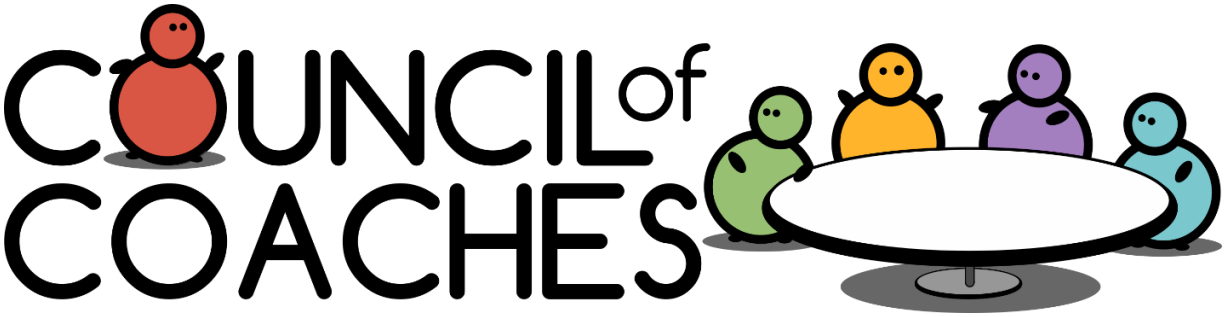

D5.3: Methods and tools for moving from dialogue game to updating individual and shared knowledge bases

Dissemination level: Public

Document type: Report

Version: 1.0.0

Date: February 27, 2019

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement #769553. This result only reflects the author's view and the EU is not responsible for any use that may be made of the information it contains.

Document Details

Project Number	769553
Project title	Council of Coaches
Title of deliverable	D5.3: Methods and tools for moving from dialogue game to updating individual and shared knowledge bases
Due date of deliverable	February 27, 2019
Work package	WP5
Author(s)	Mark Snaith (UDun), Alison Pease (UDun), Tessa Beinema (RRD), Harm op den Akker (RRD)
Reviewer(s)	Randy Klaassen (CMC)
Approved by	Coordinator
Dissemination level	Public
Document type	Report
Total number of pages	18

Partners

- University of Twente – Centre for Monitoring and Coaching (CMC)
- Roessingh Research and Development (RRD)
- Danish Board of Technology Foundation (DBT)
- Sorbonne University (SU)
- University of Dundee (UDun)
- Universitat Politècnica de València, Grupa SABIEN (UPV)
- Innovation Sprint (iSPRINT)

Abstract

This deliverable contains Dialogue Game Description Language (DGDL) descriptions that implement the theoretical dialogue game specifications provided in Deliverable 5.2. These descriptions, when executed using the Dialogue Game Execution Platform (DGEP) will provide formal dialogue actions that will be translated into communicative acts for the coaches.

Table of Contents

Symbols, abbreviations and acronyms

AIF	Argument Interchange Format
AIF+	Argument Interchange Format (+) (updated version of AIF)
ASPIC	Argumentation Service Platform with Integrated Components
ASPIC+	Argumentation Service Platform with Integrated Components (updated version of ASPIC)
CMC	Centre for Monitoring and Coaching
COUCH	Council of Coaches
D	Deliverable
DoA	Document of Action
DGDL	Dialogue Game Description Language
DGEP	Dialogue Game Execution Platform
DBT	Danish Board of Technology Foundation
EC	European Commission
ISPRINT	Innovation Sprint
M	Month
MS	Milestone
p	proposition
RRD	Roessingh Research and Development
SU	Sorbonne University
UDun	University of Dundee
UPV	Universitat Politècnica de València
UT	University of Twente
WP	Work Package

Glossary

We provide below a short glossary of terms used throughout this deliverable, and their attributed meaning as used in this document.

Proposition – abbreviated to p – a span of text with a truth value that is used as a component in an argument, usually either supporting or attacking another proposition.

Conclusion – a proposition that forms the main point of the argument that is being constructed.

Premise – a proposition which (together with other premises) is used to justify a conclusion

Graph – a structure consisting of a set of nodes connected by vertices (edges).

Directed graph – a graph in which the direction of edges is specified, e.g. A->B means there is an edge from A to B but not from B to A.

Argument Interchange Format - abbreviated to AIF; a graph-based representation format for the analysis and storage of monological arguments.

Node – a discrete component of an AIF graph.

1 Introduction

In Deliverable 5.2, we provided formal specifications for dialogue games for health coaching.

This deliverable takes the formal specifications from D5.2 and implements them in the Dialogue Game Description Language (DGDL). These implementations provide the mechanisms required for:

1. The coaches and user to engage in a structured dialogue, by providing communicative acts for use in WP6 (see D6.3 for details on the early work towards this);
2. Updates to the shared and individual knowledge bases based on these dialogues.

2 Objectives

The objective of this deliverable is to provide methods and tools that will allow the formal dialogue game specifications from D5.2 (in the DoA it specifies D5.1, but this is incorrect) to be used to conduct a structured coaching dialogue with a user, and consequently update the individual and shared knowledge bases. This consists of an initial implementation of the dialogue game in Dialogue Game Description Language (DGDL).

3 Background

3.1 Dialogue Game Description Language (DGDL)

The Dialogue Game Description Language (DGDL) is a language equipped with everything one might expect to need for the rapid development of a new dialogue system for a new domain or application (Wells & Reed, 2012; Lawrence, Snaith, Konat, Budzynska, & Reed, 2017).

The DGDL is a domain-specific language for capturing the properties, rules and moves of a dialogue game. Game specifications written in DGDL consist of three main parts: composition, rules, and interactions.

3.2 Dialogue Game Execution Platform (DGEP)

Having specified a game in the DGDL, this specification can then be processed using the Dialogue Game Execution Platform (DGEP). DGEP interprets and transforms a DGDL specification into a dialogue framework that enforces the rules in that specification.

DGEP provides a suite of tools for testing dialogue game specifications. These tools were used in the development of the game presented in Section 5.

3.3 Argument Interchange Format (AIF)

A by-product of executing dialogue games using DGEP is an argumentative structure, expressed in AIF+, which is a reflection of the outcome of the dialogue. This structure will feed into the shared knowledge base (WP3) and provide a means of reasoning about the outcome of the dialogue; for instance, determining an agreed diet plan, or new exercise goal. These structures can also be used to influence future dialogues, by allowing the coaches to reflect on the effectiveness of the strategy they adopted.

The Argument Interchange Format (AIF) (Chesñevar, et al., 2006) was specified with the aim of developing a means of expressing argument that would provide a flexible – yet semantically rich – way of representing argumentation structures. The AIF was put together to try to harmonise the strong formal tradition initiated to a large degree by (Dung, 1995), the natural language research described at Computational Models of Natural Argument (CMNA) workshops since 2001[1], and the multi-agent argumentation work that has emerged from the philosophy of (Walton & Krabbe, 1995), amongst others (see, e.g. (Parsons & Jennings, 1996; McBurney & Parsons, 2002)). As originally specified, the AIF accounted for only monological argument; it has, however, been subsequently extended by (Reed, Wells, Devereux, & Rowe, 2008) into AIF+ to support dialogical argument. This extension allows for the individual speakers in an argumentative dialogue to be identified and associated with their utterances in that dialogue.

Where we subsequently refer to further extensions to the core AIF, these apply equally to AIF+.

3.4 Abstract and structured argumentation

The work of (Dung, 1995) on abstract argumentation laid the groundwork for many theories in computational argumentation. Dung abstracted the notion of argument into two simple concepts: *arguments* and a notion of *attack* between them, with no consideration for the internal structure of the arguments. The resultant *argumentation frameworks*, expressed as directed graphs, can then be evaluated under a number of different semantics, based on the attacks; the two most common semantics, as defined by Dung, are:

[1] http://www.cmna.info

- **Grounded**, the most sceptical semantics in which an argument is *acceptable* if anything that attacks it is unacceptable. This yields a single set.
- **Preferred**, a credulous semantics in which includes all maximal subsets of acceptable arguments. This can yield multiple sets, with each set being internally consistent with respect to the attack relations

The ASPIC+ framework (Prakken, 2010) built upon the original ASPIC framework (Caminada & Amgoud, 2007) and creates a connection between models of structured argument and Dung-style argumentation frameworks. This allows real, natural arguments to be expressed in an abstract form and subsequently evaluated for acceptability. The work of (Bex, Prakken, Modgil, & Reed, 2013) developed a connection between AIF(+)[2] and the ASPIC+ framework, allowing arguments expressed in AIF to be evaluated for acceptability using Dung's theories.

[2] AIF+ is a superset of AIF and as such theories that apply to AIF equally apply to AIF+.

4 From game design to implementation

4.1 Implementation and testing

DGDL is a bespoke language that remains in the early stages of development. As such, tools for implementing dialogue games are limited; the current process is to implement small parts of a description using a simple text editor, then verify this implementation using the testing tools provided as part of DGEP.

The game specification provided in Section 5 implements all of the structural, turn-taking, termination and outcome rules in the "Patient Interview" game specified in deliverable D5.2. Commitment rules are not implemented in this specification because commitments in this game can be handled externally by the knowledge bases. Only where the presence or otherwise of a commitment causes a change to the structural or turn-taking rules is it necessary to incorporate commitments explicitly in the game specification.

4.2 Pre-interview and post-interview sessions

In deliverable D5.2 we showed formal dialogue games for pre-interview sessions and post-interview sessions, in addition to scenarios with the user. In the first of these, healthcare practitioners are invited to participate in a patient interview, and briefly discuss the patient and what they expect to cover during the interview. In the post-interview session, the practitioners discuss how it went, which coaching strategies were effective and what next steps for them might be. While we continue to see these pre and post-sessions as important parts of a coaching interaction, we have made the decision not to implement the formal dialogue games for these from deliverable D5.2. We outline our reasons below.

Firstly, it is not clear that we need to follow the agent metaphor for the pre and post-sessions, since the user is not expected to see these parts. Current models of the overall architecture of the COUCH system contain a shared knowledge base rather than individual, therefore all of the tasks done in the pre and post-sessions can be performed without a dialogue. Even so, it may still be beneficial to show a dialogue, for two reasons: (a) as a user-friendly way of showing developers what is going on under the hood, and (b) as an attractive way of offering the user transparency regarding decisions behind their sessions. In the first case, since one of our goals is to develop an open agent platform in which third party developers can add their own coaches, this may well be useful. In the second case, giving the user the option of seeing the reasoning behind their interactive session may enhance their sense of ownership and understanding of the system. For instance; Why were certain coaches present/absent? Why did they talk about the negative consequences of non-action? How will they handle the session next week? We plan to design transparency into the system from the ground up, and to allow the user to see this sort of thinking in the form of a dialogue would be a valuable way of doing so. Of course, in the case of either developer or user seeing the dialogues, these could be scripted rather than needing formal dialogue and dialogue specifications. Alternatively, since we will already have the dialogue framework in place, it may be sensible to employ it for this purpose as well.

Since it is not the priority, we plan to work on the interactive dialogues for the user sessions first. If we can, we will then also implement the pre and post sessions as a dialogue. In this case we will need to conduct a separate study to determine what information is used and shared by the virtual agents and how this can usefully be presented as a dialogue to the user. The current dialogue games from deliverable D5.2 were based on very simplified interactions in the pre and post phases. Therefore, either way, it does not make sense to implement these. We will continue to discuss the issue.

5 Dialogue Game Description Language specification

We provide here the current DGDL specification that will underpin coaching sessions in the Council of Coaches. The game itself is highly expressive, with several different move types available at each point of the dialogue. Choosing which move(s) to make at which time will be determined by the coaching strategies defined during WP3. A cut-down version of the game is currently used in the second technical prototype. This cut-down version was designed to test the integration between DGEP and other components in the COUCH system; this integration is described in Deliverable 6.3.

5.1 DGDL specification

Below is the DGDL specification for the "Patient Interview" dialogue game, which will underpin coaching sessions. In Section 5.2 we provide natural language descriptions of the key components of the game specification. Additionally, a handbook describing the construction of DGDL specifications is available online[3].

```
System{PatientInterview{
    turns{magnitude:single, ordering:liberal}
    roles{Coach,Patient,Person,speaker,listener}
    players{min:2, max:undefined}

    player{id:Coach, roles{Coach,Person}}
    player{id:Patient, roles{Patient,Person}}

    backtrack{on}

    /* -- RULES -- */
    rule{id:StartingRule, scope:initial,
        {
            assign(Coach, speaker)
            & move(add, next, PureQuestion, $Patient, {p}, Coach)
        }
    }

    /* -- INTERACTIONS -- */
    interaction{PureQuestion, $Person, {p}, PureQuestioning, {p}, "$p?",
        {
            move(add, next, Assert, $Person, {p}, Target)
            & move(add, next, Assert, $Person, {q}, Target)
        }
    }

    interaction{Assert, $Person, {p}, Asserting, {p}, "$p",
```

[3] https://drive.google.com/open?id=1N76nzRSlAkZYzp-zS3cZDQZiGZ0vG3gZ This will always link to the latest PDF version of the handbook.

```
28          {
29                  if{ inrole(Target, Coach) } then
30                  {
31                       /* If the Target is a coach then last person to speak
32                                was the patient... */
33                       move(add, next, Assert, {q}, Person)
34                       & move(add, next, ReportSpeech, $Person, {q}, Person)
35                       & move(add, next, AssertiveQuestion, $Person, {q}, Person)
36                       & move(add, next, RhetoricalQuestion, $Person, {q}, Person)
37                       & move(add, next, PureQuestion, $Person, {p}, Person)
38                       & move(add, next, PureChallenge, $Person, {p}, Person)
39                       & move(add, next, Agree, $Person, {p}, Person)
40                       & move(add, next, Disagree, $Person, {p}, Coach)
41                  } else
42                  {
43                       /* ...and vice versa */
44                       move(add, next, Assert, $Person, {q}, Person)
45                       & move(add, next, ReportSpeech, $Person, {q}, Person)
46                       & move(add, next, AssertiveQuestion, $Person, {q}, Person)
47                       & move(add, next, RhetoricalQuestion, $Person, {q}, Person)
48                       & move(add, next, PureQuestion, $Person, {p}, Person)
49                       & move(add, next, PureChallenge, $Person, {p}, Person)
50                       & move(add, next, Agree, $Person, {p}, Person)
51                       & move(add, next, Disagree, $Person, {p}, Patient)
52                  }
53
54          }
55      }
56
57      interaction{ReportSpeech, $Person, {p}, Reporting, {p}, "Report $p",
58          {
59              if{ inrole(Target, Coach) } then
60              {
61                       move(add, next, Assert, $Person, {q}, Person)
62                       & move(add, next, ReportSpeech, $Person, {q}, Person)
63                       & move(add, next, AssertiveQuestion, $Person, {q}, Person)
64                       & move(add, next, RhetoricalQuestion, $Person, {q}, Person)
65                       & move(add, next, PureQuestion, $Person, {p}, Person)
66                       & move(add, next, PureChallenge, $Person, {p}, Person)
67                       & move(add, next, Agree, $Person, {p}, Person)
68                       & move(add, next, Disagree, $Person, {p}, Coach)
69              } else
70              {
71                       move(add, next, Assert, $Person, {q}, Person)
72                       & move(add, next, ReportSpeech, $Person, {q}, Person)
73                       & move(add, next, AssertiveQuestion, $Person, {q}, Person)
```

```
74                      & move(add, next, RhetoricalQuestion, $Person, {q}, Person)
75                      & move(add, next, PureQuestion, $Person, {p}, Person)
76                      & move(add, next, PureChallenge, $Person, {p}, Person)
77                      & move(add, next, Agree, $Person, {p}, Person)
78                      & move(add, next, Disagree, $Person, {p}, Patient)
79              }
80          }
81      }
82
83      interaction{AssertiveQuestion, $Person, {p}, AssertiveQuestioning, {p}, "$p?",
84          {
85              if{ inrole(Target, Coach) } then
86              {
87                  move(add, next, Agree, $Person, {p}, Target)
88                  & move(add, next, Disagree, $Person, {p}, Target,
89                          event(last, AssertiveQuestion, Patient))
90                  & move(add, next, Rephrase, {q, p}, Target)
91              } else
92              {
93                  move(add, next, Agree, $Person, {p}, Target)
94                  & move(add, next, Disagree, $Person, {p}, Target)
95              }
96          }
97      }
98
99      interaction{RhetoricalQuestion, $Person, {p}, RhetoricalQuestioning, {p}, "$p?",
100         {
101             move(add, next, Assert, $Person, {q}, Person)
102             & move(add,next, PureQuestion, $Person, {q}, Person)
103             & move(add, next, AssertiveQuestion, $Person, {p}, Person)
104             & move(add, next, AssertiveQuestion, $Person, {q}, Person)
105             & move(add, next, PureChallenge, $Person, {p}, Person)
106             & move(add, next, Rephrase, $Person, {q, p}, Person)
107         }
108     }
109
110     interaction{PureChallenge, $Coach, {p}, PureChallenging, {p}, "Why $p?",
111         {
112             move(add, next, Assert, $Person, {q}, Target)
113             & move(add, next, Rephrase, $Person, {q, p}, Target)
114         }
115     }
116
117     interaction{Agree, $Person, {p}, Agreeing, {p}, "I agree with $p",
118         {
119             move(add, next, Assert, $Person, {q}, Person)
```

```
120             & move(add, next, ReportSpeech, $Person, {q}, Person)
121             & move(add, next, AssertiveQuestion, $Person, {q}, Person)
122             & move(add, next, PureQuestion, $Person, {q}, Person)
123             & move(add, next, Agree, $Person, {p}, Person)
124         }
125     }
126
127     interaction{Disagree, $Person, {p}, Disagreeing, {p}, "I disagree with $p",
128         {
129             if{ event(last, Disagree, Patient) } then
130             {
131                 move(add, next, Assert, $Person, {q}, Patient)
132                 & move(add, next, ReportSpeech, $Person, {q}, Patient)
133                 & move(add, next, PureChallenge, $Person, {p}, Patient)
134                 & move(add, next, PureChallenge, $Person, {! p}, Coach)
135             } else
136             {
137                 move(add, next, PureChallenge, $Person, {! p}, Person)
138                 & move(add, next, Assert, $Person, {q}, Coach)
139                 & move(add, next, ReportSpeech, $Person, {q}, Coach)
140                 & move(add, next, AssertiveQuestion, $Coach, {! p}, Coach)
141                 & move(add, next, PureQuestion, $Person, {q}, Coach)
142                 & move(add, next, Agree, $Person, {! p}, Coach)
143             }
144         }
145     }
146 }}
```

5.2 Description of implementation

5.2.1 Composition

Lines 1-9 make up the composition of the game specification, defining various parameters under which the game will operate.

Turns defines whether or not a single participant can speak more than once when it is their move in the dialogue ("magnitude"), and whether or not there is a strict ordering between speakers ("ordering"). For this game, we allow a single turn per move, but the ordering is liberal – that is, there is no defined order on who can speak.

Roles defines the types of participant in this dialogue. The roles of "speaker" and "listener" are special roles that are used to control when a participant in permitted to speak.

Players defines the minimum and maximum number of players (participants) in the dialogue. And "undefined" maximum allows for a theoretically unlimited number of participants.

Backtrack is a Boolean "on" or "off" that, if on, allows participants to return to a previous point in the dialogue; that is, moves do not necessarily have to be sequential.

5.2.2 Rules

Line 12 defines the initial rule for dialogues that use this specification. The initial rule is executed when the dialogue commences and sets the initial conditions from which the dialogue will proceed. In this dialogue game, the initial rule assigns any participant in the role of "Coach" to be a current speaker (line 14), then makes available to those coaches the move of Assert, targeted at the patient (line 15).

5.2.3 Interactions

Line 27 onwards defines the interactions, which form the communicative acts within the dialogue. Since all interactions follow a similar pattern, we will describe here only the first ("Assert").

The signature of the interaction (line 27) defines:

1. its name ("Assert");
2. (optionally) a participant role at which this interaction can be targeted ("Person", which is anyone in the dialogue)
3. (optionally) the illocutionary force associated with this move ("Asserting")
4. a variable that will represent the content of the interaction, passed into the dialogue by the participant ("p")
5. a free-text string that can, if required, be used in a user interface ("$p" – which will be the content of the variable "p")

Line 28 onwards defines the effects of communicating this interaction, which are conditional (line 29) on the specific role occupied by the participant to whom this interaction is targeted (in point 2 above). In both cases of the conditional, the only effects are to add next available moves to certain participants. Line 33, for instance, adds the move "Assert" as a next available move to any "Person" in the dialogue. The set "{q}" represents new content that is passed to the move. Line 40, on the other hand, adds the move "Disagree" only for participants in the role of Coach. This has a counterpart in line 51, where the "Disagree" move is added for participants in the role of "Patient". This is the reason for the conditional in line 29: the formal game specification requires that if a coach was the last person to speak, only the patient can disagree, and vice versa.

5.3 Knowledge base updates

Encoded within the DGDL specification are illocutionary actions that, when uttered by a user or a coach, cause an argumentative structure, expressed in AIF+, to be built. This argumentative structure is then used to bring about updates to the shared and individual knowledge bases[4], possibly following argumentative evaluation (see subsection 5.4 below).

For example, the *Assert* interaction (Line 27) has an associated illocutionary action of *Asserting*. The content of this action will be whatever propositional content is assigned to the variable *p* – that is, what the patient actually says. This content will then be added to shared and individual knowledge bases, and subsequently used both later in this dialogue, and in future dialogues.

5.4 Argumentative evaluation of dialogues

Either during or after a dialogue, the argumentative structure that is built can be evaluated to determine either the outcome (after a dialogue) or current status of a particular proposition (during); for instance, a proposed course of action put forward by a coach might not currently be accepted by the user because they have put forward a counter-proposal, or expressed some reservation.

[4] The choice of which knowledge base to update will lie at the application level, external to the dialogue and argumentation framework.

To carry out this evaluation, the AIF+ structure generated by the dialogue is first translated into ASPIC+ (Prakken, 2010) using established theories and tools (Snaith & Reed, 2012; Bex, Prakken, Modgil, & Reed, 2013). The resultant translation is the evaluated using different argument semantics (see subsection 3.4) to determine the currently acceptable arguments (statements).

6 Summary and Conclusions

In this deliverable we have provided a DGDL implementation of the "Patient Interview" dialogue game from deliverable D5.2. This implementation, when executed, provides communicative acts to the user interface (WP6; D6.3) and generates structures in Argument Interchange Format (AIF) that can be used to update the individual and shared knowledge bases. Argumentative evaluation of these AIF structures can also subsequently be used to influence future dialogues between the coaches and the user.

7 Bibliography

Bex, F., Prakken, H., Modgil, S., & Reed, C. (2013). On Logical Reifications of the Argument Interchange Format. *Journal of Logic and Computation, 23*(5), 951-989.

Caminada, M., & Amgoud, L. (2007). On the evaluation of argumentation formalisms. *Artificial Intelligence, 171*(5), 286-310.

Chesñevar, C., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., . . . Willmott, S. (2006). Towards and Argument Interchange Format. *Knowledge Engineering Review, 21*(4), 293-316.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence, 77*(2), 321-357.

Lawrence, J., Snaith, M., Konat, B., Budzynska, K., & Reed, C. (2017). Debating Technology for Dialogical Argument: Sensemaking, Engagement and Analytics. *ACM Transactions on Internet Technology, 17*(3), 24:1-24:23.

McBurney, P., & Parsons, S. (2002). Dialogue Games in Multi-Agent Systems. *Informal Logic, 22*(3), 257-274.

Parsons, S., & Jennings, N. (1996). Negotiation through Argumentation - a preliminary report. *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS 1996)* (pp. 267-274). Kyoto, Japan: AAAI Press.

Prakken, H. (2010). An abstract framework for argumentation with structured arguments. *Argument and Computation, 1*(2), 93-124.

Reed, C., Wells, S., Devereux, J., & Rowe, G. (2008). AIF+: Dialogue in the Argument Interchange Format. *Proceedings of the Second International Conference on Computational Models of Argument (COMMA 2018)* (pp. 311-323). Toulouse, France: IOS Press.

Snaith, M., & Reed, C. (2012). TOAST: online ASPIC+ implementation. *Proceedings of the 4th International Conference on Computational Models of Argument (COMMA 2012)* (pp. 509-510). Vienna, Austria: IOS Press.

Tang, Y., Norman, T. J., & Parsons, S. (2009). A model for integrating dialogue and the execution of join plans. *Proceedings of the Eigth International Conference of Autonomous Agents and Multi-Agent Systems* (pp. 883-890). Budapest, Hungary: International Foundation for Autonomous Agents and Multi-Agent Systems.

Walton, D. N., & Krabbe, E. C. (1995). *Commitment in Dialogue.* New York: SUNY Press.

Wells, S., & Reed, C. (2012). A Domain Specific Language for Describing Diverse Systems of Dialogue. *Journal of Applied Logic, 10*(4), 309-329.