



HAL
open science

Dialogue management in task-oriented dialogue systems

Philippe Blache

► **To cite this version:**

Philippe Blache. Dialogue management in task-oriented dialogue systems. 1st ACM SIGCHI International Workshop on Investigating Social Interactions with Artificial Agents (ISIAA'17), Nov 2017, Glasgow, United Kingdom. 10.1145/3139491.3139507. hal-01708376

HAL Id: hal-01708376

<https://hal.science/hal-01708376>

Submitted on 13 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dialogue Management in Task-Oriented Dialogue Systems

Philippe Blache

Aix Marseille Univ, CNRS, LPL, Aix-en-Provence, France
blache@lpl-aix.fr

ABSTRACT

This paper presents a new framework for implementing a dialogue manager, making it possible to infer new information in the course of the interaction as well as generating responses from the virtual agent. The approach relies on a specific organization of knowledge bases, including the creation of a *common ground* and a belief base. Moreover, the same type of rules implement both inference and control of the dialogue. This approach is implemented within a dialogue system for training doctors to break bad news (ACORFORMed).

CCS CONCEPTS

• Human-centered computing → Natural language interfaces;

KEYWORDS

Task-oriented dialogue, Common Ground, Inference, Dialogue management

ACM Reference Format:

Philippe Blache. 2017. Dialogue Management in Task-Oriented Dialogue Systems. In *Proceedings of 1st ACM SIGCHI International Workshop on Investigating Social Interactions with Artificial Agents (ISIAA'17)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3139491.3139507>

1 INTRODUCTION

We investigate in this paper a particular case of *task-oriented* dialogue system in which the user has to deliver information to an artificial agent. Our system aims at training professional to face-to-face interaction, in which doctors have to break bad news. In such situation, the doctor interacts with an artificial agent playing the role of a patient.

The trainee receives first a scenario describing the damage (e.g. intestinal perforation during a *coelioscopy*) and the proposed solution (e.g. surgery). The task consists in explaining the patient the problem in an understandable manner for a non-specialist, using an adequate attitude (excuses, empathy) and adapting the message in function of patient's emotions.

This work is part of the ACORFORMed¹ project (*Virtual Reality for Training Doctors to Break Bad News*) [10] aiming at developing a conversational avatar playing the role of a patient in a virtual reality environment. This type of system is particular in terms of dialogue management: the virtual agent has access beforehand to



Figure 1: The ACORFORMed environment

the entire knowledge, including the information to be delivered as well as the way the doctor is supposed to deliver it. Moreover, at the difference with an actual conversation, the agent only reacts occasionally with specific types of productions: backchannels, emotions, clarification questions and answers to doctor's questions. The role of the dialogue manager is then to identify what type of production to trigger as well as generating its contents.

2 KNOWLEDGE ORGANIZATION

The management of task-oriented dialogue systems relies on different sources of knowledge: the description of the context, the evolution of the information instantiated during the interaction as well as the respective beliefs of the stakeholders of the dialogue. We present in this section the characteristics of these knowledge bases.

2.1 The Universe of Interpretation

From the perspective of the doctor, the information has to be delivered according to certain requirements and recommendations [12], and is organized around three different phases: *greetings*, *damage description*, *remediation*. The doctor has a complete knowledge of the context, the degree of severity, the risk, etc. The agent also has access to the same knowledge. Figure 2 presents a subpart of the knowledge base describing an intestinal perforation that occurred during a surgery for polyp resection. All different events are represented as an attribute-value pair.

In the case of a human-human conversation, information updating consists in building a shared knowledge, called "*common ground*" [1, 3, 13], made of what is supposed to be known by all stakeholders. The task consists in adding new information (the *increment*) in relation to an item of the common ground (the *anchor*). In this common knowledge base, many information is also presupposed or can be inferred incrementally. What is specific to the common ground is that all participants suppose the others also have access to the same knowledge.

¹ACORFORMED project ANR-14-CE24-0034-01; <http://www.lpl-aix.fr/acorformed/>

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

ISIAA'17, November 13, 2017, Glasgow, UK

© 2017 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.

ACM ISBN 978-1-4503-5558-2/17/11...\$15.00

<https://doi.org/10.1145/3139491.3139507>

Initial pathology	Remediation
Diagnosis: <i>polyp</i>	Type: <i>surgery</i>
Symptom: <i>abdominal pain</i>	Urgency: <i>high</i>
Action: <i>surgery</i>	Operator: <i>surgeon</i>
Expected_benefit: <i>cure</i>	Complexity: <i>low</i>
	Expected_benefit: <i>cure</i>
Damage	Personal data
Type: <i>perforation</i>	Name: <i>Camille</i>
Cause: <i>instrument</i>	Gender: <i>female</i>
Operator: <i>surgeon</i>	Age: <i>35</i>
Severity: <i>high</i>	Physical condition: <i>good</i>
Treatment: <i>obligatory</i>	

Figure 2: Example of Universe of Discourse

In the case of a training dialogue environment, the context, the scenario and the recommendations are already known by the system (i.e. the virtual agent) before the interaction (this fact is hidden to the trainee). The evolution of information transfer from the doctor to the patient, instead of an updating mechanism, simply comes to specify in the knowledge base what has been transferred. As a consequence, the system knows at any time what has been updated and what remains to be instantiated. We distinguish the three types of information that constitute the knowledge base:

- The **universe of discourse**: what is actually known by the system and the trainee.
- The **increment**: what is instantiated by the trainee during the session.
- The **beliefs**: what the trainee supposes the virtual agent knows.

We propose in the following to precise our knowledge organisation taking into account these specificities.

2.2 Different Point of Views, Different Bases

The particular situation of a training environment necessitates to distinguish and precise different types of information concerning knowledge and beliefs from the doctor and the patient.

First, at the difference with natural interactions, the discourse universe is a *closed word* in the logical sense: it is fully specified and all information that does not belong to the universe is considered to be false. This renders possible not only to reduce ambiguity and uncertainty, but also to control drastically the possible inferences. Concretely, a complete knowledge base is created, containing the general medical knowledge, the scenario of the specific damage to be announced and the way the damage has to be announced (the different phases and their contents). This constitutes the complete universe of interpretation and control of the dialogue. As mentioned above, the particularity here is that the trainee and the virtual agent have both access to what we call here the *knowledge base*.

Besides this global knowledge, it is necessary to create and update a base of what the trained doctor thinks the patient knows or believes. This type of knowledge is overt for both stakeholders, human and virtual and corresponds to what is usually considered as the *common ground*. In this application, the common ground is a subpart of the knowledge base, representing in fact what the doctor believes the patient knows.

Finally, from the patient (i.e. the virtual agent) point of view, it is also necessary to create and update, besides the common ground, a specific *belief base* consisting in inferences from the information that is delivered. This base makes it possible to verify soundness and coherence of doctor's discourse.

To sum up, three different bases are maintained in our training dialogue environment:

- **Knowledge base**: complete knowledge of the dialogue context and goals, accessible by both the trainee and the virtual agent.
- **Common ground**: the initial overt knowledge (that the patient is supposed to have) plus the information *instantiated* during the dialogue.
- **Belief base**: contains the common ground plus the knowledge that can be *inferred* by the patient from doctor's discourse.

We present in the following how to take advantage of such organization for the supervision of the dialogue system.

3 KNOWLEDGE REPRESENTATION

The different bases contain facts and beliefs that constitute events, in the sense of [7]. We propose in this section to detail the organization of such events first by proposing a global representation in terms of attribute/value structures, and then by specifying some particular features.

3.1 Feature Structure Representation

Events are typed feature structures, their values can be atomic or complex and correspond to different types: strings, boolean, composites, etc. Events are then associated to definition domains, that can be lists, rendering possible for a feature to be associated to an exhaustive list of values. Among them, it is often the case that one can be chosen as a *default* value.

Moreover, we have seen the importance of tracking knowledge base updating: knowing whether the value of an event has been *inferred* or *instantiated* during the discourse is necessary. Finally, another important information is whether the feature belongs or not to the common ground or, in other words, whether the trainee doctor thinks the agent knows the event (i.e. has the event instantiated in the base).

In our system, all three bases of knowledge are gathered into a single one, which corresponds to the global knowledge base in which the status of each information (whether it also belongs to the common ground and/or the belief base) is specified. These different information are represented explicitly as shown in figure 3.a. Moreover, we integrate to this representation two features **DEPENDENT-VALUES** and **REQUIREMENT** that will be described in the following sections.

The example given in 3.b illustrates a diagnosis event for intestinal surgery: the possible values are listed, among which the default is *polyp*². During its discourse, the doctor has instantiated the value *colorectal-polyp* which is more specific than the possible values (but not incompatible as we will explain later). Finally, this

²The two last features **DEPENDENT-VALUES** and **REQUIREMENT** will be illustrated later.

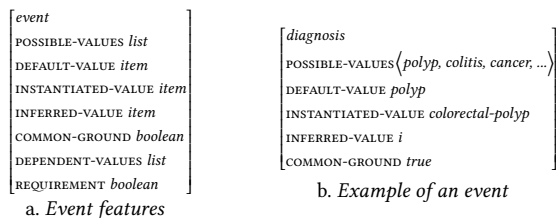


Figure 3: Feature structure organisation

event is supposed by the doctor to be known by the patient, and then belongs to the *common ground*.

3.2 Dependent Values

Information is not only structured, it is also organized temporally: the occurrence of an event is often conditioned by the realization of another one. For example, the prescription of a surgery can only be done after establishing the diagnosis. In the same way, the diagnosis of a disease is usually done after the apparition of some symptoms.

In our knowledge organization, this constraint can be encoded directly within the representation of the event thanks to a specific feature. The figure 4 presents the example of the *surgery* event, that can be described on the basis of a diagnosis, relying on the analysis of symptoms.

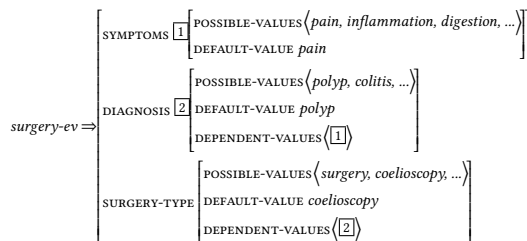


Figure 4: Dependency between events within a structure

As it is usually the case with feature structures, elements can be indexed and referred to within a same structure. This mechanism is the basis of structure sharing for example in HPSG. It stipulates the fact that the coindexed feature structures refer to the same element. In this example, the *diagnosis* and *surgery-type* events have a dependent value (respectively *symptoms* and *diagnosis* events).

3.3 Weighting Events

In the description of an event, some features can be more important than others. It is possible to encode such information with continuous values, representing the weights and making it possible to rank the events thanks to their importance. This information can be of some interest when evaluating the coherence or the completeness of a structure. In our approach, we reduce the possible weights to two values, representing the fact that some events has to be instantiated obligatorily, whereas some others can be left free or underspecified. This constraint is encoded by the *requirement* feature that can take two values: *hard* and *soft*.

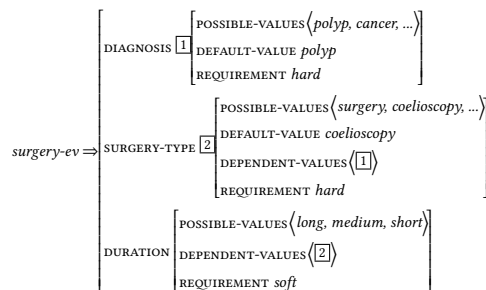


Figure 5: Relative weights

The example in figure 5 represents the fact that the instantiation of *diagnosis* and *surgery-type* sub-events is mandatory for the event *diagnosis* to be complete. Their *REQUIREMENT* value is then set to *hard*, whereas the *DURATION* of the surgery can be left underspecified.

4 INFERENCE AND RULES

Knowledge management requires an inference relation making it possible to generate new knowledge starting from the one which is instantiated during the discourse. We present in this section how rules can implement this relation. A first section proposes a brief sketch of the inference rule language we use. The two last sections describes how rules can implement both feature inference and dialogue actions: inference rules constitute a single framework for knowledge management and dialogue supervision.

4.1 A Language for Representing Inference Rules

Many different representations of inference rules exist from artificial intelligence [4, 5], knowledge management [6, 9], etc. We present here an adapted version of the agent programming language *AgentSpeak/Jason* [2, 8, 11]. In our representation, rules distinguish three components:

- The specification of the context, events, features that are involved in the inference relation.
- The conditions that trigger the inference.
- The consequent of the inference.

These three parts are represented in rules as follows³:

context : conditions -> consequent

The definition of the context and the access to feature values relies on the identification of the elements (events, features) to be involved in the rule and unification. The first operation consists in labelling an object with a type, following the syntax of the *casting* operation. For example, the operation (feature)f1 consists in creating an object of type *feature* with the label f1. Unification is represented by the symbol “=” and makes it possible to specifies values in the identification of the triggering objects. For example, the context (event.surgery)s1 \wedge [s1.duration = d1] indicates an event s1 of type *surgery* which has a feature *duration* of label d1.

³At the difference with Jason, the left-hand side of the rule, made of the context and the conditions, represents the antecedent, the right-hand side the consequent.

4.2 Rules for Inferring New Knowledge

The instantiation of new values during the discourse opens the possibility to infer new knowledge. For example, the doctor may announce an urgency for the remediation surgery, which means that the problem is serious. This commonsense knowledge can be instantiated in the base as *inferred-value* (simplified to *infer*) of the corresponding feature thanks to the following rule:

```
R1: (event.surgery)s1 ∧ [s1.urgency = u1] : u1 == high ->
    s1.severity.infer = high;
```

Note that this new knowledge is *inferred* which means, according to our definitions, that the patient *believes* now this information. In some cases, the inferred knowledge may not be necessarily true, but only believed by the patient. For example, if the doctor announces the necessity of a long operation, the patient may think that the problem is serious, the risk high and the complexity of the surgery important (which may or may not be true).

This commonsense knowledge can be instantiated in the base the *inferred-value* (simplified to *infer*) of the corresponding features describing the surgery:

```
R2: (event.surgery)s1 ∧ [s1.duration = d1] : d1 == long ->
    s1.severity.infer = high;
    s1.risk.infer = high;
    s1.complexity.infer = high;
```

4.3 From Knowledge Inference to Action Inference

Besides instantiating inferred values, the same type of rules can be used directly to manage dialogue processing. In most of the cases, the virtual patient speaks in reaction to the doctor's discourse by answering a question, expressing emotion or asking for precisions. The two last situations can be controlled by inference rules.

For example, if the doctor delivers an unexpected or incoherent information, this can be identified by the fact that the corresponding instantiated value is different than the default value of the same feature. This is implemented thanks to the following rule:

```
R3: (feature)f1 ∧ [f1.instantiated = i1] ∧ [f1.default = d1]
    : i1 != d1 ->
    ask-question(default-mismatch, i1, d1);
```

We do not describe here the module of language generation in the dialogue system. Simply note that the *ask-question* action passes as parameters the type of the question (here *default-mismatch*) and the values concerned by the question. The generating module will elaborate then the corresponding message to be produced by the virtual agent.

This mismatch situation can be refined to the cases where the term used by the doctor is similar but not exactly the same as that expected. This is not a mismatch in the sense of the previous rule, but rather a *similarity gap* between the terms. Such discrepancy can be taken into account by introducing distributional information in the feature values: instead of being terms such as *polyp*, *cancer*, *surgeon*, etc. feature values can be represented by the corresponding distributional vectors \vec{polyp} , \vec{cancer} , $\vec{surgeon}$, etc. (extracted from existing corpora of transcribed training sessions). In this case, the mismatch problem comes from a similarity value (calculated with

classical measures derived from cosine, for example) that is lower than a certain threshold, as implemented by the following rule:

```
R4: (feature)f1 ∧ [f1.instantiated = i1] ∧ [f1.default = d1]
    : sim(i1, d1) < threshold ->
    ask-question(default-similarity, i1, d1);
```

Note that the *ask-question* action takes the adequate type value as for the generation of the answer.

The last situation we illustrate as for the use of rules is dialogue management. We have introduced in the knowledge representation the notion of *dependent values*, implementing the synchronization of feature instantiation: some features have to be instantiated before others. This type of mismatch is identified by the fact that the feature has a dependent value which value is not already instantiated or inferred.

```
R5: (feature)f1 ∧ [f1.dependent = d1] ∧
    [d1.instantiated = null] ∧ [d1.infer = null] ->
    ask-question(default-synchro, i1, i2);
```

This rule can be applied for controlling the synchronization of the different phases of a dialogue (greetings, damage description, etc.): each phase corresponds to an event that may have as dependent value another one. Typically, the event *damage description* has the vent *greetings* as dependent value.

5 CONCLUSION

We have presented in this paper an approach implementing a dialogue supervision system relying on the management of a knowledge base, a common ground and a belief base. Inference rules exploit this information in order to generate new knowledge as well as controlling the coherence and the progress of the dialogue. This approach has been developed in the framework of a specific human-machine dialogue task: training the human agent to deliver information in a specific context.

ACKNOWLEDGMENTS

This work, carried out within the Labex BLRI (ANR-11-LABX-0036) and the Institut Convergence ILCB (ANR-16-CONV-0002), has benefited from support from the French government, managed by the French National Agency for Research (ANR) and the Excellence Initiative of Aix-Marseille University (A*MIDEX).

REFERENCES

- [1] ALLAN, K. What is common ground? In *Perspectives on linguistic pragmatics* (2013), Springer, pp. 285–310.
- [2] BORDINI, R. H., HÜBNER, J. F., AND WOOLDRIDGE, M. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley and Sons, 2007.
- [3] CLARK, H. *Using language*. Cambridge University Press, 1996.
- [4] COLMEAUER, A. Theoretical model of prolog ii. In *Logic Programming and its Applications*, M. van Caneghen and D. Wane, Eds. Ablex Series in Artificial Intelligence, 1986.
- [5] FRÜHWIRTH, T. *Constraint Handling Rules*. Cambridge University Press, 2009.
- [6] HORROCKS, I., PATEL-SCHNEIDER, P., BOLEY, H., TABET, S., GROSF, B., AND DEAN, M. Swrl: A semantic web rule language combining owl and ruleml. Tech. rep., W3C Member Submission, 2004.
- [7] McRAE, K., AND MATSUKI, K. People use their knowledge of common events to understand language, and do so as quickly as possible. *Language and Linguistics Compass* 3, 6 (2009), 1417–1429.
- [8] MILLER, T., PFAU, J., SONENBERG, L., AND KASHIMA, Y. Logics of common ground. *Journal of Artificial Intelligence Research* 58 (2017), 859–904.
- [9] MOTIK, B., PATEL-SCHNEIDER, P. F., AND GRAU, B. C. Web ontology language: Direct semantics. Tech. rep., W3C Recommendation, <http://www.w3.org/TR/owl2-direct-semantics/>, 2012.

- [10] OCHS, M., DONVAL, B., AND BLACHE, P. Virtual patient for training doctors to break bad news. In *proceedings of WACAI-2016* (2016).
- [11] PFAU, J., MILLER, T., AND SONENBERG, L. Modelling and using common ground in human-agent collaboration during spacecraft operations. In *proceedings of SpaceOps Conferences* (2014).
- [12] SCHNEBELEN, C., POTHIER, F., AND FURNEY, M. Annonce d'un dommage associé aux soins. Tech. rep., Haute Autorité de Santé, 2011.
- [13] STALNAKER, R. Common ground. *Linguistics and Philosophy* 25, 5 (2002), 701–721.