective control paths. For each control path it will be important to estimate:

1. The strength of the causal effect of the control path.
2. The time frame required by this control path to become effective.
3. The investment costs associated with this control path.

This will help the policy maker to make a price/performance analysis, on the basis of which the best control paths are selected.

5 The E-government Application

In this section we demonstrate the procedure described in Section 4.4 to derive an Extended Causal Diagram in a systematic and structured way from the description of a focus in the context of an underlying conceptual model.

In Section 3 we have described a conceptual model for the successful implementation of e-government and defined as focus of attention the stimulation of citizens to use the digital services that are offered in the context of e-government. Our main goal in this section is to extract all possible opportunities, threats and other related variables that affect e-government implementation, and to investigate what causal relations exist between these variables.

Since ORM is a language-based conceptual modeling technique, we can use the associated semi-natural domain language (referred to as Object Role Calculus, ORC) to define these variables such that (non-technical) domain experts can easily recognize (and validate) the variables chosen.

We will make our analysis in a number of steps. The first step is very basic and only focuses on viewing the number of citizens using manual services (manual users) and the number of those using digital services (digital users). In the second step, the effect of education is considered, while the third step focuses on the impact of infrastructure. The rationale for this is that in order to be using digital services, a citizen need to have sufficient skills and sufficient tools.

After these steps, we have a complete causal overview of the focus area. This Extended Causal Diagram then will be used in Section 7 to construct a simulation tool.

5.1 Step 1: Basics

Our goal is to keep track of the number of users who are using digital services and those who are using manual services. This goal is visualized in Figure 8. Keeping record of how these figures develop over time helps in understanding the development progress. We analyze this goal via the steps described in Section 4.4 to extract the relevant variables for the extended causal diagram being developed, and their causal relations.

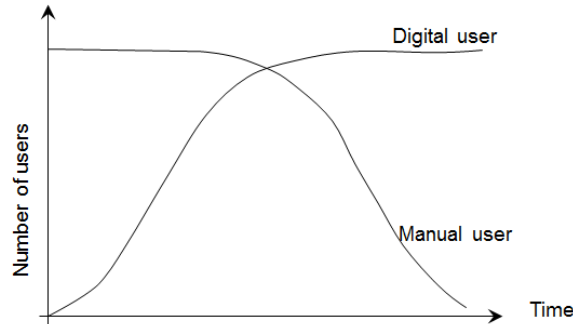


Figure 8. Keeping track of manual and digital users

To achieve this goal we restrict the e-government conceptual model (see Figure 1) to those object types that are associated with manual and digital use of services. Manual and digital users by themselves are not defined by this conceptual scheme. But they can be defined in terms of concepts within this conceptual scheme. Digital users are described as citizens who use electronic tools. They are recognized by sending service requests using electronic tools. Manual users are citizens who do not use electronic tools. They are identified as citizens making requests to manual services. The relevant subscheme is highlighted in Figure 9. Using the semi-natural language defined by the ORM scheme, we may introduce the following new concepts as follows:

- DigitalUser IS Citizen involved in Request for DigitalService.
- ManualUser IS Citizen BUT NOT DigitalUser.

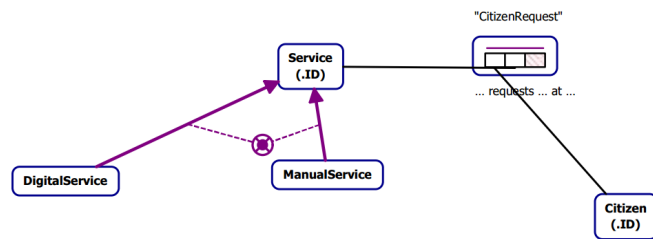


Figure 9. Digital and Manual Users

By introducing the object types DigitalUser and ManualUser, we have presented a conceptual base for the object type Citizen (since each citizen is in either state). These new object types are the factors for the extended causal diagram under construction (see Figure 10). In this example we do not consider the death rate of the population. The assumption is that both manual and digital users are proportionally influenced by birth rate factor of a population. Furthermore, we assume that new born citizens are categorised as manual users. Manual users are transformed into digital users when they start to use digital services. External factors can influence this transition positively or negatively. For instance, offering training programs and improving infrastructure will have an

effect on this transformation. We will come back to this in the next steps. The effect of these steps will be the flow of instances from ManualUser to DigitalUser. The flow will be refined later.

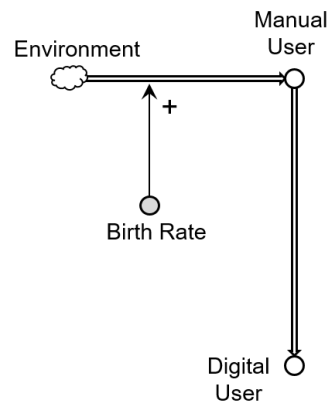


Figure 10. Basic scheme

5.2 Step 2: Adding Education

In this step we add education as a means to stimulate the use of digital services, or, in our terms, stimulate the transition from ManualUser to DigitalUser. The motivation is that having the relevant skills will promote citizens to involve in digital service practices. The conceptual scheme from Figure 1 records the trainings that have been taken by citizens, and the trainings that also have been completed. The relevant scheme fragment is displayed in Figure 11.

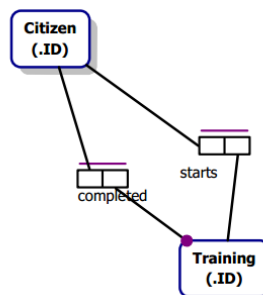


Figure 11. Citizens and training

Citizens first register for a training, then it is recorded when they successfully complete the training. For our purposes, we will focus on the following new states for citizens. First we will consider those manual users who have registered for a training. We refer to such citizens as CitizenStartTraining. After successfully completing a course, such citizens are trained, and ready to become digital user. We refer to such citizens as TrainedCitizen. The flow from ManualUser to DigitalUser from Figure 10 is now redirected via CitizenStartTraining and TrainedCitizen.

We introduce a factor RegistrationRate to control the registration process for trainings (the flow from ManualUser to CitizenStartTraining): the more untrained citizens, the more will be starting a training. Also, the more trainings are being offered, the more citizens will register for a training. However, when the number of trainings offered becomes too big, citizens get confused, and this will negatively influence citizens to enroll in a training.

Furthermore, we assume a factor SuccessRate that controls the probability that a citizen after registering for a training also completes that training successfully. The intended effects of promoting digital service usage are caused by trainings that have been completed. As a

consequence, the quality of trainings (the SuccessRate) is an important factor. This factor can be described as the ratio of starting users who are successful.

This leads to the introduction of the following factors for the extended causal diagram:

1. CitizenStartTraining IS Citizen start Training.
2. RegistrationRate IS COUNT Citizen start Training BUT NOT completed Training / COUNT Citizen BUT NOT completed Training.
3. TrainedCitizen IS Citizen completed Training.
4. SuccessRate IS COUNT TrainedCitizen / COUNT CitizenStartTraining.

From the application domain we obtain the following causal influences. A higher offer of trainings will lead to an increase in subscriptions for trainings. On the other hand, an overload of trainings being offered will also confuse people, and therefore have a negative impact on starting a training. The more trainings, the more people enrolling to trainings. By increasing the success rate of courses, this effect will increase the number of citizen trained. More trained citizen will (1) negatively influence the number of citizens starting as manual user, (2) positively influence starters to become digital users, and (3) encourage the transition from manual to digital service usage. This leads to the ECD as displayed in Figure 12.

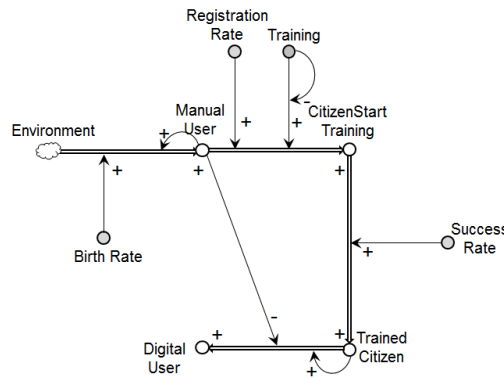


Figure 12. The ECD for step 2

5.3 Step 3: Adding Infrastructure

The next important factor for successful implementation of e-government is the infrastructure that is required for the digital services. We will restrict ourselves in this section to the connectivity by which citizens are connected to the Internet. In Figure 13 we find the relevant concepts for this step.

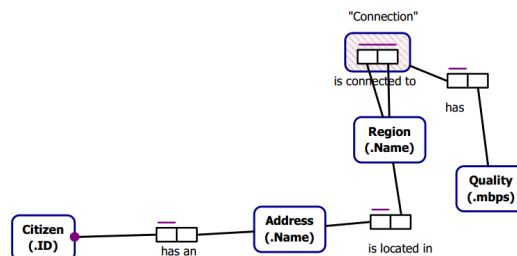


Figure 13. Citizen and infrastructure

From the domain expert we learn that connectivity may be defined as the number of citizens who are connected to the Internet; citizens living in a region that is connected to another region may be considered connected. Reformulating this definition in the conceptual domain language leads to:

Connection IS COUNT Address located in Region being connected to

Furthermore, the domain expert explains that the average connection quality can be used as the overall quality indicator for the network of that country:

ConnectionQuality IS AVERAGE Quality of Connection

These both factors influence manual and digital users and their transition as indicated in Figure 14.

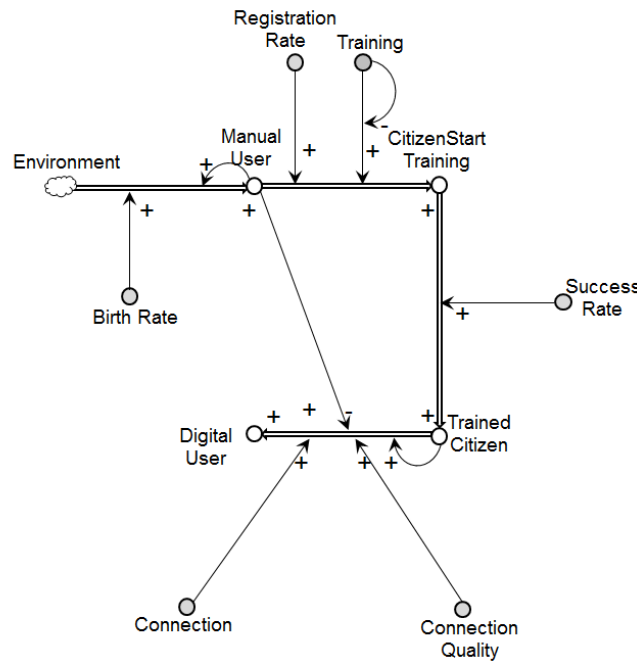


Figure 14. Extended Causal Relation

5.4 The Zanzibar Case

Figure 14 represents study focus on e-government implementation process from Zanzibar case domain. This example highlights important focus for policy makers to apply strategic decision in government services.

Zanzibar is a semi-autonomous Island with a population of 1.3 million [47]. To forecast the development of e-government adoption, citizen behaviour is a core factor to predict changes. Citizen behaviour reflects a proportion of the population in accepting changes especially in ICT adoption. These behaviours were very much accepted and proved as an intrinsic driving force in any adoption process. This article has taken into consideration adoption factors that contribute to the transformation process. The details of these factors have been described in Bass model. The Bass model uses constant coefficient to present the influence of innovation [48].

These factors help to estimate the pessimistic picture of the transformation process from manual service to digital service. They represent the internal and external influence for every transformation step. Figure 14 shows the transformation steps for CitizenStartTraining, TrainedCitizen, and DigitalUsers, where citizen adoption behavior contributes to realize the transformation process. For each transformation step, we need to estimate the number of citizens who get influenced by external communication and those influenced by internal communication. These factors attract wide influence in marketing when used in Bass model. They have been tested for most of the products and fit in a large range of variations [48].

6 Transformation to System Dynamics

Typically, a causal diagram is a first step in the development of an SD model. In the SD model basically the causal relations of the causal diagram are quantified. These quantifications then are the basis for a simulation tool. Such tool helps the policy maker to quickly and conveniently see the effects of their decisions on the target factors by (for instance) manipulating the independent factors. Quantification of the causal relations requires a level of detail and precision that adds an extra complexity to the modeling of the application domain.

In this section we provide a definition of the basic SD constructs, and define the semantics of a SD scheme. A SD scheme is constructed from 2 kinds of variables: (1) stock variables are similar to substantial variables in ECD, and (2) exogenous variables corresponding to dashboard variables in ECD. Sinks and sources are special kinds of stock variables, that represent the environment of the SD scheme. These variables are connected by 2 kinds of arrows. A flow is a connection between stock variables representing the flow of instances, the other connections are called links representing causal relations. Links start in variable (exogenous and stock), and lead to another exogenous variable, or to another flow or link. The source of arrow f is denoted as $\text{src}(f)$ and the destination $\text{dst}(f)$. See Figure 15 for the graphical representations. For our purposes, a SD structure is similar to an ECD except for naming conventions. However, construction rule 5 (see Figure 15) is not possible in SD. We will nevertheless assume this construction rule, and use the technique of flattening (see Lemma 2) repeatedly until no longer possible to transform a construct generated by applying construction rule 5.

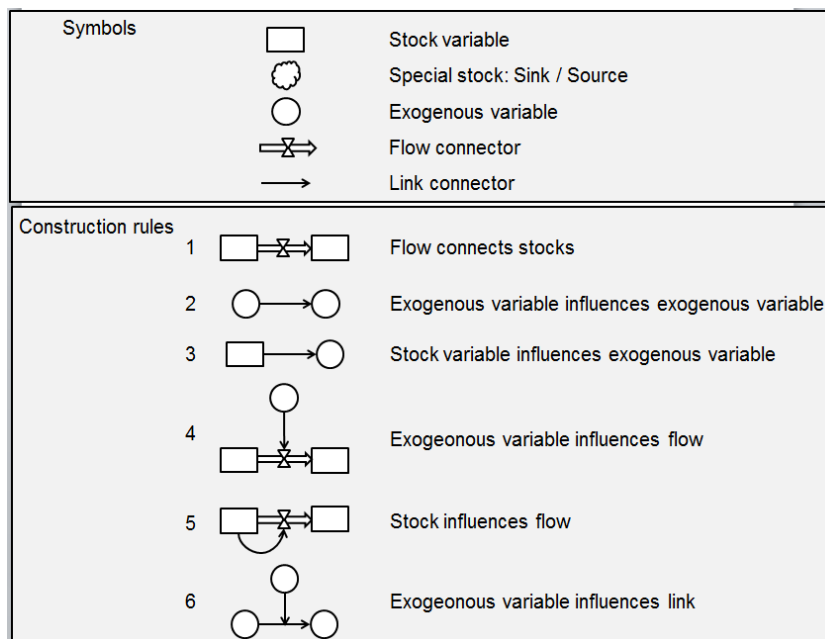



Figure 15. Dynamic modeling constructs

There is a special stock  that represents the environment of the system. Input via a flow from this special stock corresponds to the birth of instances in the system, output to this special stock corresponds to the death of instances. We call a flow external flow if either $\text{src}(f) = \text{cloud}$ or $\text{dst}(f) = \text{cloud}$; this will be denoted as $\text{ext}(f)$. A flow cannot have both $\text{src}(f) = \text{cloud}$ and $\text{dst}(f) = \text{cloud}$.

The main difference between ECD and SD structure is that the SD structure is used to quantitatively describe the behavior of the system by explicitly quantifying the arrows, whereas the ECD restricts itself to a qualitative description of influential aspects. SD assumes time to be discrete rather than continuous, and represents time as a set \mathcal{T} of equidistant points of (real) time.

Let Δt be the size of an elementary (real) time step. For convenience these points in time are consecutively numbered. Then the semantics of the behavior of a SD are described by equations that express how the state of time $t + 1$ is evaluated from the state at time t .

The state of the SD at time t is described by assigning a value $\text{Val}[X](t)$ to all its elements X , both variables and connections. We describe the value function for stock variables and exogenous variables differently.

Stock variables get a special treatment in the description of the behavior semantics (see also Section 4.2). At each point in time t a stock S stores a set $\text{Pop}[S](t)$ of instances of the associated conceptual type. Instances may be moved via a flow to another stock. The value of a stock S is defined as the number of instances $\text{Val}[S](t)$ in the stock. Consequently we have:

$$\Delta \text{Pop}[S](t) = \sum_{f:\text{dst}(f)=S} \text{instances coming from } \text{src}(f) - \sum_{f:\text{dst}(f)=S} \text{instances disappearing via } \text{src}(f)$$

In terms of the value function:

$$\nabla \text{Val}[S](t + 1) = \sum_{f:\text{dst}(f)=S} \text{Val}[f](t) - \sum_{f:\text{src}(f)=S} \text{Val}[f](t)$$

where $\nabla \text{Val}[S](t + 1) = \text{Val}[S](t + 1) - \text{Val}[S](t)$ is the increment of the value function at time $t + 1$, and $\text{Val}[F](t)$ the number of *instances coming from flow f* at point of time t . As a consequence, the change of total number of instances at each point of time equals the number of new instances (of any type) minus the number of disappearing instances (of any type).

Lemma 1. *Conservation of instances*

$$\sum_{\text{Stock } S} \nabla \text{Val}[S](t + 1) = \nabla \text{Val}[\text{☁}](t + 1)$$

Proof. By applying the definition and reversing the summation order, we get:

$$\begin{aligned} \sum_{\text{Stock } S} \nabla \text{Val}[S](t + 1) &= \sum_{\text{Stock } S} \left(\sum_{f:\text{dst}(f)=S} \text{Val}[f](t) - \sum_{f:\text{src}(f)=S} \text{Val}[f](t) \right) \\ &= \sum_{f:\neg \text{ext}(f)} (\text{Val}[f](t) - \text{Val}[f](t)) \\ &\quad + \sum_{f:\text{src}(f)=\text{☁}} \text{Val}[\text{☁}](t) - \sum_{f:\text{dst}(f)=\text{☁}} \text{Val}[\text{☁}](t) \\ &= \sum_{f:\text{src}(f)=\text{☁}} \text{Val}[\text{☁}](t) - \sum_{f:\text{dst}(f)=\text{☁}} \text{Val}[\text{☁}](t) \\ &= \nabla \text{Val}[\text{☁}](t + 1) \end{aligned}$$

since links that are not internal correspond to an internal move of instance, and therefore have no contribution to a change in the number of instances in the system.

The tap symbol used in the graphical representation of a flow (see Figure 15) indicates a regulation that controls this flow. The regulating effect of flow f is influenced by its incoming links (if any).

If link f itself has no incoming links, then its value is simply defined as its base value: $\text{Val}[f](t) = \text{Val}[\text{src}(f)](t)$. Let g be an incoming link for f , or, $\text{dst}(g) = f$, then this base value

$Val[src(f)](t)$ will be influenced by g . We use the pair $g:Val[g](t)$ to denote this influential factor. Let $Inputs[f](t)$ be the set of all influential factors for link f at time t :

$$Inputs[f](t) = \bigcup_{g \rightarrow f} g : Val[g](t)$$

The base value and the influential factors of the incoming links of f are aggregated by the aggregation function $aggr[f]$ associated with f to obtain the resulting value $Val[dst(f)](t)$:

$$Val[f](t+1) = aggr[f](Val[src(f)](t), Inputs[f](t))$$

Similarly, we assume an aggregation function $aggr[E]$ for incoming links of exogenous variable E . Then we have:

$$Val[E](t+1) = aggr[E](Val[E](t), Inputs[E](t))$$

where $Inputs[E](t)$ is the extension of the inputs-function to exogenous variables. In the remainder of this article we will use $X(t)$ as a shorthand notation for $Val[X](t)$. Furthermore, aggregation functions will assume some ordering of the incoming links, allowing to simplify the definition of aggregate functions.

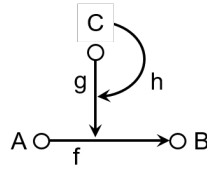


Figure 16. Sample scheme

Example 1. Consider the sample scheme fragment in Figure 16, consisting of the variables A , B and C and the links f , g and h with the following aggregation functions:

$$\begin{aligned} aggr[f](x, y) &= x \cdot e^{-y} \\ aggr[g](x, y) &= x \cdot \ln(y) \end{aligned}$$

Then we get:

$$\begin{aligned} f(t+1) &= aggr[f](A(t), g(t)) = A(t) \cdot e^{-g(t)} \\ g(t+1) &= aggr[g](C(t), h(t)) = C(t) \cdot \ln(h(t)) \\ h(t+1) &= C(t) \end{aligned}$$

Finally we consider exogenous variables E without incoming links. Such variables are sensors that obtain their value from an inspection $insp(E)$ of the associated application domain. Therefore, for such variables we have $Val[E](t) = insp[E](t)$.

Next we consider the flattening procedure as described in Section 4.2. The semantics of the links in Figure 3(a) are described by:

$$\begin{aligned} B(t+1) &= f(t) \\ f(t+1) &= aggr[f](A(t), g(t)) \end{aligned}$$

The semantics for Figure 3(b) are:

$$B(t+1) = aggr[B](f(t), g(t))$$

So the fragments from Figures 3(a) and 3(b) are equivalent if the function $aggr[f]$ from Figure 3(a) is identical to $aggr[B]$ from Figure 3(b). Consequently we have the following lemma:

Lemma 2. (Flattening) *The fragments $C \xrightarrow{*} A \xrightarrow{*} B$ from Figure 3(a) and $C \xrightarrow{*} B; A \rightarrow B$ from Figure 3(b) are equivalent.*

This lemma shows that the use of construction rule 6 in Figure 15 can be avoided.

7 A Simulation Tool for Policy Makers

Using the conceptual scheme from Figure 1 as a framework for reasoning about the application domain, we defined in Section 5 the relevant factors for our focus of attention, and their causal relations. This resulted in the Extended Causal Diagram displayed in Figure 14. In Section 4.5 we gave an impression how an ECD can be used to do qualitative reasoning.

In this section we extend the ECD with a discretization approach and quantitative rules as described in Section 6, leading to a quantitative description of the application domain via a set of recurrence relations. Specific tools have been developed for such SD models. Usually these tools differ (slightly) in the way that the model is graphically represented. Some well-known platforms are Vensim [49] and Sysdea [50]. In this article we will use the Sysdea online platform. Sysdea is a commercial software for modeling how items change over time. This software enhances the overall functions of the system and manages the continuing performance of significant changes over periods of time. The software can display a time chart for each item in the model, which makes it easy to quickly visualize the model behavior.

7.1 The Transformation

Figure 17 shows the transformation rules that transform ECD scheme as shown in Figure 14 to SD scheme in Figure 18. The transformation aims to quantify the ECD diagram using SD techniques. In general, the quantification process has opened a way to simulate the behavior of the system components on a time basis. The process intends to simplify administration process and enhance decision makers to better understand the behaviour of system influences. Also, the realization of ECD scheme into simulation model measures the dynamic feature that need to be monitored in e-government implementation. The transformation process results in the system dynamic scheme in web application simulation tool of Sysdea (see Figure 19).

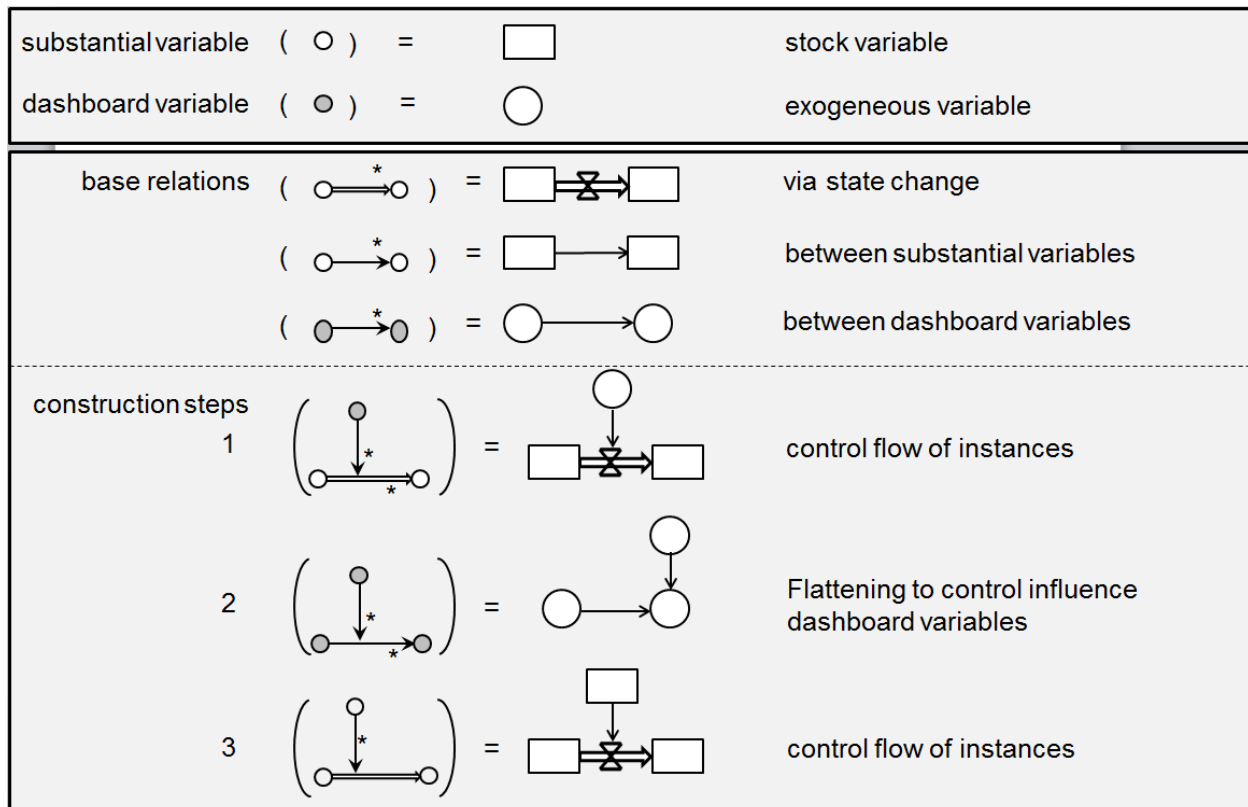


Figure 17. The transformation rules

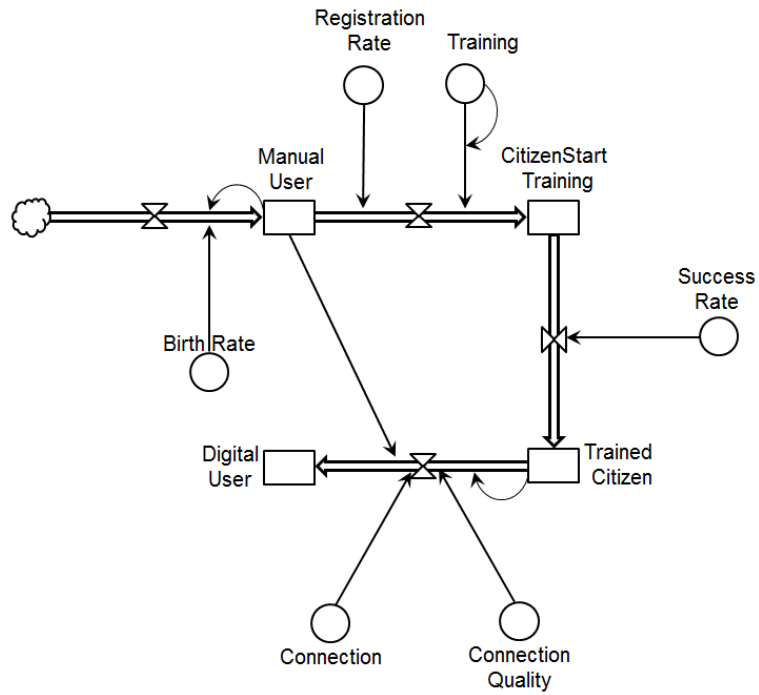


Figure 18. The e-government SD

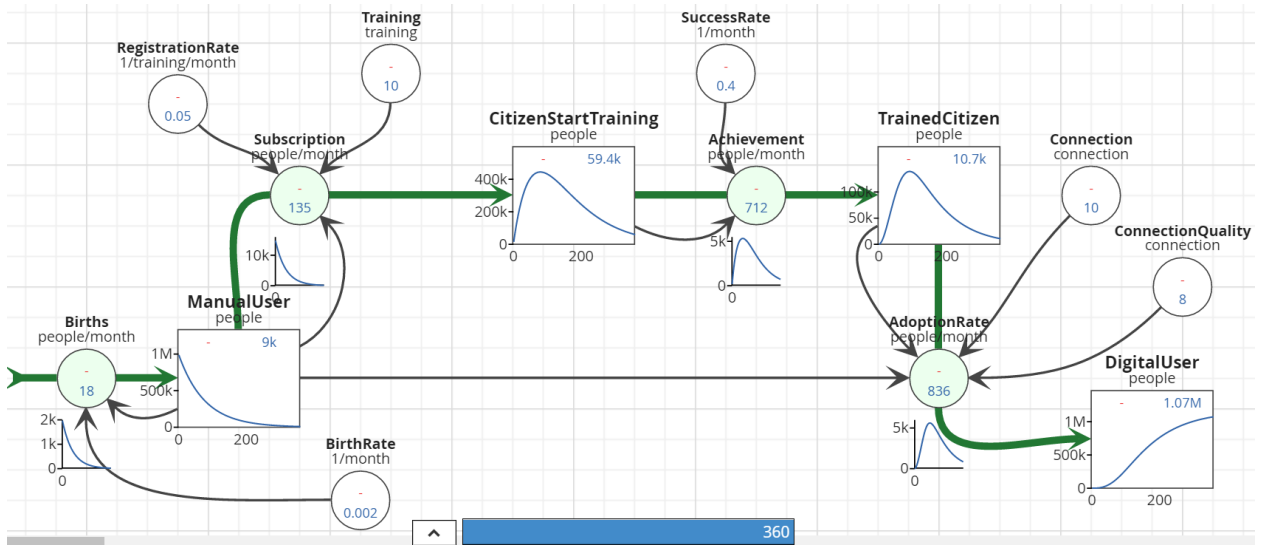


Figure 19. E-government model in SD

7.2 Adding a Time Step and Quantification Rules

First we focus on the discretization scale. A fair choice seems to use 1 month for each time step. Smaller time step leads to a lower systematic error caused by the approximation of a continuous process by a discrete process. However, a smaller time step also leads to more calculations to consider some period of time. This has an impact on the propagation of errors caused by using an imperfect representation of real numbers.

The quantification rules are straightforward estimates of the intended effect. For instance, the factor Births is influenced by the BirthRate (x) and ManualUser (y) and calculated as the fraction x of the current population size y :

$$\text{aggr}[\text{Births}](x, y) = x \cdot y$$

7.3 Experience With the Simulation Tool

The online use of Sysdea platform gives an opportunity to understand system behavior easily. The online simulation enhances the exploration behavior, where each variable can be visualized independently. It allows observing the changes that influence the entire model. Another feature is the sharing property. The model can be shared and modified with other online users without affecting the original model. This feature motivates users to run the model without fear of software installation.

The model itself has adopted the incremental behavior described in ECD construction rules. The simulation results have quantified the variable influences defined in Section 5, especially the influences between target variables and influence variables. Changes of target variables have direct influence on the final results. This feature helps to create awareness to the target variables to meet a long term plan.

Still the Sysdea platform has some limitations. Some features have not yet been implemented. Nevertheless, the flattening process (see Lemma 2) was a perfect approach to align an ECD model to SD modeling.

8 Conclusion and Future Work

The method developed in this study considers domain study as an initial part to understand any system. The domain understanding provides an opportunity to realize the existing problems and find appropriate solutions. The process deploys analysis, assessing and identifying of key components to understand the interconnection, causality and mutual interactions of the system. The method can support policy makers to support strategic decisions in those cases where adequate solutions are hard to design.

The introduction of Extended Causal Diagram (ECD) aims to increase the expressive power of Causal Diagrams by adding 2nd order effects and proposes extra dimensions speed and strength of a causal effect. Using the introduction of e-government in Zanzibar as a non-trivial example, we showcased how to derive a simulation tool in a systematic way from a linguistic-based conceptual description of the application domain.

The construction of an ECD diagram does not concentrate on loop formation, instead it is striving on understanding existing requirements and to transform them into an implementation model. This process transforms domain requirements into a tool that can help decision makers into a strategic decision. Furthermore, the introduction of influence variables in ECD diagrams provides an opportunity to analyse the case domain without further quantification method.

The causal relations that were identified provide a clear perspective on the predicted causes on system implementation. The formalization of this process leads to a more thorough and deep understanding of the designed system and implementation process. This process provides a full

range of relevant variables that we use in policy decision. The decision may cause policy resistance if made incorrectly. This approach provides a quick solution on qualitative perspective. We believe that the formalization of this process will validate this approach and bring a great contribution to the understanding of causal relations in system implementation. E-government implementation system was used as an example, however, we assume that the method can be applied in any complex system.

Extended Causal Diagrams still lack of fine tuning elements. For instance, there is no way to make comparisons between the strength of different causal relations. However, a policy maker may want to choose for the policy with the strongest effect. Also, the speed of the effect is beyond the expressive power of ECD reasoning. This is problematic when different causal relations need very different time periods to become effective. For instance, when a reinforcing and balancing loop intersect, then the policymaker can use the reinforcing loop to start a development and the balancing loop to stabilize at the required level. But the situation when the balancing loop is instantaneous while the reinforcing loop is a long term activity, should be treated very differently from the situation where the balancing loop is a long term and the reinforcement is instantaneous.

References

- [1] T. Andrew, D. Petkov, "The need for a systems thinking approach to the planning of rural telecommunications infrastructure", *Telecommunications Policy*, vol. 27, no. 1, pp. 75–93, 2003. Available: [https://doi.org/10.1016/S0308-5961\(02\)00095-2](https://doi.org/10.1016/S0308-5961(02)00095-2).
- [2] B. Richmond, S. Peterson, "An introduction to systems thinking", High Performance Systems., Incorporated, 2001.
- [3] E. F. Wolstenholme, "Qualitative vs quantitative modelling: the evolving balance", *Journal of the Operational Research Society*, vol 50, no 4, pp. 422–428, 1999. Available: <https://doi.org/10.1057/palgrave.jors.2600700>.
- [4] G. P. Richardson, "Introduction to System Dynamics", in: *Encyclopedia of operations research and management science*, Springer, pp. 807–810, 2001. Available: https://doi.org/10.1007/978-1-4614-9152-1_1.
- [5] System Dynamics Society, "Introduction to System Dynamics", accessed: March 2017. Available: <http://www.systemdynamics.org/what-is-s/>.
- [6] M. Saleh, R. Oliva, C. E. Kampmann, P. I. Davidsen, "A comprehensive analytical approach for policy analysis of system dynamics models", *European Journal of Operational Research* vol. 203, no 3, pp. 673–683, 2010. Available: <https://doi.org/10.1016/j.ejor.2009.09.016>.
- [7] L. Schmid, J. Gallati, K. Hügel, M. Loher, "Success dynamics-a concept for building system dynamics models as decision support within strategic management", the 30th International Conference of the System Dynamics Society, Tutorial Lectures, 2008.
- [8] A. t. Hofstede, H. Proper, T. v. d. Weide, "Formal definition of a conceptual language for the description and manipulation of information models", *Information Systems* vol. 18, no. 7, pp. 489–523, 1993. Available: [https://doi.org/10.1016/0306-4379\(93\)90004-k](https://doi.org/10.1016/0306-4379(93)90004-k).
- [9] T. Halpin, M. Curland, "Automated Verbalization for ORM2", in: *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, Springer LNCS, Heidelberg, vol. 4278, pp. 1181–1190, 2006. Available: https://doi.org/10.1007/11915072_21.
- [10] M. Huth, M. Ryan, "Logic in Computer Science: Modelling and reasoning about systems", Cambridge University Press, 2004. Available: <https://doi.org/10.1017/CBO9780511810275>.

- [11] G. Barthe, J. Forest, D. Pichardie, V. Rusu, “Defining and reasoning about recursive functions: a practical tool for the Coq proof assistant”, in: *International Symposium on Functional and Logic Programming*, Springer, pp. 114–129, 2006. Available: https://doi.org/10.1007/11737414_9.
- [12] D. Gentner, A. L. Stevens, “*Mental models*”, Psychology Press, 2014.
- [13] J. K. Doyle, D. N. Ford, “Mental models concepts revisited: some clarifications and a reply to Lane”, *System Dynamics Review*, vol. 15, no. 4, pp. 411, 1999. Available: [https://doi.org/10.1002/\(sici\)1099-1727\(199924\)15:4<411::aid-sdr181>3.0.co;2-r](https://doi.org/10.1002/(sici)1099-1727(199924)15:4<411::aid-sdr181>3.0.co;2-r).
- [14] D. C. Lane, “Diagramming Conventions in System Dynamics”, *The Journal of the Operational Research Society*, vol. 51, no. 2, pp. 241–245, 2000. Available: <http://www.jstor.org/stable/254265>.
- [15] M. Schaffernicht, “Causality and diagrams for system dynamics”, in: *Actas de la 50th International Conference of the System Dynamics Society*, Boston, pp. 147–154, 2007.
- [16] J. Pearl, “Causal diagrams for empirical research”, *Biometrika*, vol. 82, no. 4, pp. 669–688, 1995. Available: <https://doi.org/10.1093/biomet/82.4.702>.
- [17] J. Homer, R. Oliva, “Maps and models in system dynamics: a response to Coyle”, *System Dynamics Review*, vol. 17, no. 4, pp. 347–355, 2001. Available: <https://doi.org/10.1002/sdr.224>.
- [18] J. Grimmer, “We are all social scientists now: how big data, machine learning, and causal inference work together”, *PS: Political Science & Politics*, vol. 48, no. 1, pp. 80–83, 2014. Available: <https://doi.org/10.1017/s1049096514001784>.
- [19] J. R. Burns, “Simplified translation of CLDs into SFDs”, proceedings of the International Conference of the System Dynamics Society, 2001.
- [20] G. P. Richardson, “Problems in causal loop diagrams revisited”, *System Dynamics Review*, vol. 13, no. 3, pp. 247–252, 1997. Available: [https://doi.org/10.1002/\(sici\)1099-1727\(199723\)13:3<247::aid-sdr128>3.0.co;2-9](https://doi.org/10.1002/(sici)1099-1727(199723)13:3<247::aid-sdr128>3.0.co;2-9).
- [21] S. Rebich, C. Gautier, “Concept Mapping to Reveal Prior Knowledge and Conceptual Change in a Mock Summit Course on Global Climate Change”, *Journal of Geoscience Education*, vol. 53, no. 4, pp. 355–365, 2005. Available: <https://doi.org/10.5408/1089-9995-53.4.355>.
- [22] K. Weinerth, V. Koenig, M. Brunner, R. Martin, “Concept Maps: A Useful and Usable Tool for Computer-based Knowledge Assessment? A Literature Review with a Focus on Usability”, *Comput. Educ.*, vol. 78, pp. 201–209, 2014. Available: <https://doi.org/10.1016/j.compedu.2014.06.002>.
- [23] F. Ackermann, C. Eden, S. Cropper, “Getting started with cognitive mapping”, Banxia Software, 1992.
- [24] F. Ackermann, J. Alexander, “Researching complex projects: Using causal mapping to take a systems perspective”, *International Journal of Project Management*, vol. 34, no. 6, pp. 891–901, 2016. Available: <https://doi.org/10.1016/j.ijproman.2016.04.001>.
- [25] J. Bryson, F. Ackermann, C. Eden, C. Finn, “Visible Thinking: Unlocking Causal Mapping for Practical Business Results”, Wiley, 2004.
- [26] S. Nadkarni, P. P. Shenoy, “A causal mapping approach to constructing Bayesian networks”, *Decision Support Systems*, vol. 38, no. 2, pp. 259–281, 2004. Available: [https://doi.org/10.1016/s0167-9236\(03\)00095-2](https://doi.org/10.1016/s0167-9236(03)00095-2).
- [27] J. B. Homer, G. B. Hirsch, “System dynamics modeling for public health: background and opportunities”, *American journal of public health*, vol. 96, no. 3, pp. 452–458, 2006. Available: <https://doi.org/10.2105/ajph.2005.062059>.

- [28] S. Minegishi, D. Thiel, “System dynamics modeling and simulation of a particular food supply chain”, *Simulation Practice and Theory, Modelling and Simulation of Complex Production Systems: Selected Papers from the 2nd French Conference on {MOdelling} and {SIMulation}*, vol. 8, no. 5, pp. 321–339, 2000. Available: [https://doi.org/10.1016/s0928-4869\(00\)00026-4](https://doi.org/10.1016/s0928-4869(00)00026-4).
- [29] F. F. Tulinayo, P. P. van Bommel, H. E. Proper, “Grounded System Dynamics: A Procedure for Underpinning System Dynamics with a Domain Modeling Method”, in: *IFIP Working Conference on The Practice of Enterprise Modeling*, Springer, pp. 112–125,, 2011. Available: https://doi.org/10.1007/978-3-642-24849-8_9.
- [30] S. Howick, F. Ackermann, D. Andersen, “Linking event thinking with structural thinking: methods to improve client value in projects”, *System Dynamics Review*, vol. 22, no. 2, pp. 113–140, 2006. Available: <https://doi.org/10.1002/sdr.332>.
- [31] D. L. Moody, “Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions”, *Data & Knowledge Engineering*, vol. 55, no. 3, pp. 243–276, 2005. Available: <https://doi.org/10.1016/j.datak.2004.12.005>.
- [32] J. P. Zwart, M. Engelbart, S. Hoppenbrouwers, “Fact Oriented Modeling with FCO-IM: Capturing Business Semantics in Data Models with Fully Communication Oriented Information Modeling”, Technics Publications, 2015.
- [33] G. M. Nijssen, T. A. Halpin, “Conceptual Schema and Relational Database Design: a fact oriented approach”, Prentice-Hall, Inc., 1989.
- [34] P. Frederiks, T. van der Weide, “Information modeling: The process and the required competencies of its participants”, *Data and Knowledge Engineering*, vol. 58, no. 1, pp. 4–20, 2006. Available:https://doi.org/10.1007/978-3-540-27779-8_11.
- [35] K. Popper, “Philosophy of science”, *British Philosophy in the Mid-Century* (ed. CA Mace). London: George Allen and Unwin, 1957.
- [36] N. E. Fuchs, U. Schwertel, R. Schwitter, “Attempto Controlled English—not just another logic specification language”, in: *International Workshop on Logic Programming Synthesis and Transformation*, Springer, pp. 1–20, 1998.
- [37] M. Khamis, T. van der Weide, “Conceptual Framework for Sustainable e-Government Implementation in low Infrastructure Situation”, in: *ECEG2016-Proceedings of 16th European Conference on e-Government ECEG*, Academic Conferences and publishing limited, pp. 283, 2016.
- [38] T. Halpin, “Object-Role Modeling Fundamentals: A Practical Guide to Data Modeling with ORM”, Technics Publications, LLC, USA, 2015.
- [39] K. Evans, “Requirements Engineering with ORM”, in: *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops: OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, GADA, MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2005*, Agia Napa, Cyprus, October 31 - November 4, 2005. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 646–655, 2005.
- [40] D. M. West, “E-government and the transformation of service delivery and citizen attitudes”, *Public administration review*, vol. 64, no. 1, pp. 15–27, 2004. Available: <https://doi.org/10.1111/j.1540-6210.2004.00343.x>.
- [41] P. F. Tulinayo, S. S. Hoppenbrouwers, H. E. Proper, “Integrating system dynamics with object-role modeling”, in: *IFIP Working Conference on The Practice of Enterprise Modeling*, Springer, pp. 77–85, 2008. Available: https://doi.org/10.1007/978-3-540-89218-2_6.

- [42] T. van der Weide, F. Tulinayo, P. van Bommel, “Static and Dynamic Aspects of Application Domains: An Inductively Defined Modeling Technique That Allows Decomposition”, *Complex Systems Informatics and Modeling Quarterly*, vol. 7, pp. 25–50, 2016. Available: <https://doi.org/10.7250/csimq.2016-7.02>.
- [43] H. Sayama, et al., “Introduction to the modeling and analysis of complex systems”, Open SUNY Textbooks, Milne Library, 2015.
- [44] J. D. Sterman, “Business dynamics: systems thinking and modeling for a complex world”, HD30. 2 S7835 2000, McGraw-Hill, 2000.
- [45] J. Pearl, “Causality”, Cambridge university press, 2009.
- [46] J. E. Cohen, “Population growth and earth’s human carrying capacity”, *Science*, vol. 269, no. 5222, pp. 341, 1995. Available: <https://doi.org/10.1126/science.7618100>.
- [47] BBC News, “Zanzibar Profile”, accessed: April, 2017. Available: <http://www.bbc.com/news/world-africa-14115176>.
- [48] F. M. Bass, “A new product growth for model consumer durables”, *Management science*, vol. 15, no. 5, pp. 215–227, 1969. Available: <https://doi.org/10.1287/mnsc.15.5.215>.
- [49] R. Eberlein, “Vensim User’s Guide (version 5)”, Harvard, MA.: Ventana Systems, 2007.
- [50] K. Warren, “Sysdea User Guide. Building Dynamics Business Model”, Strategy Dynamics Ltd., 2016.

Additional information about the article:

Author’s ORCID iD	M.M. Khamis – n/a on 26 July 2017 Th.P. van der Weide - www.orcid.org/0000-0002-6517-2079
Article history	Received 25 April 2017 Accepted 10 July 2017 Available online 31 July 2017
Article PII	S225599221400066X