Towards Modelling and Using Common Ground in Tutorial Dialogue

Mark Buckley and Magdalena Wolska
Department of Computational Linguistics
Saarland University
66041 Saarbrücken, Germany
{buckley|magda}@coli.uni-sb.de

Abstract

In order to avoid miscommunication participants in dialogue continuously attempt to align their mutual knowledge (the “common ground”). A setting that is perhaps most prone to misalignment is tutoring. We propose a model of common ground in tutoring dialogues which explicitly models the truth and falsity of domain level contributions and show how it can be used to detect and repair students’ false conjectures and facilitate student modelling.

1 Motivation

In order to communicate efficiently, participants in a dialogue take into account the information believed to be mutually known to them: the “common ground” (Clark and Schaefer, 1989). This concerns not only knowledge accumulated in the course of dialogue, but also common ground (context) that is presupposed prior to the interaction. In order for a piece of information to become common ground (henceforth CG), it must be explicitly or implicitly acknowledged by the interlocutor in the process called grounding (Clark and Schaefer, 1989; Traum, 1994). Lack of grounding may lead to incorrect beliefs about CG or, using Stalnaker’s term, “defective context” (Stalnaker, 2002) i.e. a situation in which discourse participants presuppose different things. This, in turn, may lead to miscommunication.

A setting that is perhaps most prone to misalignment in discourse participants’ beliefs is tutoring. Student-tutor teaching interactions are characterised by an inherent asymmetry of knowledge possessed by the tutor and the learner (Munger, 1996; Lee and Sherin, 2004). In order to effectively guide the student in a problem-solving task, the tutor must make inferences as to the student’s state of knowledge based on his/her interpretation of student’s utterances. Empirical research shows, however, that untrained tutors tend to perform specific targeted moves (e.g. use curriculum scripts, example problems, give immediate evaluative feedback) that locally address the student’s progress (or lack thereof) on the task at hand, instead of focusing mainly on coordinating the CG, i.e. establishing complete understanding of the students’ state of beliefs, and so cognitive alignment (Graesser et al., 1995). Considering this, it is difficult for tutors to identify, let alone repair, deep misconceptions that underlie students’ errors. However, when a misalignment in student’s beliefs as to CG becomes apparent based on the linguistic content of the student’s utterance, the tutor may choose to explicitly address it by challenging the student’s statement. Moreover, an explicit model of CG can feed into a module that monitors the student’s performance (McArthur et al., 1990; Merrill et al., 1995).

In this paper, we propose a preliminary model of CG in tutoring dialogues on problem solving, in the context of building a conversational intelligent tutoring system for mathematical proofs. As the dialogue progresses the CG in our model develops as a store of the truth and falsity of the contributions that the student has made, based on the evaluations that the tutor has given. In addition, discourse referents for formulas are introduced to support modeling their salience. The CG forms the context for judging whether an utterance constitutes evidence that misalignment has occurred.

We begin by discussing examples of student-tutor interactions that exemplify the use of CG in this domain and motivate the need for modelling CG in an automated system (Section 2). We present the structure of the CG maintained by the...
dialogue model, the mechanism of updating the CG, and discuss the uses of CG in the tutorial dialogue: detecting and repairing student’s false conjectures and facilitating student modelling (Section 3). We give a walk-through of our examples in Section 4 and related work is discussed in Section 5.

2 Linguistic Data

To collect data about naturalistic human-computer tutorial dialogue interactions, we conducted two experiments in a Wizard-of-Oz paradigm (Kelley, 1984). The subjects and the wizards were using natural language (German; typed on the keyboard) and mathematical symbols available on the GUI to interact with the simulated system, and they were unconstrained in their language production.

In the first experiment (Wolska et al., 2004) the subjects were tutored using one of the following strategies: minimal feedback (students were given feedback only on correctness and completeness of proposed proof-steps), didactic (when the student made no progress, the tutor disclosed the expected reasoning step), or socratic (a pre-defined hinting algorithm was used to guide the student toward the solution). In the second experiment (Benzmüller et al., 2006) the tutors did not follow any tutoring algorithm, but were given general instructions on socratic tutoring. The first experiment concerned naive set theory and the second binary relations. In both, students were given study material before the tutoring session containing the basic mathematical knowledge required to solve the exercises. Overall, the collected corpora consist of 22 and 37 dialogue logfiles for the first and second experiment respectively.

2.1 Dialogue Phenomena

The CG that student and tutor believe currently exists in the dialogue can become misaligned as the dialogue progresses, for instance due to misunderstanding, misconception, or the boundedness of attentional resources. Evidence of misalignment of CG can be observed, for example, in certain situations in which informationally redundant utterances (IRUs) (Karagjosova, 2003; Walker, 1993) are performed. An utterance is informationally redundant if the proposition it expresses is entailed, presupposed or implicated by a previous utterance in the discourse.

If an unmarked IRU is performed then the information it contains, which has already been grounded, is being repeated without the speaker indicating that this repetition is being done on purpose. This indicates to the hearer that this information was not in what the speaker believes to be the CG of the dialogue. The hearer must then conclude that the CG has become misaligned.

Sometimes it is necessary to repeat information that has already been grounded, for example to make a known fact salient again to support an argument (Walker and Rambow, 1994). Such utterances are informationally redundant. To prevent the hearer concluding from such utterances that misalignment has occurred, the speaker explicitly indicates that he is aware that he is repeating shared information (i.e. that the fact should be CG) by marking with phrases such as “of course”. In tutorial dialogue hints which remind students of previously proved facts and concepts from the tutoring domain are IRUs. Students also use IRUs to check if what they believe the CG to be is actually that which the tutor believes it to be. In the following examples from the corpus we give English translations and include the original German utterances where they are illustrative.

In (1) the domain content of utterance S10, that the assertion that the formula embedded in the utterance holds, is repeated in utterance S18.

\[
S10: \text{It holds that } (R \cup S) \circ T = \{(x, y) \exists z (z \in M \land (x, z) \in (R \cup S) \land (z, y) \in T) \\
T10: \text{That’s right!} \]
\[
\ldots \\
S18: \text{By definition it holds that } (R \cup S) \circ T = \{(x, y) \exists z (z \in M \land (x, z) \in (R \cup S) \land (z, y) \in T) \\
T18: \text{That’s right! You’ve already performed this step.} \\
\text{(German: Diesen Schritt haben Sie vorhin schon vollzogen.)}
\]

The confirmation (T10) of this fact puts the truth of the formula in S10 in CG, and therefore when utterance S18 is performed it is an IRU. Because S18 is unmarked for informational redundancy, the tutor concludes that misalignment of context has occurred, i.e. the fact concluded in S10 is no longer CG for the student. He augments his confirmation (T18) with the indication (“already”) that the step in S18 had already been performed, telling the student that misalignment occurred and realigning their CG.

In example (2) the student explicitly introduces by assumption a fact that he has already proved.
(2) S3: Let \((a, b) \in (R \circ S)^\dagger\). Then it holds that 
\((b, a) \in (R \circ S)\).
T3: That’s right.
...
S6: Let \((b, a) \in (R \circ S)\). Then...
T6: Since you already know that \((b, a) \in (R \circ S)\),
you don’t need to postulate it again.

In S3 the student has successfully proved \((b, a) \in (R \circ S)\),
and the tutor’s confirmation of this (T3) makes it CG. The student later wants to use this fact as
the antecedent of a derivation (S6), but wrongly introduces it as a new assumption. Thios shows
that the truth of this formula is no longer CG for the student, i.e. misalignment has taken
place. In order to realign the CG the tutor reminds the student that he has previously proved the for-
mula, and this utterance is marked with “already” (T6).

In example (3) an IRU performed by the tutor is
marked so that the student does not mistakenly
conclude that misalignment has taken place.

(3) S2: \(A \cap B = \emptyset\)
...
T4: Right. Now what?
...
T8: ...The justification could for instance be: Let
\(x\) be an arbitrary element of \(B\), then it can’t
be in \(A\) (since of course \(A \cap B = \emptyset\)) ...
(German: ... (da ja \(A \cap B = \emptyset \)) ...)
The student has proved a formula in S2 which was
confirmed in T4, making it CG. In T8 the tutor
recaps the solution proof. The formula \(A \cap B = \emptyset\)
is part of the proof, and is thus in the CG, so
the tutor marks the reminder of this fact with the
particle “of course”.

An example of a student’s marked IRU is shown
in utterance S4 of (4), in which the IRU is used
by the student to check suspected misalignment.
“Doch” is a modal particle which, when deac-
cented, marks old or shared information.

(4) S3: and for the powerset it holds that: \(P(C \cup (A \cap
B)) = P(C) \cup P(A \cap B)\)
T4: Do you really mean: \(P(C \cup (A \cap B)) = P(C) \cup P(A \cap B)\)?
S4: But I think: \(P(A) \cup P(B) = P(A \cup B)\)
(German: ich denke doch: \(P(A) \cup \ldots\))
T5: That’s not right! Maybe you should have an-
other look in your study material.
S5: sorry, it holds of course that: \(P(C \cup (A \cap
B)) \subseteq P(C) \cup P(A \cap B)\)
T6: Really?
S6: oh, no... the other way around
T7: That’s right at last!

In S3 the student claims a fact which is then ques-
tioned by the tutor. This causes the student to sus-
pect a misalignment, because a rule he used in
deriving the fact and which he believed to be true is
in fact false. In S4 the student then checks whether
this rule is in fact in the CG by stating it explicitly.
He considers S4 to be uninformative, and there-
fore marks it explicitly with “doch” (meaning “but
I thought...”). However S4 actually is informa-
tive, in the sense that it is not subsumed by the
set of facts in the CG when it is uttered. This leads
the tutor to conclude that misalignment has taken
place. In addition to rejecting the rule, he also di-
 rects the student to the study material. The next
student proof step (S5) is again rejected (T6). In
S6 the student gets the rule right, which is con-
firmed in T7. The student adds the corrected rule
to his CG, completing the realignment that began
in S4.

The data shows that misalignment occurs be-
tween student and tutor, and that it can be ob-
served in the case of informationally redundant
utterances. Unmarked IRUs (such as S18 in ex-
ample (1) and S6 in example (2)) are evidence
that CG has become misaligned, and should trig-
ger strategies for realignment. Conversely, when
IRUs are to be generated as part of pedagogical
strategies (T8 in example (3)), these should be
marked as such in order to avoid the student falsely
concluding that misalignment has occurred. Fi-
nally, misalignment can be evidenced by utter-
ances which are marked for informational redu-
dancy but are in fact not IRUs (S4 in example (4)).
To account for such phenomena a model of CG
is necessary that allows the detection of which ut-
terances are informationally redundant and which
are not, at the level of truth in the domain. The
CG must therefore model the utterances that were
performed and whether their content was accepted
by the tutor, and thus grounded.

3 Modelling Common Ground

Our model is developed within the wider scope of
a tutorial environment for mathematics. The stu-
dent’s task is to build a proof of a mathematical
theorem. The student does this by conducting a
dialogue with the tutor in which he/she performs
utterances which may contain proof steps. The en-
vIRONMENT includes study material for the theory
at hand. Turn-taking during tutoring sessions is
strictly controlled. Each correct proof step extends
the current partial proof. The task is completed when the student has constructed a complete proof of the theorem.

3.1 Elements of the Architecture

We now briefly describe the roles played by those system modules in the environment which are directly relevant to analysing CG.\footnote{We omit a discussion of other modules which are part of the architecture.}

**Discourse Interpreter** A discourse interpretation module analyses students’ natural language utterances.\footnote{For the purposes of this exposition we only consider assertion-type dialogue moves which contain domain contributions (here labelled with domain), that is, possibly underspecified proof steps. For example we do not treat questions or meta-level communication management etc.} The result of the analysis is the linguistic meaning of the utterance, the dialogue move that represents its function and, in case of domain contributions, a formal representation $p$ for the proof manager (realised as, for example, in (Wolska and Kruijff-Korbayová, 2004)). In particular, the linguistic meaning of modal particles such as “doch” is reflected in the representation in that a feature \texttt{MARKED} is present.

**Proof Manager** A proof manager maintains the solution the student is building and evaluates proof steps in this context (Dietrich and Buckley, 2007). It can check the correctness and relevance of proof steps by accessing a domain reasoner such as \texttt{OMEGA} (Siekmann et al., 2006) or \texttt{Scunak} (Brown, 2006).

**Tutorial Manager** A tutorial manager stores pedagogical expertise on when and how hints should be given and maintains a student model (Tsovaltzi et al., 2004). The tutorial manager computes what dialogue moves the system should perform. Two possible dialogue moves which are relevant for this model are \texttt{accept} and \texttt{reject}. It performs the content selection step for output generation, which includes deciding whether utterances which are informationally redundant should be marked as such. It also decides whether to realise moves in the declarative or interrogative mood, as in utterance T4 in example (4).

3.2 Our Model

We model CG as being similar to that of DeVault and Stone (2006). Their objective normative con-

\textit{text} is a product of the actions taken by the dialogue participants. In our case, actions in the dialogue result in the dialogue participants having beliefs about the truth (or falsity) of the propositions that are contributed by the student and evaluated by the tutor. This is combined with the knowledge in the study material that the students are given before the tutorial session. We assume that it is part of the CG at the start of the dialogue. In our model the CG contains the facts that propositions were uttered, the evaluations of those utterances by the tutor, and the facts that the student knows about the domain as a result of preparatory study.

3.2.1 Types of Entities in the Model

There are two types of entities in the model: discourse referents (for entities introduced in the discourse) and propositions.

Domain contributions contain or refer to formulas that the student uses or concludes. For each domain contribution the discourse interpreter delivers the discourse referent for the proposition that the utterance expresses. Our model includes these discourse referents in the common ground. When references are made to substructures of formulas, for instance “the left-hand side of . . . ” we add new referents as needed.\footnote{This account of discourse referents is intentionally simple — a full account would require for instance referents for actual utterances in order to resolve references like “what I said above”.}

The fact that a proposition was uttered is modeled as \texttt{uttered}(\texttt{speaker}, $p$), where \texttt{speaker} is the dialogue participant who performed the utterance. Having \texttt{uttered}(\texttt{speaker}, $p$) in the CG tells us only that the event took place, and does not tell us anything about the truth of the content $P$ of the utterance. Evaluations of the propositions that were uttered have the form either $\texttt{holds}(p)$ or $\neg\texttt{holds}(p)$, depending on whether they were accepted or rejected by the tutor. For previous knowledge that the student is assumed to have we use $\texttt{prev}(p)$.

Finally we model the utterances which are performed in the dialogue as objects $u$, and access the proposition $p$ expressed by an utterance $u$ with $p = \texttt{expresses}(u)$. In this way we can access the proof step that an utterance contained. The propositions expressed by utterances are treated as verbatim formulas.

The entities described above are represented in the dialogue model as shown in Figure 1, where
CG is the common ground and LU is the last utterance in the dialogue. CG/REFS contains the discourse referents which have been introduced so far. CG/PROPS contains the three types of propositions that we model, namely uttered, \(\neg\text{holds}\) and previous. Both of these slots have the type ordered set which provides functions for membership, push and pop. We use this representation as simple account of salience. In CG/REFS, the most salient discourse entity is the one whose referent is at the top of the stack.

The LU part of the dialogue model stores a representation of the last utterance. It includes which dialogue participant performed the utterance, the actual utterance itself, the set of dialogue moves that the utterance realised as well as the discourse referents that were addressed. The flag marked indicates whether the utterance was marked for informational redundancy.

### 3.2.2 Updating the Common Ground

Information is added to the CG as a result of utterances performed by the student and the tutor. This corresponds to implicit grounding at the understanding level (Clark, 1996).\(^4\)

We model three dialogue moves which lead to an update of the CG: domcon, accept and reject. Domain contributions claim the truth of formulas derived by proof steps, and in our model they correspond to Clark's proposal or presentation phase. In the case of a domcon we make the updates push(uttered\((s, p), CG/PROPS\)) and push\((i, CG/REFS)\), where \(p\) is the content of the proposition and \(i\) is the discourse referent introduced by \(p\).

The tutor's evaluations of domain contributions performed by the student are represented by accept and reject moves, which lead to the updates push(\(\text{holds}(p), CG/PROPS\)) and push\((\neg\text{holds}(p), CG/PROPS)\) respectively. Here we rely on the discourse interpreter being able to determine which domain contribution the tutor is responding to and, in turn, what its propositional content \(p\) was. Because they have the effect of putting things in CG, accept and reject moves correspond to Clark's acceptance phase. We make the assumption that the tutoring scenario gives the tutor the authority to ground content simply by evaluating it. In effect, the student is expected to accept what the tutor says as being true. A further update is made to the CG when a substructure of a formula is accessed. New discourse referents \(j\) for subformulas are generated on demand and added to the CG by push\((j, CG/REFS)\).

### 3.2.3 Testing and Using the Common Ground

Now that we can update the CG to reflect the current state of mutual belief, we define a set of predicates (see Table 1) that test properties of given propositions and referents. The predicate \(\text{exists}(x)\) holds when the discourse referent or proposition \(x\) has already been introduced, and \(\text{salient}(i)\) holds of the most salient discourse referent \(i\). Utterances are IRUs if they satisfy the predicate \(\text{iru}(u)\), that is, if the proposition they express is already in the CG. Since \(\text{expresses}\) treats formulas verbatim, \(\text{iru}(u)\) can only hold when the formula is a case-insensitive match of \(u\). We also define an operation \(\text{makesalient}(i)\) which promotes an existing discourse referent \(i\) to the top of \(CG/REFS\), making it the most salient referent. The \(\text{makesalient}\) operation is performed when \(\text{iru}(u)\) is detected because a formula is being mentioned for a second time and should become salient again.

The \(\text{exists}\) predicate allows us to determine whether utterances are informationally redundant.

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>exists()</td>
<td>(i \in CG/REFS)</td>
</tr>
<tr>
<td>exists(p)</td>
<td>(\forall \text{ holds}(p) \in CG/PROPS)</td>
</tr>
<tr>
<td>salient(i)</td>
<td>(i = \text{top}(CG/REFS))</td>
</tr>
<tr>
<td>iru(u)</td>
<td>(\text{exists}(	ext{expresses}(u)))</td>
</tr>
</tbody>
</table>

Table 1: Predicates on common ground.
in the context of the dialogue, and using exists we can now define a test on the dialogue model which tells us whether the last utterance is evidence that the common ground has become misaligned. Informally, we can conclude that misalignment has occurred if an IRU is not linguistically marked (−) for informational redundancy or if a non-IRU is marked (+) for informational redundancy. In terms of the predicates introduced above, we express this condition with the predicate misaligned:

misaligned iff (LU/MARKED− ∧ iru(LU/UTTERANCE)) ∨ 
LU/MARKED+ ∧ ¬iru(LU/UTTERANCE)

In determining when to mark tutor utterances as informationally redundant, the tutoring manager uses the CG as input when it is asked to generate tutorial content. This way it can check if iru(u) holds of a planned utterance u and if so add marking.

4 Examples

We now illustrate how our model accounts for the examples above. For the purpose of example (1) we let \( p_1 \) stand for the proposition embedded in utterance S10, so that \( p_1 = \text{expresses}(S10) \). The domain contribution realised in S10 triggers the updates push(uttered\((s,p_1)\),CG/PROPS) and push\((i_{p_1},CG/REFS)\). The accept performed by the tutor then triggers the update push\(\text{holds}(p_1)\),CG/PROPS), and the resulting CG is

\[
\begin{align*}
\text{CG} \quad & \begin{bmatrix}
\text{REFS} & \langle i_{p_1}, \ldots \rangle \\
\text{PROPS} & \langle \text{uttered}(s,p_1), \text{holds}(p_1), \ldots \rangle
\end{bmatrix}
\end{align*}
\]

When S18 is performed the predicate misaligned becomes true. This is because S18 is an IRU (the proposition it expresses is a match of \( p_1 \)) but is unmarked for informational redundancy. The system concludes that misalignment has taken place and the tutoring module generates a reminder that the student should already believe that \( \text{holds}(p_1) \), helping him to realign.

Example (2) shows a direct reference to a fact that the student should know. As in example (1), the contribution S3 followed by the acceptance T3 results in CG/PROPS containing \( \text{holds}(p_2) \), where \( p_2 = (b,a) \in (R \circ S) \). In S6 the student assumes this fact again and the system can determine that misaligned holds because the formula in S6 matches \( p_2 \). It performs makesalient\((i_{p_2})\), and the tutoring module reminds the student that he already knows the fact \( p_2 \) (T6).

As a result of the utterances S2 and T4 in example (3), the CG includes uttered\((s,p_3)\) and \( \text{holds}(p_3) \), where \( p_3 \) is the formula \( A \cap B = \emptyset \). When the system recapitulates the solution in T8, one of the utterances expresses a proposition which matches \( p_3 \). That means that \( \text{exists}(p_3) \) holds and that this utterance is an IRU. So that the student does not mistakenly conclude that misalignment took place, the system generates the utterances augmented with the marking “of course” to indicate informational redundancy.

We treat example (4) in more detail because it shows how misalignment can be detected and repaired. In Figure 1 we saw the state of the dialogue model after utterance S3. T4 is a reject, so the model is updated to

\[
\begin{bmatrix}
\text{CG} \quad & \begin{bmatrix}
\text{REFS} & \langle i_{p_1}, \ldots \rangle \\
\text{PROPS} & \langle \text{holds}(p_4), \text{prev}(P(A) \cup \ldots) \rangle
\end{bmatrix}
\end{bmatrix}
\]

where \( p_4 = \text{expresses}(S3) \). The student has a misconception that \( P(A) \cup P(B) = P(A \cup B) \) holds. Since this rule is not correct, there is no proposition \( \text{prev}(P(A) \cup P(B) = P(A \cup B)) \) in CG/PROPS. That means that when the utterance S4, in which the student checks whether the misconceived rule is correct or not, is performed, we have \( \neg \text{iru}(S4) \). However, the marking of S4 with the particle “doch” signals that the student assumes it to be shared knowledge. From these two facts the system detects that misalignment occurred. This type of misalignment informs the tutoring module to execute a strategy to resolve a misconception, namely, the student is referred to the study material. The resulting state is

\[
\begin{bmatrix}
\text{CG} \quad & \begin{bmatrix}
\text{REFS} & \langle i_{p_5}, i_{p_4}, \ldots \rangle \\
\text{PROPS} & \langle \neg \text{holds}(p_5), \neg \text{holds}(p_4), \ldots \rangle
\end{bmatrix}
\end{bmatrix}
\]

In S5, with \( p_6 = \text{expresses}(S5) \), the student tries to correct the proof step that was rejected in utterance S3 by using a different rule, but the rule he applies (that \( P(A) \cup P(B) \supseteq P(A \cup B) \)) is not the correction of his original misconception, and the step is rejected (T5). The update is analogous, and we now have

\[
\begin{bmatrix}
\text{CG} \quad & \begin{bmatrix}
\text{REFS} & \langle i_{p_6}, i_{p_5}, i_{p_4}, \ldots \rangle \\
\text{PROPS} & \langle \neg \text{holds}(p_6), \neg \text{holds}(p_5), \ldots \rangle
\end{bmatrix}
\end{bmatrix}
\]
The domain contribution S6 is an accepted (T7), correct application of the previously misconceived rule. By applying the correct rule the student shows that he has resolved the misconception that became apparent in utterance S4. The CG has now been realigned by adding \( \text{holds}(P(A) \cup P(B)) \subseteq P(A \cup B) \), and this information can be passed to the tutoring module.\(^5\)

5 Related Work

Jordan and Walker (1996) compare models of how agents in a collaborative setting decide to remind each other about salient knowledge, and argue for an approximate rather than detailed model of the attentional state. In tutoring, the decision to remind is further influenced by pedagogical strategies. Our model provides input to this decision making process, however, the decision itself is made by the tutoring module. For example, the contents of CG could be used to realise a Point-to-information hint which is part of the hint taxonomy proposed by (Tsoulvazi et al., 2004).

Baker et al. (1999) argue that learning from grounding is the basis of collaborative learning and Pata et al. (2005) show how student grounding acts serve to inform tutoring scaffolds. Intelligent tutoring systems, such as AutoTutor (Person et al., 2000) and Ms Lindquist (Heffernan and Koedinger, 2002), with simple dialogue models have no model of CG, but capture misconceptions using explicit buggy rules. In those systems, there is no clear separation between modeling the dialogue itself and modeling the tutoring task. The dialogue advances according to the local tutoring agenda. (Zinn, 2004) presents a dialogue-based tutoring system in which discourse obligations are generated from a store of task solution descriptions and the CG is maintained in the dialogue model. However, the choice of tutoring actions is not informed by the state of the CG, but rather is explicitly encoded.

6 Conclusion and Further Work

We presented a preliminary model of common ground for a domain where grounding propositional content is crucial. However in principle this model is general enough to be applied to other domains. The model takes into account Clark’s distinction between proposal and acceptance of dialogue contributions.

Indeed the current model is somewhat simplistic. There are a number of aspects of grounding which we observe in our corpus which this model does not account for but could be extended for, for instance domain content which is in a “pending” state when argumentation is taking place. Our further work will include extending the model to a larger set of dialogue moves including grounding acts. To obtain a more fine-grained model of context we need to further investigate what additional information about the problem-solving steps the domain reasoner can provide to the dialogue model, and thus to the tutoring manager. Furthermore we need a model of salience of propositions and steps in the problem-solving task, which may require a more flexible data structure. In a broader context it may be necessary to consider deletion of propositions however the conditions under which deletion rather than decay should occur need to be investigated. Current work includes implementation in TrindiKit.

References


