From a System Dynamics Causal Loop Diagram to an Object-Role Model: A stepwise Approach

F.P Fiona Tulinayo[1], P. Patrick van Bommel[1], H.A Erik Proper[1,2]
[1]Institute of Computing and Information Sciences
Radboud University Nijmegen
Heyendaalseweg 135
6525 AJ Nijmegen, The Netherlands
[2]Public Research Centre Henri Tudor
29 avenue John F. Kennedy
Luxembourg-Kirchberg, Luxembourg
F.Tulinayo@science.ru.nl, pvb@cs.ru.nl, e.proper@acm.org

Journal of Digital Information Management

ABSTRACT: *Two diagrams are most commonly used in System Dynamics (SD); Causal Loop Diagrams (CLDs) and Stock and Flow Diagrams (SFDs). CLDs are qualitative in nature while SFDs are quantitative. In this paper we draw our attention to CLDs. CLDs are visual representation of dynamic influences with inter-relationships amongst a collection of SD variables. They are used to qualitatively capture structures and interactions of feedback loops. CLDs however, have been found to be ambiguous, lack detail and not easy to conceptualize. In this paper therefore, we come up with steps on how to transfer information from a CLD into an Object Role Modeling (ORM) model. We carry out this study to improve CLD model conceptualization, make CLD variables explicit and to reduce on CLD model ambiguity. To achieve this we use ORM, a fact oriented method whose focus is on domain conceptualization through data modeling because of its conceptual focus and roots in verbalization, graphical expressiveness and well defined semantics. To apply and validate these steps, we use a case as an example in focus session setting. From these focus sessions, evaluations and discussion of results are made and finally conclusions are drawn.*

Categories and Subject Descriptors
I.3.5 Computational Geometry and Object Modeling: **I.1.3** Languages and Systems

General Terms: System Dynamics, Causal loop diagrams

Keywords: Object-Role Modeling, System Dynamics, Causal Loop Diagram, Object role model

Received: 11 January 2012, Revised 2 March 2012, Accepted 13 March 2012

1. Introduction

Understanding the model and understanding the problem are the key products that should be accomplished in system dynamics modeling [26]. To attain this, two diagrams are most commonly used in system dynamics; CLDs and SFDs. SFDs take a quantitative aspect where sets of equations are input into the stock and flow diagram resulting into simulations while CLDs are used qualitatively to capture the structure and interactions of feedback loops [41]. In this paper however, we draw our attention to CLDs which are a visual representation of dynamic influences with interrelationships amongst a collection of SD variables [30]. The dynamics in SD arises from interaction of two types of feedback loops, positive and negative. Positive loops tend to reinforce or amplify whatever is happening in the system while negative loops counteract and oppose change. These loops all describe processes that tend to be self limiting, create balance and equilibrium and are determined by the causal link polarities [19,23]. The polarity of each feedback loop is a vital part in understanding model behavior. The addition of a time delay to a negative feedback loop may change the behavior of the model say from goal-seeking to oscillating. This indicates that the behavior of an information feedback loop is highly sensitive to the kind of information used to make decisions and the accuracy of that information. Although CLDs visually represent the dynamic influences among SD variables, they have been found to have a number of issues among which is ambiguity and lack of detail [1,20,21,22,25,29]. These make transformation of CLD variables into SFD hard [21,25,29]. In [25] Richardson urges for support in understanding the links between SFD and dynamic behavior and particularly identifies formal conceptualization and model understanding as the most difficult. This therefore indicates that SD models need to be backed with methods that are better at conceptualizing problem domain(s). In our study we introduce the use of a fact oriented modeling method ORM, whose focus is on domain conceptualization through data modeling [10] into the process of transforming a CLD into an SFD[1]. The addition of ORM to this process will not only improve SD model conceptualization but also make input data reusable and transferable (from one system to another or from one organization to another). In this paper however, we limit our study to transforming a CLD into an ORM model.

ORM as a method was designed for modeling and querying an information system at the conceptual level, and mapping between conceptual and logical levels [12]. It takes a static perspective on the domain in the sense that it aims to capture the fact types and entity types that play a role in the (dynamic) domain, while SD takes a dynamic perspective in which the dynamic behavior of the domain is captured. One of the important features of ORM is its foundation in natural language analysis which makes it resistant to changes that cause attributes to be remodeled as object types or relationships [15]. ORM is not only used as a database modeling methodology but also as a graphical notation in XML-schemes [2] and is applied to a number of areas like requirements engineering [6], business rules [9] etc. In this paper however, we join scholars like [15,16,31,32] who use ORM to model domain ontologies. We use ORM here because of its strong verbalization and conceptualization facility and for its fully formal link to predicate logic. More details about ORM can be found in [11].

In our previous studies we looked at conceptual links between SD and ORM [37] mapped ORM and SD elements [38], investigated the update behavior of SD and ORM [36] and studied the transformation of an ORM model into a SFD [39]. In this paper however, we aim at showing how variables in a CLD can be transferred into ORM concepts leading to an ORM model. This study allows us transfer information from a CLD to an ORM model in a systematic manner.

The rest of the paper proceeds as follows; in section 2 we give a brief introduction to a System Dynamics CLD. In section 3 we give steps which we follow while transforming a CLD into an ORM model. In section 4 we apply the steps given in section 3 to a case in focus session environment, in section 5 we present contributions and challenges met during focus group session. In section 6 we draw conclusions.

2. CLD Brief Introduction

A CLD is made up of *variables, signs* (either a positive or nega-tive) and *causal links* with arrows (these arrows represent the causal influence). The arrows are drawn in a circulsar manner indicating the cause and effect lead-ing to a feedback loop which is a closed sequence of cause and effects (that is a closed path of action and information) [26].

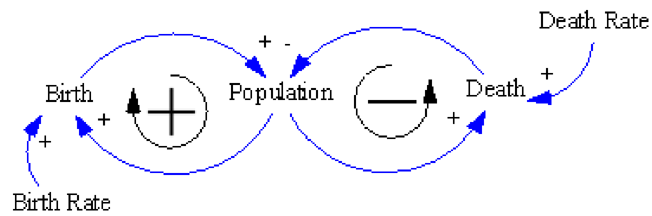


Figure 1. A Population Causal Loop Diagram

Figure 1 is an annotated CLD for a simple population. It includes elements and arrows (sometimes referred to as *causal links*) but also includes signs (positive or negative) on each link. These signs have the following meanings:

• A causal link from element *birth* to element *population* is *positive*. This is because *births* add to the *population* that is a change in birth produces a change in *population* in the *same* direction.

• A causal link from element *death* to element *population* is *negative*. This is because *death* subtracts from population or a change in death produces a change in *population* in an *opposite* direction.

In addition to the signs on each link, is a complete loop, this loop has either a positive or negative sign. The sign for a particular loop is determined by counting the number of minus (-) signs on all the links that make up the loop.

• A feedback loop is called positive (R), indicated by a plus sign in parentheses, if it contains an even number of negative causal links. *See left side of figure 1.*

• A feedback loop is called negative (B), indicated by a minus sign in paren-theses, if it contains an odd number of negative causal links. *See right side of figure 1*.

Thus, the sign of a loop is the algebraic product of the signs of its links. Often a small looping arrow is drawn around the feedback loop sign to more clearly indi-cate that the sign refers to the loop. Further explanation on how CLD influences operate can be found in [29,33].

3. Transforming a CLD into an ORM

To enable us transform CLD variables into an ORM model, we came up with the following steps;

1. Identify object(s) in each CLD variable.
2. Collect and group (Classify) similar objects.
3. Identify and connect roles played by objects to their object types.
4. Add all constraints and mandatory roles.
5. Merge the models.

The steps given in this paper have some similarities with ORM's seven Concep-tual Schema Design Procedure

[1]In this paper we limit ourselves to transforming a CLD into an ORM model, that is why we do not show how we move from ORM to SFD but part of this can be found in [39].

(CSDP) steps except here we focus on each CLD variable and not each model component (a section of a model) as is the case with ORM's CSDP. The CLD to ORM transformation steps look at each vari-able as a unique entity. To cross check the quality of the derived ORM model, we recommend that ORM's CSDP steps be considered. Secondly, for complex CLD models, we suggest the use of summarized top-down approach (three steps) as described in [11]. After outlining the CLD to ORM transformation steps, we now explain each step in detail with examples.

[*Step 1:*] *Identify object(s) in each CLD variable.*
Objects are things of interest (populations) in an object or value types. To identify objects in each CLD variable we use the questions below as a guide:

1. Who are the main role player(s) in a given CLD variable?
2. Who plays a role in this CLD variable?

A *role player* in this case is the *object*. By answering one of the questions above, objects taking part in a given CLD variable are identified. Let us use figure 2 as an example to make this more explicit. Figure 2 is made up of the following variables; *Patient arrival, Admitted patients, Patients discharged, Available ad-mission beds and Number of patients in a queue*.

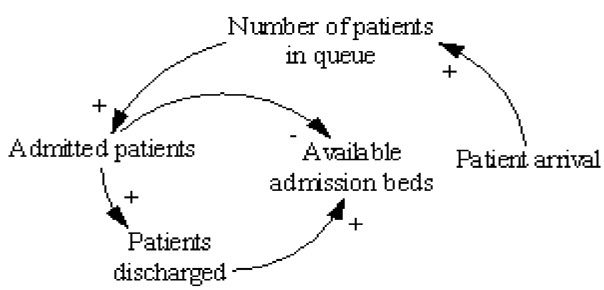


Figure 2. An Example of a Causal Loop Diagram

For us to identify *objects* we use the question(s) provided above as a guide in identifying the objects and results are displayed in table 1. For each CLD vari-able given in figure 2 we present a question with its anticipated answer.

the variable while object *'bed'* is as a result of the influence variable '*Admitted patients'* has with '*Available admission beds*' same applies to question 3. In question 4 object identified is '*Bed*' because it is the only role player. Variable '*Available admission beds'* is influenced by other variables but does not influence any other variable in the loop.

In conclusion, if a variable influences other variables then the objects in the influenced variables are also added to the list of objects in that variable. For example; if variable '*X*' is influenced by variable '*Y*', then objects identified in '' are also identified in variable '' and if variable '' influences variable '', the objects identified in '' are not listed among '' objects. This is because '' does not add to variable '' but subtracts from it.

[*Step 2:*] *Collect and group (Classify) similar objects.*
Here we collect all similar objects. Each classification of objects makes an object type. An object type is a collection of objects with similar properties. It contains objects that play roles and is designated by a solid-line named ellipse with a reference scheme indicated below the object type name. This means that in this step we collect all objects with similar properties from different CLD variables and group them into one. So, using the same example give in step 1 table 1, where we have questions and objects identified for each CLD variable, the types of objects found are two '*Patient*' and '*Bed*'. This means that these objects are contained in two different object types with similar properties. Therefore in this step we derive two object types '*Patient*' and '*Bed*'.

The number of object types in a CLD diagram is equal to the total number of object classifications. Classifications are the different groups of objects with similar properties. Therefore; $Pop\,(O_c) = Pop\,(O_t)$

Where

O_t = *Total number of identified object types*
O_c = *Total number of object classification*

In figure 3 we present CLD variables with their identified object types from each CLD variable. In the same figure,

Number	Question	Identified Object
1.	Who plays a role in CLD variable '*Patient arrival*'?	Patient
2.	Who plays a role in CLD variable '*Admitted patients*'?	Patient, Bed
3.	Who plays a role CLD variable '*Patients discharged*'?	Patient, Bed
4.	Who plays a role in CLD variable '*Available admission beds*'?	Bed
5.	Who plays a role in CLD variable '*Number of patients in queue*'?	Patient

Table 1. Objects for each CLD Variable in Figure 2

In question 1 of table 1 object identified is '*patient*' because (s)he is the only role player same applies to question 5. In question 2 of the same table, we have '*pa-tient*' and '*bed*' as objects. Object 'patient' is derived directly from

explanations as to why those object types are derived are included. For example; *who plays a role in variable 'Admitted patients'?* Our answers are *Patient* and *Bed*. Patient is directly derived from CLD variable '*Ad-mitted*

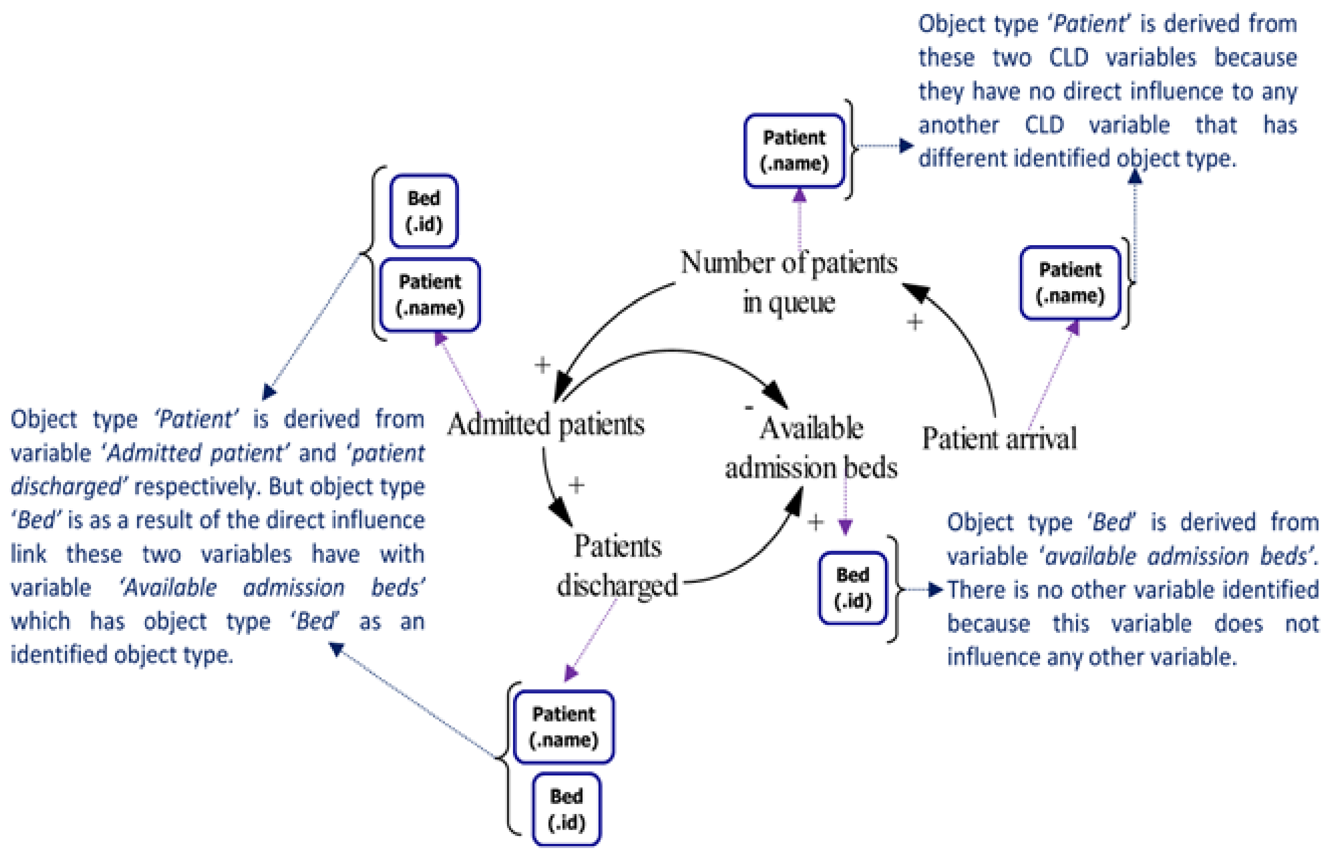


Figure 3. A CLD with Derived Object Types

patients' but 'Bed' is as a result of the influence link it has with variable '*Available admission Beds*'. Notice that, CLD variables that are influenced by other CLD variables but do not influence any other variable in the CLD model, do not have any other object types except those directly derived from them, an example of such variable are '*Available admission Beds*' and '*Patient arrival*'. Other cases are CLD variables that are influenced by other CLD vari-ables and have similar object types. For example '*Number of patients in queue*', since these variables have a similar object type, they therefore maintain object type(s) derived from their particular CLD variable. Secondly, we also note that CLD variable names are in plural form but the ORM object type names are represented in singular form. This is because ORM mainly captures information in oneness but that does not mean that mul-tiple objects cannot be represented. They can but in this paper we will limit ourselves to the simple ORM concepts.

[*Step 3:*] *Identify and connect roles played by objects to their object types.*
In ORM, roles define the relationships between object types. They are repre-sented by boxes with their relationship names and are connected to object types by solid lines. When these roles are put together they make unary, binary, ter-tiary, etc. fact types. Using the same example given in step 1, we now show how roles for each object type are identified. To achieve that, we use the following two guiding questions:

• What role (s) do objects in object type (A) play in object type (A)?

• What role do objects in object type (A) play in object type $A_1, A_2, A_3,..., A_n$?

Where; $A_1, A_2, A_3,..., A_n$ are different object types identified in step 2. Note that, for each object type identified in step 2, there is a role attached to it, *see table 2*. The number of roles played by an object type are determined by the number of times objects in that object type participate (play) in different object types. For every identified role, a connection (link) from that role to the object type it relates with is added. The roles played by objects are explicitly shown as boxes with Predicate names *see ORM representations in table 2*. These predicate names are written beside each role and are read from left to right, or top to bottom. It is through predicates that object types relate to each other.

[*Step 4:*] *Add all constraints and mandatory roles.*
After completing steps 1, 2 and 3, we now add constraints to the model. There are a number of constraints in ORM and can be found in [11]. Constraints make the boundaries of a domain. This step can also be carried out in parallel with step 3 for cases were the CLD model has many variables or is complex. Constraints cannot be obtained from CLD variable directly. Therefore, the modeler validates the ORM model input constraints through verbalizations.

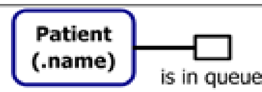
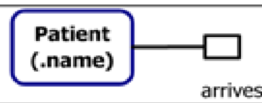
CLD variable name	Object type	Role played	ORM representation	ORM Verbalizations
Admitted patient	Patient, Bed	*is admitted at .... Is for*		Patient is admitted at Bed. **Each** Patient is admitted at **exactly one** Bed. **Each** Bed is for **exactly one** Patient.
Number of patients in queue	Patient	*Is in queue*		Patient is in queue.
Patients discharged	Patient, Bed	*is discharged from... was on*		Patient is discharged from Bed. **Each** Patient is discharged from **exactly one** Bed. **Each** Bed was for **exactly one** Patient.
Available admission beds	Bed	*is available for admission*		Bed is available for admission.
Patients arrival	Patient	*arrives*		Patient arrives.

Table 2. CLD variables, Identified Object types, roles and ORM representation

With Microsoft Visual Modeler 2005, when a constraint is added to the role(s), the readings are generated automatically which makes ORM model validation easier.

[*Step 5:*] *Merge the models.*
In the final step, we merge the models. Merging of these models, makes the ORM model complete as presented in figure 4. If the ORM model has other constraints e.g. irreflexive, ring constraints, occurrence frequency constraints, car-dinality joins etc. they should be added to the model at this step.

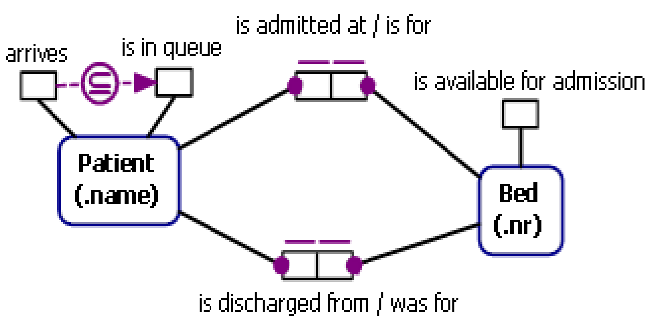


Figure 4. ORM model with all Constraints and Relationships

The verbalizations for figure 4 are as follows;

Patient is admitted at Bed.

Each Patient is admitted at **exactly one** Bed.

Each Bed is for **exactly one** Patient.

Patient is discharged from Bed.

Each Patient is discharged from exactly one Bed.

Each Bed was for **exactly one** Patient.

Patient arrives.

Bed is available for admission.

Patient is in queue.

If some Patient arrives **then that** Patient is in queue.

All in all, transformation of a CLD model into an ORM model is better done incrementally. This prevents the modeler from getting confused in cases where the model boundaries are not explicit or CLD variable names are not clear.

Secondly, we see that in this study we deal with each CLD variable indepen-dently this therefore prompts defining of facts pertaining to each particular CLD variable hence clearing the ambiguities in each CLD variable and improving the SD model conceptualization.

4. Application

To apply the CLD-ORM steps, we use a case “process in the maternity ward at Ugandan hospitals" in focus group sessions. This study works both as an experiment for the application and an evaluation mechanism of the CLD-ORM transformation steps. Case study as defined in [5] is a research strategy which focuses on understanding the dynamics present within a single setting. Case study may involve either single or multiple cases and numerous levels of analysis [42]. It combines data collection methods such as archives, interviews, question-naires and observations. To collect the case data we used observation, archives and interviews. In this section, we start by giving background to the case, then present what transpired in the focus group sessions and finally evaluations and challenges faced during the experiment.

4.1 Case: Intrapartum process in Ugandan hospitals

Intrapartum is the time from the onset of true labor until the delivery of the infant or placenta. In this case study we sampled out three Ugandan hospitals. Selection of these hospitals depended on a number of factors; Location, Availability of Health services, Human resource (doctors and nurses), Size of the hospital, Level (grade) of the hospital etc. We visited all hospitals and the health center (Mukono Health Center, Kawolo Hospital and Mulago Hospital) to observe, note down the process and record details on activities like; doctor monitoring time, patient arrival time, number of patients received per day, activities in the labor ward, archival data and observe patients day to day behaviors. This was done for a period of three month.

The process: A patient comes to the labor ward with her antenatal card from the antenatal clinic. She queues up. Her waiting time depends on a number of factors which are; her arrival time, the number of patients around and number of nurses on duty. When her turn comes, the nurse on duty takes her history and then examines her. This examination takes approximately 30 minutes for Mukono and Kawolo and 15-20 minutes for Mulago. The nurse establishes the patient's dilation stage. If the patient is 4cm dilated, she is admitted to the general ward. She only returns for examination if there is any complication or after 4 hours. During this time, after every 30 minutes monitoring of the labor progress, status of the mother and cervical dilation is done. When the patient is 8cm dilate, she is taken to the delivery room. While there, the nurse monitors descending of the head 2 hourly and the sticker. When the patient has 10cm dilate, she is ready to give birth. After delivery, she is taken back to the general labor ward. Normal delivery patients stay at the labor ward for a maximum period of 24 hours and patients who have had caesarian birth stay for a period of 4-7 days. On discharge, the baby is taken for immunization.

4.2 Focus group Sessions

In this study, we use experimental evaluation in a controlled setting (focus group) as defined in design science methodology [14]. A focus group is defined in [34] as a moderated discussion among six to twelve persons discussing a topic under the direction of a moderator whose role is to promote interaction and keep the discussion on the topic of interest. More about focus groups can be found in [3,18,34]. We use focus group discussions to obtain feedback on CLD to ORM transformation steps. This feedback is used as the basis to refine and improve CLD to ORM transformation steps.

First, we present a summary of the activities that took place in each focus group session in table 5. We conducted three ses-sions with 6 to 8 participants. Prior to conducting these sessions, participants were given reading text explaining ORM and SD constructs and how they can be applied. Provision of this text was to enable participants familiarize with the constructs, and to prepare them for the sessions. Materials used in these sessions included; makers, flip charts and an audio tape. Here we note that on the first and last days all participants were present (8 participant) but on the second day two participants sent in apologies and we could not get replacements. Secondly, the models we provide in this paper have been redrawn because flip charts were not easily transferable to the desktop.

4.2.1 First session

In the first session we used an open format because it allows group members express their views without openly guiding the direction of the discussion [41]. Participants were given approximately 30 min-utes to go through the described case in subsection 4.1, method constructs and ask questions where need be. There-after, they were asked to brainstorm and come up with CLD variables. The identified variables were used to construct a CLD. Questions like; "Why have a CLD to ORM and not a CLD to Stock and flow and then ORM?", "Of what importance is it to introduce ORM into the traditional System Dynamics CLD to SFD procedure?" were asked and addressed by the moderator. As they brainstormed, the moderator listed the identified CLD variables on a flip chart and were as follows; *Arrival time, No of patients available, Number of nurses on duty, Doctors on call, Doctor re-sponse time, Patient history, Dilation (4cm, 8cm, 10cm), Delay between dilation, Monitoring time, Labor ward(s), Patient examination time, General labor ward, Antenatal card, Waiting time, Delivery room, Normal progress, Number of beds in the delivery room, Abnormal progress, Normal delivery, Baby immunization, Patients discharged, Patient admissions, Babies born, Admission beds.* For some of these variables, consensus building was need which wasted a lot of time.

On completing the variable identification process, they embarked on iden-tifying the influences between or among the variables. In figure 6, we present an incomplete CLD model that was derived at the end of the first session.

In conclusion to this session, participants suggested that some of the participants visit one of the hospitals to verify the given case. Two of the participants working closely with the main hospital (Mulago) agreed to do so.

Before the sessions participants were given read text.

Session One Activities

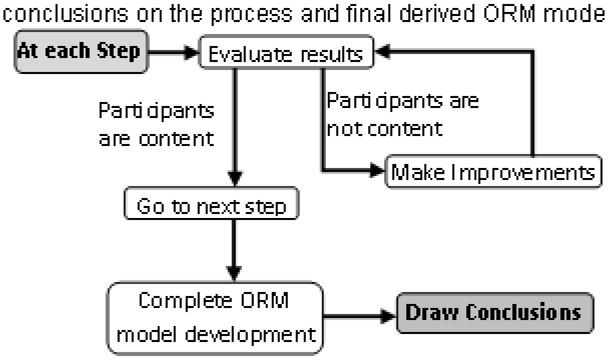
1. **Define research problem**
   Participants were given 30 minutes to familiarize with the given problem and ask questions for clarification.
2. **Identify Variables**
   Participants were asked to identify CLD variables pertaining to the problem. In this phase there was need to build consensus on some of the variables which led to mismanagement of time
3. **Start building the Causal Loop Diagram**
   - Participants identified the cause and effect relationships among CLD variables.
   - Add polarities to the causal links (influence arrows) in the CLD model.

Session Two Activities

- Participants completed the Causal Loop Diagram.
- Participants were asked to take a break of 10-15 minutes.
- After the break, participants were provided with CLD to ORM transformation steps.
- Moderator explained the steps as participants applied them.
- On completion of step1, participants proceeded with steps 2 and partially step 3.

Session Three Activities

- Participants were asked to complete step 3 before proceeding to steps 4 and 5
- At each step, the upshot was criticized by participants.
- On completion, conclusions on the process and final derived ORM model were drawn.



The figure above shows a summary of the CLD-ORM steps execution process

Table 5. A Summary of the CLD-ORM Focus Session Execution Process

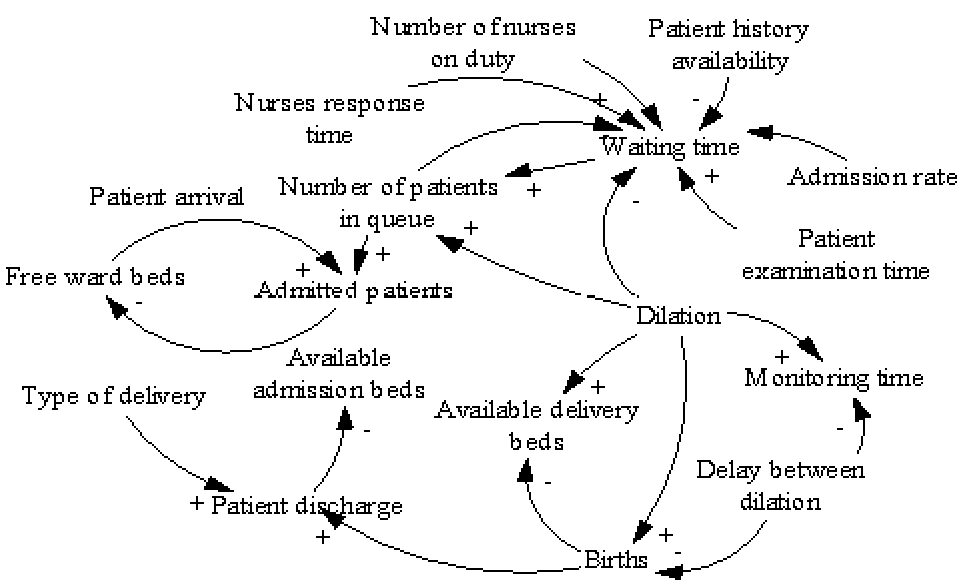


Figure 6. A Partial Causal Loop Diagram that Resulted from Session One

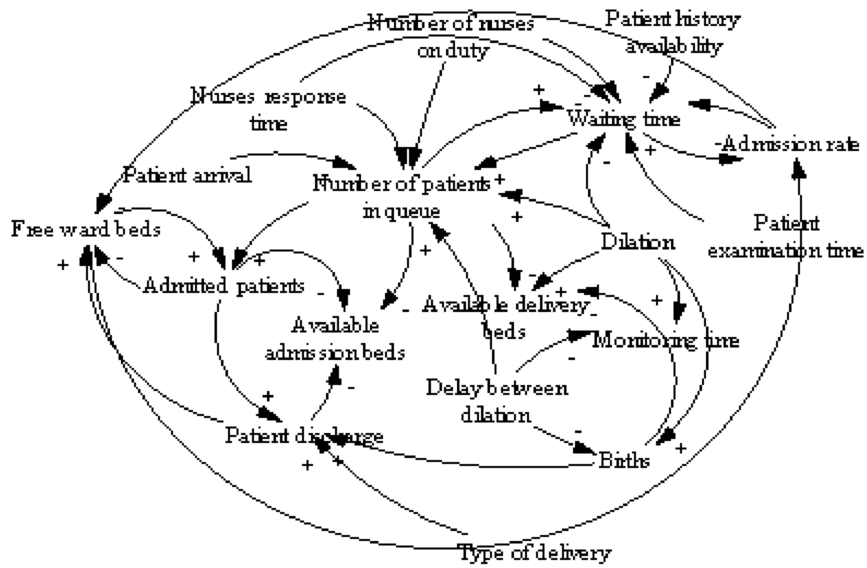


Figure 7. Causal Loop Diagram for Intrapartum in Ugandan Hospitals

4.2.2 Second Session

Here, participants started off by completing the CLD model presented in figure 6, this did not take a lot of time because part of the model had been developed in session one. In figure 7 we present a complete CLD. After completing the CLD, participants took a break of 10-15 minutes before embarking on what we refer to as the main problem for this study "*transforming a CLD into an ORM model*". Each participant was given a sheet of paper containing steps on how to transform a CLD into an ORM model. Participants were asked to follow the given steps, freely critic each step, highlight weaknesses, strength and make suggestions for improvement.

In step 1, besides the CLD-ORM trans-formation step, the moderator also provided participants with a guiding question "*Who are the main role player(s) in a given CLD variable*?" Here, participants were expected to identify object(s) in each CLD variable. As participants identified objects in each CLD variable, the moderator listed them down on a flip chart. For some of the CLD variables, object(s) were not easy to identify for example variable '*birth*'. In such cases a domain expert was asked to define '*facts'* pertaining to that particular CLD variable to enable participants clear the ambiguities.

To represent participant's views, we used different colors and before we proceeded to the next variable, we made sure that there was consensus on the identified object. In table 3 we present the identified object(s) for each vari-able name.

In conclusion, this phase took a lot of time because some of the identified CLD variables needed to be made more explicit which required asking the domain expert.

Having completed Step 1, participants proceed with step 2. In this step, participants were asked to classify similar

Variable Name	**Identified objects**	**Variable Name**	**Identified objects**
Number of nurses on duty	*Nurses*	Birth (net or number of births)	*Patient and Nurse*
Patient arrival	*Patient*	Patient discharge	*Patient, Nurse and Bed*
Number of patients in queue	*Patients*	Monitoring time	*Patient, Nurse and time*
Available admission beds	*Admission bed*	Dilation	*Patient*
Available delivery beds	*Delivery bed*	Free ward beds	*Ward bed*
Waiting time	*Patient, Time and Nurse*	Admission rate	*Patient, Nurse and Time*
Nurse response time	*Nurse and time*		
Patient examination time	*Patient, Time and Nurse*	Admitted Patients	*Patient and admission bed*
Patient History available	*Patient and Patient history*	Delay between dilation	*Patient*

Table 3. Objects Identified from CLD Variable Names

objects. By classifying these objects, they were able to come up with the following object types; *Nurse, patient, time, Antenatal card, Dilation stage, delivery bed, ward bed and patient history, Admission bed etc.* This step was neither time consuming nor hard therefore participants suggested that this step be merged with step 1 because they are much alike. On completing step 2 participants proceeded with step 3. Here, participants started by identifying roles played by objects in the same object type then identified roles played in other object types. As a result, fact types between or among object types were identified. There were some arguments within this step that required consensus building, therefore this step took more time than the previous steps. Secondly, participants kept referring to the CLD in figure 7 as more roles were identified which made us realize that the transformation process is an iterative process. In table 4 we present some of the variable names, derived ORM model and their verbalizations.

transformation steps, they easily identify issues in the already developed CLD hence, better and improved CLD variable names were attained.

4.2.3 Third session

This session started off with a recap of what transpired in the previous session. Thereafter discussions on whether to merge the model first or add the constraints started. Majority of the participants agreed to merge the model before adding constraints to avoid repetition of step 4. This indicated that steps 4 and 5 should be reversed or performed in parallel to avoid repetition of tasks.

Please note that adjustments have been made to this model. Since we used a flip chart, the model was drawn on different pages. In order for us to present the final model, we had to join different pages which made presentation of the model difficult. We therefore reconstructed the model

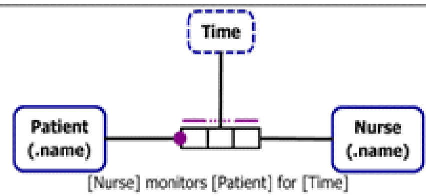
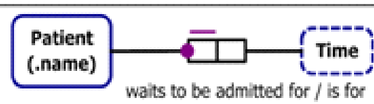
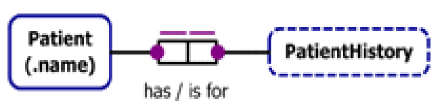
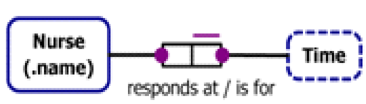
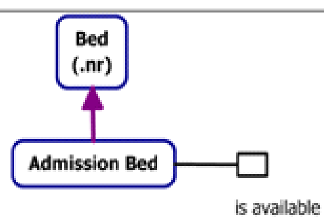
Variable name	ORM model	Verbalization
Monitoring time New variable name *['Patient monitoring time']*		For each Patient, some Nurse monitors that Patient for some Time. For each Nurse and Patient, that Nurse monitors that Patient for at most one Time.
Waiting time *New variable name* *['Patient admission waiting time']*		Each Patient waits to be admitted for exactly one Time. It is possible that the same Time is for more than one Patient.
Patient history available	a) 	Each Patient has exactly one PatientHistory. Each PatientHistory is for exactly one
	b) 	PatientHistory is available.
Nurse response time		Each Nurse responds at some Time. Each Time is for exactly one Nurse. It is possible that the same Nurse responds at more than one Time.
Available admission beds		Each Admission Bed is an instance of Bed. Admission Bed is available.

Table 4. Some of the Results of Step 3 Discussions

In conclusion, having participants engage in the CLD devel-opment process was of great advantage because participants had a better understanding of the variables which made the transformation process very interactive. Secondly, as participants proceeded with CLD-ORM

presented in figure 8 using Microsoft Visual Studio. Secondly, we adjusted some constraints to give more meaning to the model verbalizations. After transforming this model onto Microsoft Visual Studio, we emailed it to all participants for their final comments. These comments

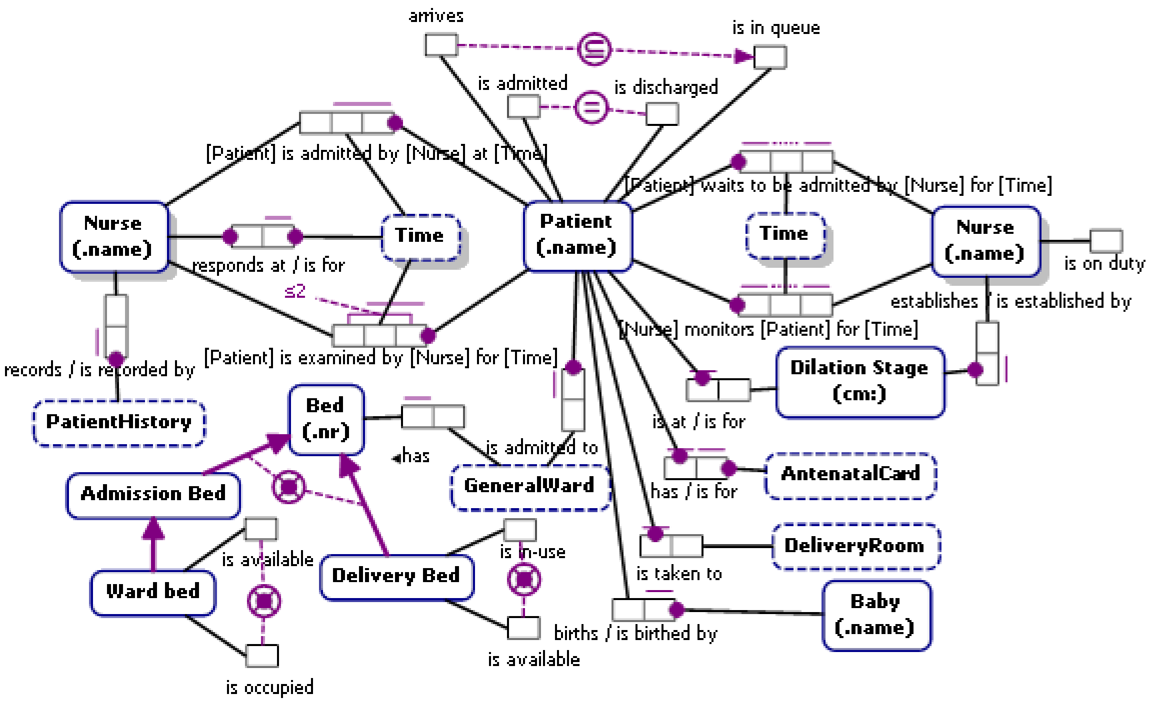


Figure 8. Derived ORM model

A	B	C
If some Patient arrives **then that** Patient is in queue. Patient has AntenatalCard. **Each** Patient has **exactly one** AntenatalCard. **Each** AntenatalCard is for **exactly one** Patient. Nurse records PatientHistory. **Each** PatientHistory is recorded by **exactly one** Nurse. **It is possible that the same** Nurse records **more than one** PatientHistory. Patient is examined by Nurse for Time. **Each** Patient is examined by **some** Nurse for **some** Time. **For each** Time and Patient, **that** Patient is examined by **at most one** Nurse for **that** Time. **Each** Nurse, Patient **combination in the population of "Patient is examined by** Nurse **for** Time" **occurs there at most** 2 **times**. Nurse responds at Time. **Each** Nurse responds at **some** Time. **Each** Time is for **exactly one** Nurse. **It is possible that the same** Nurse responds at **more than one** Time. Dilation Stage is established by Nurse. **Each** Dilation Stage is established by **exactly one** Nurse. **It is possible that the same** Nurse establishes **more than one** Dilation Stage.	Patient is at Dilation Stage. **Each** Patient is at **exactly one** Dilation Stage. **It is possible that the same** Dilation Stage is for **more than one** Patient. Patient waits to be admitted by Nurse for Time. **Each** Patient waits to be admitted by **some** Nurse for **some** Time. **For each** Patient and Nurse, **that** Patient waits to be admitted by **that** Nurse for **at most one** Time. Patient is admitted by Nurse at Time. **For each** Time and Patient, **that** Patient is admitted by **at most one** Nurse at **that** Time. **Each** Patient is admitted by **some** Nurse at **some** Time. Patient is admitted to GeneralWard. **Each** Patient is admitted to **exactly one** GeneralWard. **It is possible that more than one** Patient is admitted to **the same** GeneralWard. GeneralWard has Bed. **For each** Bed, **at most one** GeneralWard has **that** Bed. **It is possible that the same** GeneralWard has **more than one** Bed. **For each** Bed, **exactly one of the following holds:** **some** Admission Bed is **that** Bed; **some** Delivery Bed is **that** Bed.	**Each** Admission Bed **is an instance of** Bed. **Each** Delivery Bed **is an instance of** Bed. **Each** Ward bed **is an instance of** Admission Bed. **For each** Ward bed, **exactly one of the following holds:** **that** Ward bed is free; **that** Ward bed is occupied. **For each** Delivery Bed, **exactly one of the following holds:** **that** Delivery Bed is in-use; **that** Delivery Bed is available. Nurse monitors Patient for Time. **For each** Patient, **some** Nurse monitors **that** Patient for **some** Time. **For each** Nurse and Patient, **that** Nurse monitors **that** Patient for **at most one** Time. Patient is taken to DeliveryRoom. **Each** Patient is taken to **exactly one** DeliveryRoom. **It is possible that more than one** Patient is taken to **the same** DeliveryRoom. Patient births Baby. **Each** Baby is birthed by **exactly one** Patient. **It is possible that the same** Patient births **more than one** Baby. Patient is discharged. **For each** Patient, **that** Patient is discharged **if and only if that** Patient is admitted.

Table 5. Verbalizations for Figure 8

were used to further refine the ORM model shown in figure 8. As a validation mechanism for figure 8, we present its verbalizations in the script below.

In table 5, we present figure 8 verbalizations. As we stated in section 1, ORM has a strong foundation in natural language analysis. By stating the facts for each variable in the model, were able to validate the model. Letters A, B, C are used to provide an appealing presentation for figure 8 verbalizations.

5. Focus group contributions and challenges

To obtain our first feedback on CLD to ORM transformation steps, we employed some of the Group Model Building (GMB) technique of involving clients in SD model formulation and conceptualization [28,40]. In our case however, we used focus group sessions with a small number of modelers (participants with knowledge on ORM, SD and modeling in other fields) not clients as is the case in GMB. A mixture of modeling techniques e.g. brainstorming, case study etc. were also applied during the focus group sessions [27,40]. We used the case study technique as a subset in the focus group sessions to enable us apply the CLD to ORM transformation steps [28].

5.1 New insights and refined steps

During the discussions a number of views were give some within the scope and other outside the scope, all these views were used not only to refine the CLD steps in section 3 but also to streamline the entire Grounded System Dynamics procedure. In table 6, we present the refined steps, purpose for each step, strengths and weaknesses identified during the application of the steps.

ORM is used for structural domain modeling; It models types as collections of individuals which can be individuated, This means that they are countable and have a definite identity [8] System Dynamics on the other hand is typically used to model, well, dynamics of the domain and not necessarily involving the flow of countable discrete entities. This implies that contiguous entities cannot be represented in an ORM model unless a specific technical treatment is given to it [7]. This therefore makes quantification of some CLD variables and their roles played hard to represent in an ORM model. Using the case study presented in this paper participants suggested that it would be helpful to have concrete examples of data for a given causal loop diagram because some of the CLD variables are hard to be represented in an ORM model. These variables may require a domain expert's input to make them explicit. For example variable 'birth' the ORM modeler would want to know how and what exactly to model for this variable. Should (s)he record that some baby was born at a particular time, the number of births, when it happened, the name of the baby, or which mother gave birth? Therefore, just giving variable name '*birth'* without being explicit about the data captured, makes ORM representation difficult. To successfully transform a CLD model into an ORM model, the modeler needs to have a dialogue with a domain expert to enable him/her understand what some of the terms in the CLD model mean. Looking at the given case, there would still be some aspect of the CLD model that is unclear.

All in all, the CLD to ORM transformation process is better carried out either in parallel or iteratively and with a domain expert present. The role of the domain expert is to help the modeler clarify the ambiguities found with CLD variables.

<table>
<tr><th>Steps</th><th>Description</th><th>purpose</th><th>Strength</th><th>weakness</th></tr>
<tr><td>1</td><td>Identify object(s) in each variable</td><td>stating objects pertaining to a particular CLD variable</td><td rowspan="5">- Gives a clear understanding of objects in each variable.
- Easy to complete.
- CLD variable names are made explicit.
- Easy for the modulator due to limited involvement in succinct formulation of ideas.
- Engaging for participants if they have taken part in the CLD modeling process.
- Produces clear, non-redundant and relevant contributions that lead to refinement of the CLD model.
- promotes shared understanding of the CLD and ORM constructs among participants.</td><td rowspan="5">- The steps are exhausting if the CLD model has numerous variables or complex.
- Identified object types need to be further streamlined.
- Does not differentiate value types from object types.
- Requires a domain expert to be present.
- Does not work well if participants do not understand the underlying ORM and CLD concepts.
- Participant's bias may sift out good ideas.
- Consensus building among participants is exhausting for the moderator and takes time.
- Hard for participants with limited ORM or SD modeling experience.
- Semantically, SD concepts were hard to fit into ORM concepts</td></tr>
<tr><td>2</td><td>Collect and group (classify) similar objects</td><td>Specifying object types in a CLD</td></tr>
<tr><td>3</td><td>Identify and connect roles played by objects in a particular object type
a. To other object types
b. To objects in the same object type</td><td>specifying roles played by objects in each object type</td></tr>
<tr><td>4</td><td>Merge the models</td><td>To view the model as a whole.</td></tr>
<tr><td>5</td><td>Add all constraints and mandatory roles</td><td>To define model boundaries</td></tr>
</table>

Table 6. Refined ORM to CLD Steps, strengths and weaknesses

5.2 Challenges Faced during the Validation Experiment

We admit that conducting focus sessions gave us immense benefits but there were numerous challenges too. One of the participants concern was that the two methods (SD and ORM) have different underpinned meaning (concepts) which makes mapping more complex. For example, in this experiment, the effects of both CLD polarities (positive and negative) and loops (rein-forcing and balancing) cannot be represented in ORM. This therefore means that constructs for both methods cannot be exhaustively captured.

Considering participant's views during the focus sessions, some participants were either in agreement or disagreement. This made analysis of tape recorded data difficult because the researcher could not easily tell whether participants' agreement(s) resulted from coercion or proper explanations from the moderator. Secondly, getting all participants fit the experiment sessions into their schedules was not an easy task. Therefore we feel that better results would have been reached if a one to one discussion was conducted. Furthermore, the loca-tion (Makerere School of Computing and Information Technology) where these focus sessions were conducted also left a lot to be desired because of the constant power cuts, strikes etc. This led to postponement and rescheduling of sessions hence delays in the overall experiment.

Finally, design research strives to recruit participants that are familiar and are potential users of the study results [35]. Using participants from different backgrounds (by different we mean they had knowledge of either both or one method(s) under study but specialize in one), we noted that some of the participants were rigid especially those using one method. They seemed not to want any changes although they admitted to the weakness in the methods and saw the importance of the study and the arguments between participants during the sessions consumed a lot of time leading to postponement of some activities for the next session.

6. Conclusions and Further Research

In our overall study we use the design science methodology [13]. This methodology provides a means to refine and improve artifacts (models, methods, constructs and instantiations) that solve a particular problem through evaluation [4]. Evaluation is a key phase in design science research project. In this paper therefore we have used focus groups as an evaluation technique to enable us refine and obtain feedback on CLD to ORM transformation. Applying the CLD to ORM transformation steps in a focus group setting indicated that one form of valida-tion experiment is not enough to have conclusive results. As stated in [17,24] focus groups are found to be inadequate as a stand-alone method but are a self-contained means of collecting data and a supplement to other research methods. We therefore propose to further carry out more experiments with SD and ORM experts. These experiments are to be done in form of walkthroughs.

References

[1] Binder, T., Belyazid, S., Haraldsson, H., Svensson, M. (2004). Developing system dynamics models from causal loop diagrams. 22nd International Conference of the System Dynamics Society.

[2] Bird, L., Goodchild, A., Halpin T. A. (2000). Object Role Modeling and XML-Schema. *In:* Alberto H. F. Laender, Stephen W. L., Veda C., Storey, editors, Conceptual Modeling ER 2000, 1920 p. 661-705. Springer LNCS, Berlin Heidelberg, Berlin, Heidelberg.

[3] Bloor, M. (2001). Focus groups in social research. Introducing qualitative methods. SAGE Publications.

[4] Bukvova, H. (2009). Research as a Process: A Comparison between Different Research Approaches. Sprouts Working papers on Information Systems, 9 (29).

[5] Eisenhardt, K. M. (1989). Building Theories from Case Study Research. The Academy of Management Review, 14 (4) 532-550.

[6] Evans, K. (2005). Requirements engineering with ORM. *In:* R. Meersman, Z. Tari, and P. Herrero, editors, On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops, volume 3762 of LNCS, p. 646-655. Springer Berlin / Heidelberg.

[7] Guizzardi, G. (2010). On the Representation of Quantities and their Parts in Conceptual Modeling. *Frontiers in Artificial Intelligence,* 209, 103-116.

[8] Guizzardi, G., Wagner, G., Guarino, N., van Sinderen, M. (2004). An Ontologically Well Founded Profile for UML Conceptual Models. *In:* Advanced Information Systems Engineering, 16th International Conference, CAiSE, 3084:112-126. Springer-Verlag LNCS, Berlin.

[9] Halpin, T. A. (1996). Business rules and Object Role Modeling, *Database Programming & Design*, p. 1-6.

[10] Halpin, T. A. (1998) 'Object-Role Modeling (ORM/NIAM)' Architectures of Information Systems. Springer Berlin Heidelberg.

[11] Halpin, T. A., Morgan, T. (2008). Information Modeling and Relational Databases. Morgan Kaufmann Publishers, 2nd edition.

[12] Halpin, T. A., Wagner, G. (2003). Modeling Reactive Behavior in ORM. In Conceptual Modeling, *In:* Proc. of the 22nd ER2003 Conference, 2813, p. 567-569. Springer LNCS, Chicago.

[13] Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19 (2) 87-92.

[14] Hevner, A. R., March, S. T., Park, J., Ram, S. (2004). Design Science in Information Systems Research, *MIS Quarterly*, 28 (1) 75-105.

[15] Jarrar, M. (2007). Towards automated reasoning on ORM schemes - mapping ORM into the dlridf description logic. *In:* Proceedings of the 26th international conference on Conceptual modeling, p. 181-197. Springer-Verlag, Berlin, Heidelberg.

[16] Jarrar, M., Demey, J., Meersman, R. (2003). On Using Conceptual Data Modeling for Ontology Engineering. *Journal on Data Semantics*, (1) 185-207.

[17] Kidd, P. S., Parshall, M. B. (2000). Getting the focus and the group: enhancing analytical rigor in focus group research. *Qualitative Health Research,* 10 (3) 293-308.

[18] Krueger, R. A., Casey, M. A. (2009). Focus groups a practical guide for applied research. SAGE Publications, 4th edition.

[19] Lane, D. C. (2000). Diagramming Conventions in System Dynamics. *The Journal of the Operational Research Society,* 51(2) 241-245.

[20] Lane, D. C. (2008). The Emergence and use of diagramming in system dynamics: a critical account. *System Research and Behavioral Science,* 43, 1135-1150.

[21] Morecroft, J. D. W. (1982). A Critical Review of Diagramming Tools for Conceptualizing Feedback System Models. *Dynamica,* 8 (1) 20-29.

[22] Papageorgiou, G. N., Hadjis, A. (2008). New Planning Methodologies in Strategic Management; An Inter-Paradigm System Dynamics Approach. *In:* B. C. Dangerfield, editor, Proceedings of the 26th International Conference of the System Dynamics Society. System Dynamics Society, Athens, Greece.

[23] Pfahl, D., Lebsanft, K. (1999). Integration of system dynamics modeling with descriptive process modeling and goal oriented measurement. *Journal of Systems and software*, 41, 135-150.

[24] Reed, J., Payton, V. R. (1997). Focus groups: issues of analysis and interpretation. *Journal of Advanced Nursing*, 26(4) 765-771.

[25] Richardson, G. P. (1996). Problems for the future of system dynamics. *System Dynamics Review*, 12, 141-157.

[26] Richardson, G. P., Pugh, L. A. (1981). Introduction to System Dynamics Modeling with DYNAMO. MIT Cambridge, MA, USA., edition.

[27] Richardson, G. P., Andersen, D.F. (1995). Teamwork in group model building. *System Dynamics Review*, 11(2) 113-137.

[28] Rouwette, E. A. J. A., Vennix, J. A. M., Mullekom, T. (2002). Group model building effectiveness: a review of assessment studies. *System Dynamics Review*, 18(1) 5-45.

[29] Schaffernicht, M. (2010). Causal loop diagrams between structure and behavior: A critical analysis of the relationship between polarity, behavior and events. *Systems Research and Behavioral Science,* 27(6) 653-666.

[30] Spector, J. M., Christensen, D. L., Sioutine, A. V., McCormack, D. (2001). Models and simulations for learning in complex domains: Using causal loop diagrams for assessment and evaluation. Computers in Human Behavior, 17, 517-545.

[31] Spyns, P. (2005). Adapting the object role modelling method for ontology modelling. *In:* M. Hacid, N. Murray, Z. Ras, and S. Tsumoto, editors, Foundations of Intelligent Systems, 3488, p. 89-120. Springer Berlin / Heidelberg.

[32] Spyns, P. (2005). Object role modelling for ontology engineering in the dogma framework. *In:* Meersman, R., Tari, Z., Herrero, P. editors, On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops, 3762, p. 710-719. Springer Berlin / Heidelberg.

[33] Sterman, J. D. (2000). Business Dynamics- Systems Thinking and Modeling for a Complex World. McGraw Hill Higher Education, edition.

[34] Stewart, D. W., Shamdasani, N. P., Rook, D. (2007). Focus Groups: Theory and Practice. SAGE Publications, Newbury Park, CA, 2nd edition.

[35] Tremblay, M. C., Hevner, A. R., Berndt, D. J. (2010). The Use of Focus Groups in Design Science Research, 22, p. 121-143. Springer US.

[36] Tulinayo, F. P., Groessler, A., Hoppenbrouwers, S., van Bommel, P., Proper, H. A. (2009). Complementing System Dynamics with Object-Role Modeling. *In:* Ford, A., Ford, D. Anderson, E. editors, Proc. of the 27th Int. Conf. of the System Dynamics Society, Albuquerque, New Mexico, USA.

[37] Tulinayo, F. P., Hoppenbrouwers, S., Proper, H. A. (2008). Integrating System Dynamics with Object-Role Modeling, *In:* Stirna, J., Persson, A. editors, The Practice of Enterprise Modeling, 15, 77-85. Springer Berlin Heidelberg, Stockholm, Sweden.

[38] Tulinayo, F. P., Hoppenbrouwers, S., van Bommel, P., Proper, H.A., (2009). Proper, Integrating System Dynamics with Object-Role Modeling and Petri Nets. *In:* Mendling, J., Rinderlema, S., Esswein, W. editors, Enterprise Modeling and information systems Architectures, p. 41-54. IFIP GI-Edition, Ulm, Germany.

[39] Tulinayo, F. P., van Bommel, P., Proper, H. A., (2011). Grounded System Dynamics: A procedure for Underpinning System Dynamics with a Domain Modeling Method. *In*: Johannesson, P., Krogstie, J., Opdah, A.L. editors, The Practice of Enterprise Modeling, 92, 112–125. Springer, Oslo, Norway.

[40] Vennix, J. A. M. (1996). Group model building: Facilitating team learning using system dynamics. Chichester: Wiley, edition.

[41] Vennix, J. A. M., Andersen D. F, Richardon G. P, Rohrbaugh, J. (1992). Model building for group decision support: Issues and alternatives in knowledge elicitation. *In:* Morecoft J. D. W., Sterman J. D. editors, Modeling for Learning Organizations. OR, p. 29-49. Productivity press: Portland,

[42] Yin, R. K. (2009). Case Study Research: Design and Methods. Sage Publications, Inc., 4th edition.