

# Operationalizing Dialogue Games for Collaborative Modeling

Stijn Hoppenbrouwers<sup>1,2</sup>, Rob Thijssen<sup>1</sup> and Jan Vogels<sup>1</sup>

<sup>1</sup>Radboud University Nijmegen, the Netherlands

<sup>2</sup>HAN University of Applied Sciences, Arnhem, the Netherlands

[stijn.hoppenbrouwers@han.nl](mailto:stijn.hoppenbrouwers@han.nl),

[jan.h.vogels@gmail.com](mailto:jan.h.vogels@gmail.com),

[thijssen\\_rob@hotmail.com](mailto:thijssen_rob@hotmail.com)

**Abstract.** In our ongoing attempt to create and support workable “rules of play” for effective, goal-driven conversations in the context of collaborative modeling, we have now reached a stage in which a first operational setup for such ‘dialogue games’ has been developed and tested. In this paper we present the basic, generic structure of a game-like procedure description, based on a prototype game for conceptual (information) modeling. The procedure has been tested and refined in a modest design cycle and has been demonstrated to work realistically. Its generic structure is meant to be applicable for different types of conceptual modeling, e.g. process modeling, goal modeling, task modeling, and so on (requiring specific rules and configurations for the different types). However, in this paper we merely aim to present our basic approach. An illustrative example is provided.

## Dialogue Games

Dialogue Games are an innovative and, admittedly, experimental interaction form meant to support, structure and guide conversations for modeling. This functionality is desirable in context of helping people (in particular, novice analysts and modelers) to efficiently and effectively proceed in a goal-driven, focused collaborative effort to create a rational conceptualization or model (Hoppenbrouwers & Rouwette, 2012). Building on previous work, we present the outline of a workable set-up for dialogue games that can be used as a basis for a

set of such games, to be used in education and possibly also in practice in fields like requirements modeling, business modeling, process modeling, etc.

Dialogue games originated as a theoretical concept from argumentation theory going back to Wittgenstein's "language games". More recently they have been operationalized in the form of structured chats (Ravenscroft & McAlister, 2006) using the simple mechanism of *openers*: offering a small set of preset text fragments from which chat participants can choose to start a chat entry. For example, basic openers could be: "I have a question: ...", or "That is a bad idea because ...".

These openers guide participants in structuring the chat, and also make this structure visible. They create clear, natural structure in the log resulting from the chat. (Hoppenbrouwers & Rouwette, 2012) first employed this idea in context of collaborative modeling. The underlying concept here is that every collaborative modeling effort essentially is a conversation producing that model (Ssebuggwawo, Hoppenbrouwers, & Proper, 2009). As the conversation progresses, the model advances, every step being traceable in the log. The chat includes not only propositions for additions or changes to the developing model, but also the extensive conversation *about* the propositions: questions, answers, argumentation, negotiation.

(Hoppenbrouwers & Rouwette, 2012) showed that the opener mechanism can be extended to include semantic and syntactic structuring besides discourse structuring, by means of providing not only openers but also 'fill-in' patterns for answers. For example, as an answer to the (predefined) question "Please give the actor role responsible for executing this activity", the predefined answer structures offered in that context could be:

- "The actor role responsible for activity [ACT] is [AR]"
- "Sorry, I cannot answer that question"

By making the structured chat opener mechanisms context sensitive, it is possible to purposefully shift conceptual and conversational focus, dynamically entering consecutive Focused Conceptualization modes or 'FoCons' (Hoppenbrouwers & Wilmont, 2010). FoCons represent brief spans in a conversation in which a clear and limited focus is maintained, for example "list all roles for process P and their related activities". Which focus to choose at what point in the conversation is typically decided by a facilitator. We offer a workable, game-like structure ("rules of play") providing guidance and support in such conversations-for-modelling, to both the facilitator (primarily asking questions) and one or more participants (mostly doing the answering). The spirit of this setup is not unlike that of a semi-structured interview, but with more explicit, model-oriented goals and structure. By also providing some generic conversational openers throughout the game, it is possible to blend highly focused conversation with more generic questioning and answering. Also, it is possible to combine the use of dialogue games with existing diagramming tools

(Hoppenbrouwers & Rouwette, 2012), for example SeeMe (Herrmann, 2009). Finally, the setup is meant to be combined with practices and principles from group facilitation and collaboration engineering (Hoppenbrouwers & van Stokkum, 2012).

Below we will briefly present the main aspects of a prototype implementation of our dialogue game concept. The game outline presented is an integration of two related projects in which prototypes were developed cyclically (testing leading to improvements) and finally evaluated through more tests, leading to observations, remarks from test subjects, and their completing brief questionnaires addressing matters like understandability and playability of the game, perception of purpose and usefulness, and participant experience. For more on this, we refer to the original master's theses: (Vogels, 2013) and (Thijssen, 2013).

## Game Structure and Rules

The dialogue game sketched in this section aims at the creation of a specific dialogue game for the conceptualization phase of model elicitation following the FCO-IM method for conceptual information modeling (Bakema, Zwart, & van der Lek, 2002). For our current purpose this method should be seen as a case and illustration. It was chosen because its operationalization is a main interest in our research group, and because it includes a well described elicitation procedure. We focus mainly on the generic setup, which should be useful for a broad range of modeling or conceptualization games. The illustrative examples, however, still concern basic FCO-IM conceptual modeling, which is a form of fact-oriented data modeling.

The game distinguishes between two roles: the **facilitator**, typically a skilled modeler/analyst who initiates and leads the game, and one or more **participants**, typically people with good domain knowledge but not necessarily with any background in systems thinking or modeling. In our prototype, we work with only one participant. The game as presented thus supports a collaboration between facilitator and participant only. However, by adding a multi-participant layer to the game, it is quite simple to extend the dialogue to a collaboration *between multiple participants* (as well as the facilitator): they propose, discuss and improve an answer before (and after) it is entered as a statement in the main dialogue. This mechanism has been successfully explored and demonstrated in (Hoppenbrouwers & Rouwette, 2012).

The top node in the hierarchical decomposition of our game procedure is the **game** as a whole. One focused modeling session will typically be done by playing the game once. The game is divided into **phases**. A phase represents a distinct part of the modeling procedure, with a distinct conceptualization goal. For example, in our prototype we split the FCO-IM elicitation procedure into two.

The example game covers only the first phase (“conceptualization”): identifying and explicitly formulating the domain concepts and relations (words, phrases) sought for, within the conceptual frame (meta-model) of the FCO-IM method. The second phase, discarded in this paper, concerns the systematic elicitation of *constraints* on populations of the model. The FCO-IM procedure requires the gathering of example sentences (‘facts’) used in recurrent communication patterns in the domain. These examples are then rephrased at a higher level of abstraction (thus going from instance level to ‘fact type’ level). The elementary type-level phrases so created are the basis for a formal structure (model) that can also be represented diagrammatically, showing the related concepts and relations (language elements) used in the domain: its ‘information grammar’ (Bakema et al., 2002).

Covering a phase iteration from start to end is called a **round**; a round is one instance of the activities belonging to a phase. In our example, an attempt at fully charting one phrase (objects and the relation(s) between them), covering one element of the domain description (called a fact type), constitutes one round. The structure of the phase allows for some freedom of action within a round, so rounds do not have to be played out identical to one another. The end of a round is a decision whether to start a new round (because the result is still incomplete or otherwise not satisfactory) or end the phase. The rules clearly describe how to proceed at this point to force a decision about ending the round. Importantly, at any point in the game the facilitator may decide to revisit completed concepts for amendments. These *ad hoc* flow decisions can override the standard flow.

Each phase of the game is subdivided in a number of **steps**. Steps are small separate sets of related activities within a round that are more operationally described than phases and rounds. Going through one iteration of a step is called a **turn**. Multiple turns can be taken to finish a step.

A **mission list** is created at the start of the game and continuously updated throughout the game. It plays a vital part in keeping focus and helps the facilitator make decisions. The mission list consists of a top level goal (the main goal of the session) and sub-goals. The facilitator has full control over the mission list. For each concept which emerges during elicitation, a number of sub-goals are created and then pursued, systematically asking questions related to the concept (based on the method’s meta-model). The current conceptualization goal is highlighted and finished goals are crossed off. The mission list (or a summary thereof) can also be viewed by the participant. Below we show an example of the mission list in mid game. Strikethrough items are goals accomplished; the highlighted concept is the one focused on in the current round. The various sub-goals are achieved through the various FoCons in the steps of the round. Space prevents us from explaining the method specific modeling terms used in the example; for more on this, see (Bakema et al., 2002).

- Create FCO-IM model of “student registration” domain
  - ~~Concept “Course” [ + ]~~
  - Concept “Student” [ - ]
    - ~~Generate 4 examples of “Student”~~
    - ~~Generate elementary fact~~
    - Elicit identifier
    - Generate LTL-FTE for repository
    - Generate OTL-FTE for repository
    - OPTIONAL: identify uniqueness constraint (UC)
    - OPTIONAL: identify totality constraint (TC)
    - Draw part of the Information Grammar Diagram
    - Validate drawn Information Grammar Diagram and repository information

The flow of the game is directed by **decisions** made by the game facilitator. Some decisions can be made by applying clear-cut, discretely decidable rules, like “if only one unfinished concept goal remains on the mission list, select this goal and move to step 2.” Sometimes, however, a more intelligent involvement of the facilitator is required. For example, it can be decided to skip a repeated and possibly annoying validation step, at some risk (for example: “is this phrase correct in the domain: <student> follows <course>”). In such cases, the rules may offer suggestions (“only skip a validation if you feel fully confident that further validation is not necessary”) but leave the decision to the facilitator.

The standard activities of which a step consists are: **FoCon**, **game procedure** and **decision**. FoCon dialogue rules to a certain degree regulate and support the actual facilitator-participant interaction within the game: the focused question-and-answer part. We divided the prototype phase into three steps based on three FoCons identified in the original FCO-IM procedure: “Generate Concepts”, “Qualify Concepts”, and “Validate Information Grammar Diagram (IGD)”.

‘Game procedures’ concern administrative background work done by the facilitator, that in advanced digital implementations of the game could be partly or fully automated. For example, the updating of the mission list is such game procedure, as is the drawing of an IGD. Decisions are concern the flow of the game as explained above.

The components as discussed shape the flow of the game, which is unpredictable (within limits). The **rules** specify the details, as explained above. Flexibility is important to deal with unexpected situations and allow for ad hoc iterations. We provide defaults which can be followed by novice information analysts who are unsure how to proceed or react. With this mechanism, we avoid the bogging down of the procedure. The flow rules guide the facilitator. The only flow rule for the domain expert is: “follow the lead of the facilitator”.

**Questions and answers** are at the heart of the game. The facilitator will try to elicit the content by systematically (but not too rigidly!) asking questions to the participant. The questions structure the conversational part of the game. Each FoCon comes with a dedicated repertoire of question and answer templates. As explained in the first section, some of these are highly focused, others very generic. The facilitator may give an example answer to help the domain expert conceive and formulate a correctly formed answer (S. Hoppenbrouwers, 2012). It is not mandatory to use every question available in a turn. Some examples of predefined questions in step 2 (phase 1) of the game are the following:

- Could you give a meaningful name for this <object type>? For instance, Fido is a *Dog*; Mercedes Benz is a *Car Brand*. (Elicits a meaningful type name for an object, label or fact type)
- How are <object>s identified? For example, a 'Dutch Citizen' has a name but also a unique Citizen Service Number (Elicits an identifier for a concept)
- How do you distinguish between <object>s in your communication? (Auxiliary question for eliciting an identifier for a concept)
- Can there be two <object>s with the same <identifier>? (Validates the uniqueness of an identifier)

The question and answer mechanism can be implemented using some basic software. Main inspiration and example for such software is the Interloc system for generic dialogue games (Ravenscroft & McAlister, 2006). Central to this enhanced chat system is the use of 'openers', as explained in the first section. Additional rules are the following.

1. A participant and facilitator alike can only speak when it is their turn to speak (i.e. turn taking is enforced). Experiences with Interloc have confirmed that this is a highly desired feature in a structured chat.

2. Relevant categorized elements (words, phrases) in the answers are marked (in the prototype: by hand, by the facilitator) so they can be easily recognized and traced by both human players and the system. Such items, once identified, can be added to the mission list, planning further question asking about them. As mentioned, the items are typically related to the meta-model of the modeling language underlying the game, but may also represent intermediate concepts ('half products') in the process towards a target conceptualization. For example, an elicited instance-level object ("John Doe") will still have to be categorized and named, and will be used as input for the elicitation of an object type concept ("Student").

3. In addition to offering a dedicated repertoire of question and answer patterns for each FoCon, a context-sensitive mechanism is added that limits the availability of questions and answers for every consecutive chat entry within the FoCon (i.e. each conversational **move**). This mechanism is role differentiated, meaning that depending on the context (step, previous move) a move within the

FoCon (for example, asking one question or giving one answer) is offered only to either the facilitator or the participant. Through the mechanism, ‘question flow’ is constrained, which turned out to work nicely. However, the flow rules can also be left void, allowing an unrestricted open FoCon flow based on a set of available questions and answers without a constrained temporal ordering or role-specific availability.

4. The interface for choosing openers first offers a choice between some briefly described question or answer options. Only after a choice is made, a matching *text template* is offered which can be altered at will by the player. This includes the tagging of specific items that make them machine recognizable. For example, at a particular point in step 2, one of the options for a facilitator move is “Ask for distinctive object identifier”. Choosing this option provides the template (already listed above) “How do you distinguish between <object>s in your communication?”. The facilitator can then fill in the <object> slot, effectively entering the move “How do you distinguish between distinguish Students in your communication?” in the FoCon dialogue. Next, one of the moves offered to the Participant is “Give the identifier asked for”. If chosen, the template plus example offered are “<object>s are distinguished by .... (For example: Cars are distinguished by Licence Numbers).

In an advanced version, it should be possible to make some parts of the template freely adjustable, and some not adjustable. This can be augmented by partially automated tagging.

## Conclusion and future directions

The synthesis of two prototypes, also extending previous and less guided experiments (Hoppenbrouwers & Rouwette, 2012), have served well in creating operational and clearly structured setups for advanced, specialized collaborative dialogue games. The next step will be to create a robust, user friendly and generically usable digital environment combining the best features of the various prototypes. As in any software development project, there will no doubt be new challenges and insights as we continue our effort to provide user friendly support for guided/structured conversations in collaborative modeling. In addition, we aim to include links with visualizations and verbalizations mirroring the concepts put forward in the conversation, thus completing the connection with more traditional, diagram-based forms of modeling. Finally, we continue our effort to include ‘groupware’ and group facilitation features in our dialogue games.

## References

- Bakema, G., Zwart, J. P., & van der Lek, H. (2002). *Fully Communication Oriented Information Modeling (FCO-IM)*. Arnhem: FCO-IM Consultancy.
- Herrmann, T. (2009). Systems Design with the Socio-Technical Walkthrough. In B. Whitworth & A. d. Moor (Eds.), *Handbook of research on Socio-Technical Design and Social Networking Systems* (pp. 336-351). Hershey: Idea Group Publishing.
- Hoppenbrouwers, S. (2012). *Asking Questions about Asking Questions in Collaborative Enterprise Modelling*. Paper presented at the The Practice of Enterprise Modeling, 5th IFIP WG 8.1 Working Conference PoEM 2012.
- Hoppenbrouwers, S., & Rouwette, E. (2012). A Dialogue Game for Analysing Group Model Building: Framing Collaborative Modelling and its Facilitation. *International Journal of Organisational Design and Engineering, special issue on Collaborative Modeling*, 2(1), 19-40.
- Hoppenbrouwers, S., & Wilmont, I. (2010). Focused Conceptualisation: Framing Questioning and Answering in Model-Oriented Dialogue Games. In P. Bommel, S. Hoppenbrouwers, S. Overbeek, E. Proper & J. Barjis (Eds.), *The Practice of Enterprise Modeling* (pp. 190-204). Berlin, Heidelberg: Springer.
- Hoppenbrouwers, S. J. B. A., & van Stokkum, W. (2012). From Dialogue Games to m--ThinkLets: Overview and Synthesis of a Collaborative Modeling Approach. *International Journal of E-Collaboration (IJeC), special issue on Collaborative Usage and Development of Models and other Visualizations*.
- Ravenscroft, A., & McAlister, S. (2006). Designing interaction as a dialogue game: Linking social and conceptual dimensions of the learning process. In C. Juwah (Ed.), *Interactions in Online Education: implications for theory and practice* (pp. 73-90). New York: Routledge.
- Ssebuggwawo, D., Hoppenbrouwers, S., & Proper, E. (2009). Interactions, Goals and Rules in a Collaborative Modelling Session. In A. Persson & J. Stirna (Eds.), *2nd IFIP WG8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2009)*. Berlin, Heidelberg: Springer.
- Thijssen, R. (2013). *An argument based approach for easier process modeling using dialog games*. Radboud University Nijmegen, Nijmegen.
- Vogels, J. (2013). *A serious gaming approach to content elicitation for FCO-IM*. Radboud University Nijmegen, Nijmegen.