Fusing ISO 24617-2 Dialogue Acts and Application-Specific Semantic Content Annotations

Andrei Malchanau[1], **Volha Petukhova**[1], **Harry Bunt**[2]
[1] Spoken Language Group, Saarland University, Germany
[3] Tilburg Center for Cognition and Cmmunication, Tilburg University, The Netherlands
{andrei.malchanau; v.petukhova}@lsv.uni-saarland.de
harry.bunt@tilburguniversity.edu

Abstract

Accurately annotated data determines whether a modern high-performing AI/ML model will present a suitable solution to a complex dialogue application challenge, without wasting time and resources. The more adequate the structure of incoming data is specified, the more efficient the data can be interpreted and used by the application. This paper presents an approach to an application-specific dialogue semantics design which integrates the dialogue act annotation standard ISO 24617-2 and various domain-specific semantic annotations. The proposed multi-scheme design offers a plausible and a rather powerful strategy to integrate, validate, extend and reuse existing annotations, and automatically generate code for dialogue system modules. Advantages and possible trade-offs are discussed.

Keywords: interoperable annotations, dialogue acts, application semantics

1. Introduction

In context-update approaches to dialogue modelling, a *dialogue act* has two components: a *semantic content*, which describes the objects, properties, relations, or actions that the dialogue act is about, and a *communicative function*, which specifies how an addressee should update their information state with the semantic content. From 1980s, a number of dialogue act annotation schemes has been developed, ranging from simple lists of mutually exclusive tags to complex multi-layered taxonomies. Either used for the analysis of dialogue phenomena or to design dialogue systems, dialogue act annotation has for the most part been limited to marking up communicative functions.

In 2012, the ISO 24617-2 dialogue act annotation standard has been released, which presents a comprehensive multidimensional annotation scheme. The standard was also focused mostly on annotation of communicative functions, however, introduced the notion of type of semantic content - *dimension* - as a shallow characterisation of semantic content of the performed act, i.e. particular type of information state that is updated (ISO, 2012a). The annotation of semantic content is optional, since only task-related acts have full-fledged domain-specific semantic content, while dialogue acts performed for the purpose of dialogue control have marginal semantic content; the meaning of such a dialogue act is concentrated in its communicative function and dimension. In ISO 24617-2 2nd Edition (2019), a protocol is proposed to specify and integrate annotations of semantic content into dialogue act annotations as a 'plug-in', linking structures of the host annotation scheme to those of the plug-in scheme, see (Bunt, 2019).

Since a single annotation scheme that fully specifies the meaning of natural language dialogue contributions and has sufficient expressive capabilities to build efficient applications is challenging and maybe even not desirable for practical reasons, we deal in practice with multiple existing and newly defined annotation schemes that address different aspects of utterance meaning. Aiming to achieve an adequate coverage of the application-specific semantic content of dialogue acts, in this paper we present an approach that combines a general domain-independent scheme to represent annotations of functional aspects of dialogue contributions (viz. the ISO 24617-2 Dialogue Act Markup Language, DiAML) with multiple possible annotation schemes for representing application-specific semantic content.

Annotation efforts are labour intensive, therefore practical considerations along with theoretical clarity and soundness are important. Multiple schemes can be imported and included, and transformed to be re-usable for certain classes of applications. The schemes need to be explicitly defined and decisions concerning their fusion should be made prior to their use for annotation. The more explicitly dialogue act components are defined, the higher the interoperability level can be achieved and the more robust dialogue applications can be developed. The ISO 24617-2 standard does not prescribe content annotation schemes to be defined directly within a specific annotation design effort. In the following sections we consider the design of annotation schemes for applications of

ISA-20 Proceedings @LREC-COLING-2024, pages 122–132

various semantic complexity, discussing three use cases, and introduce the methodology for their integration with DiAML.

This paper is structured as follows. Section 2 discusses the DiAML annotation scheme and its XSD-based architecture. Section 3 discusses semantic content specifications for scenarios of various complexity, representing (1) intent and slot-filling; (2) term-based information retrieval; and (3) elaborate situation and experience modelling. Section 4 deliberates on the value of the proposed design for real-world applications, discussing advantages and possible trade-offs for system design and annotation work, its costs and its quality. Section 5 concludes the paper with observations on the experiences reported in preceding sections, and outlines directions for future development.

2. DiAML

The ISO 25617-2 dialogue annotation scheme has been designed according to the ISO principles of semantic annotation (Bunt, 2015) and has a three-part definition consisting of (1) an abstract syntax specifying the possible *annotation structures* as set-theoretical constructs; (2) a semantics specifying the meaning of the annotation structures defined by the abstract syntax; (3) a concrete syntax which specifies a representation format for annotation structures.

2.1. Abstract Syntax

The abstract syntax specifies a store of basic concepts, called the 'conceptual inventory'. The DiAML conceptual inventory consists of:

- a set of dimensions;
- a set of communicative functions;
- a set of qualifiers;
- a set of semantic and pragmatic relations for relating dialogue acts within a dialogue:
- a set of dialogue participants;
- primary data, segmented into markables.

Given a conceptual inventory, the abstract syntax specifies certain pairs, triples, and more complex nested structures made up from the elements of the inventory. Two types of structure are distinguished: *entity structures* and *link structures*. An entity structure contains semantic information about a segment of primary data, and is formally a pair ⟨m, s⟩ consisting of a markable and certain semantic information. A link structure contains information about the way segments of primary data are semantically related.

Formally, an entity structure in DiAML is a pair ⟨m, ⟨S, A, H, D, F, E, Q⟩⟩ consisting of a markable and a functional dialogue act structure, which is made up by seven components: (1) a sender (S), (2) one or more addressees (A), (3) zero or more other participants (H), (4) a dimension (D), (5) a communicative function (F), (6) zero or more dependence relations to a set (E) of other dialogue acts, and (7) zero or more qualifiers (Q), where the components H, E, and Q are not necessarily present.

A link structure in DiAML is a triple $\langle e, E, R\rangle$ consisting of an entity structure e, a set of entity structures E, and a relation R.

A full-blown annotation structure for a dialogue in DiAML is a set of entity (ϵ_i) structures and (link (L_j)) structures $\{e_1, ..., e_n, L_1, ..., L_k\}$.

2.2. Semantics

The DiAML semantics consists of the specification of a recursive interpretation function I_{DA} which, applied to a semantic content, forms an information state update operation. The DiAML semantics is compositional in the sense that the interpretation of an annotation structure is obtained by combining the interpretations of its component entity and link structures, see (Bunt, 2014) for details.

Semantic issues in using annotations from multiple schemes are addressed in (Bunt, 2024).

2.3. Concrete Syntax

The annotation structures defined by the DiAML abstract syntax can be represented in a variety of semantically equivalent ways, which can encode the structures of the abstract syntax. The official DiAML specification as part of the ISO 24617-2 standard includes a a reference representation format based on XML.

For the representation of entity structures an XML element <dialogueAct> is defined, with an attribute `@xml:id` whose value is a unique identifier; an attribute `@target`, whose value anchors the annotation in the primary data; and the following attributes: `@sender`, `@addressees`, `@other participants` (optional), `@dimension`, `@communicative function`, `@dependences` (optional), and `@qualifiers` (optional).

The XML elements <rhetoricalLink> and is defined for expressing representing rhetorical ('pragmatic') relations between dialogue acts, with the attributes `@dact`, `@rhetorelatum`, and `@relType`.

2.4. XSD Definition and Use

DiAML definitions are specified in the form of XSD schema files. XSD schemes provide 'namespaces' as a scoping mechanism for XML across multiple schemes and support their integration, inclusion and transformation. Unfortunately, neither homogeneous, nor heterogeneous nor chameleon design patterns (Costello, 2006; Ko and Yang, 2017) can fulfill the requirements of complex interoperable semantic annotation design. We propose a mixed-patterns approach that steers developers towards a clear data organization and the interoperability of annotations; in addition it enables formal validation of XML documents and automatic code generation to represent and use data from those XML documents inside an application. Code generation is important from a practical point of view; it supports the design of applications that are based on standard interoperable annotations.

The main definitions of the DiAML standard are stated in `DiAML_Types.xsd` and defined within `diaml namespace`. Auxiliary definitions are `namespace-less` concepts and are defined in `DiAML_Containers.xsd`. The main element <DialogueAct> is defined as follows[1]:

(1)
```
<xs:schema
targetNamespace=
   "http://www.iso.org/diaml"
xmlns:xs="http://www.w3.org/2001/
   XMLSchema"
xmlns:diaml="http://www.iso.org/
   diaml">
...
<xs:complexType name="DialogueAct">
<xs:attribute ref="xml:id"
   use="required"/>
...
</xs:complexType>
</xs:schema>
```

The semantic content of a dialogue act is defined at application level. For this purpose the <dialogueAct> element is re-defined in the application-specific scheme, e.g. for the DBOX project[2]:

(2)
```
<xs:schema
targetNamespace=
   "http://www.dbox.eu/content_spec"
xmlns:xs=
   "http://www.w3.org/2001/XMLSchema"
xmlns:diaml="http://www.iso.org/diaml"
xmlns:dbox=
   "http://www.dbox.eu/content_spec"
>
...
<xs:element name="dialogueAct">
<xs:complexType>
<xs:complexContent>
<xs:extension
base="diaml:DialogueAct">
<xs:sequence>
<xs:element
ref="dbox:semanticContent"
minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
...
</xs:schema>
```

[1]Note that here and elsewhere in the text XSD and XML examples are excerpts from complete schemes and documents, for reasons of space.

[2]https://www.lsv.uni-saarland.de/past-projects/d-box/

For convenience, `DiAML_Containers.xsd` scheme is included without a `namespace`:

(3)
```
<xs:include
schemaLocation="DiAML_Containers.xsd"/>
```

The scheme contains definitions of elements such as <diaml> which in turn contain sequences of dialogue acts and possibly some other elements:

(4)
```
<xs:element name="diaml">
<xs:complexType>
<xs:sequence>
<xs:element ref="dialogueAct"
minOccurs="1"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
```

Note that `DiAML_Containers.xsd` refers to <dialogueAct> elements defined in the application-specific scheme and not to the <DialogueAct> type in `DiAML_Types.xsd`. Since `Containers` have no `namespace` they are placed into the application `namespace`, e.g. `dbox`, following the *chameleon namespace design*.

The proposed architecture allows to bypass forward-referencing from `Containers` to application-specific <dialogueActs> with application-specific semantic content, while maintaining XML document verification and XML bindings code generation. The semantic content of a dialogue act is defined outside `diaml` and is represented as `<SemanticContent>` elements in the corresponding application-specific XSD, see examples in Section 3. Similar to DiAML_Types.xsd, further relevant schemes specifying semantic content can be included, for example, those developed with the ISO Semantic Annotation Framework (SemAF), see (Pustejovsky and Ide, 2017). This follows the *heterogeneous design pattern*.

Dialogue acts are included into `DiAML_Containers.xsd` in: (1) dialogue annotations defining participants, tokens, sounds and functional segments; (2) a corpus which consists of (3) dialogue sessions with reference to segmented primary data; and (4) messages exchanged between dialogue system modules.

Containers are supporting types and are not obligatory to use. However, they help maintain annotation consistency, and serve as examples to design one's own `Containers`, for instance, when different primary data representation formats are desired or other types of annotations need to be performed.

The architecture enables formal validation with standard tools like Oxygen[3] and automatic code generation with, for example, Java XML bindings (JAXB). Consider examples in (5) for automatically generated Java code for dialogue act and in (6) for a <diaml> element:

(5)
```
@XmlRootElement(name =
"dialogueAct")
public class DialogueAct
extends org.iso.diaml.DialogueAct
{
protected DboxSemanticContent
    DboxSemanticContent;
    /* Gets the value of the
    * DboxSemanticContent property.
    * @return possible object is
    * @link DboxSemanticContent
    */
public DboxSemanticContent
    getDboxSemanticContent() {
return DboxSemanticContent;
}
    /* Sets the value of the
    * DboxSemanticContent property.
    * @param value allowed object is
    * {@link DboxSemanticContent }
    */
public void setDboxSemanticContent(
    DboxSemanticContent value)
    {
    this.DboxSemanticContent = value;
   }
}
```

(6)
```
@XmlRootElement(name = "diaml")
   public class Diaml {
@XmlElement(required = true)
    protected List<DialogueAct>
    dialogueAct;
/**
    * Gets the value of the
    * dialogueAct property.
    * This accessor method
    * a reference to the live
    * list, not a snapshot. Therefore
    * any modification you make to the
    * returned list will be present
    * inside the JAXB object.
    */
public List<DialogueAct>
    getDialogueAct() {
    if (dialogueAct == null) {
    dialogueAct = new
    ArrayList<DialogueAct>();
    }
    return this.dialogueAct;
    }
}
```

It may be observed that the automatically generated code in (5) and (6) strictly follows XML element definition patterns from the specified XSD schemes. More specifically, `dbox DialogueAct class` extends `diaml DialogueAct class` by adding the `dbox semanticContent` field. `dbox Diaml class` contains a field of type `List<DialogueAct>`, where <DialogueAct> refers to `dbox <DialogueAct>` and not `diaml <DialogueAct>`, i.e. `Diaml` is the list of application-specific (`dbox`) dialogue act types, rather than generic `diaml` dialogue acts.

3. Application Schemes: use cases

This Section presents XML Schemes which capture semantics of the problem domain (Application Semantics) at conceptual level and represent it in XML schema definition language (XSD). We proceed from simple to more complex schemes featuring real use case scenarios.

3.1. Intents and Slot Filling

In the past few years, conversational AI agents have become extremely popular. Traditional conversational agents are often modeled based on *intents*,[4], which refers to the primary goal of a dialogue utterance. Intents are typically identified by analyzing the words and phrases in an utterance and mapping them to predefined categories or concepts. For example, an utterance like "What time are there trains from Norwich to York?" might be mapped to an intent like `request_depatureTime` where the first part corresponds to the communicative function of an utterance and the second part specifies a high-level semantic content, e.g. 'topic'. Additional entities are extracted to refine, modify and provide more context to the intent. For example, 'Norwich' and 'York', and specified

[3] https://www.oxygenxml.com/xml_editor.html

[4] Currently, intentless agents are claiming the ground, see https://rasa.com/blog/breaking-free-from-intents-a-new-dialogue-model.

as slot types of `departure_location` and `destination_location` respectively. Many task-oriented information-seeking dialogues are modelled this way (Larson and Leach, 2022).

Contrary to the traditional two-component intent definition, we break up intent specifications into two schemes: (1) a DiAML representation for a functional component; and (2) an Application Scheme for a semantic content for a particular domain. For example, the DBOX dialogues collected to design Question Answering Dialogue System (QADS) are modelled using this approach. Players ask questions about biographical facts of an unknown person in order to guess their identity. Questions are classified with their communicative function (e.g. Propositional, Check, Set and Choice Questions) and semantic content based on the Expected Answer Type (EAT). For the latter, 59 semantic relations between entities (e.g. between participants or between an event and participants) have been defined extending the Knowledge Base Population Slot Filling Task (TAC KPB, Min and Grishman (2012)). Each relation has two arguments and is one of the following types:

- RELATION(Z,?X), where Z is the person in question and X the entity slot to be filled, e.g. CHILD OF(einstein,?X);
- RELATION(E1, ?E2) where E1 is the event in question and E2 is the event slot to be filled, e.g. REASON(death,?E2); and
- RELATION(E,?X) where E is the event in question and X the entity slot to be filled, e.g. DURATION(study,?X).

The slots are categorized by the content and quantity of their fillers. Slots are labelled as name (person, organization, or geo-political entity), value (a numerical value or a date), or string. Slots can be as single-value (e.g. date of birth) or list-value (e.g. employers) based on the number of fillers they can take (Petukhova et al., 2018). Consider an excerpt from the DBOX XSD scheme:

```
  <xs:schema
xmlns:xsd=
   "http://www.w3.org/2001/XMLSchema"
targetNamespace=
    "http://www.dbox.eu"
xmlns="http://www.dbox.eu"
elementFormDefault="unqualified">
<xs:import namespace=
    "http://www.iso.org/diaml"
schemaLocation="DiAML_Types.xsd"/>
<xs:include schemaLocation=
    "DiAML_Containers.xsd"/>
<xs:simpleType name="eatRelation">
    <xs:restriction base="xs:token">
    <xs:enumeration value="origin"/>
...
    <xs:enumeration value="locBirth"/>
   </xs:restriction>
   </xs:simpleType>
...
<xs:simpleType name="SlotFiller">
   <xs:restriction base="xs:token">
   <xs:enumeration value="name"/>
   ...
   </xs:restriction>
</xs:simpleType>
...
<xs:simpleType name="GPE">
    ...
</xs:simpleType>
</xsd:schema>
```

A simple representation of semantic content can be defined as a list of attribute-value pairs as in 7.

(7) Player (P1): What country are you from?
System (P2): US

```
<dialogueAct xml:id="dap1TSK0"
      sender="#p1" addressee="#p2"
      dimension="task"
      communicativeFunction="setQuestion"
      target="#fsp1TSKCV0"
 <dbox:semanticContent>
     <entity xml:id="x1" target="#ne1"
      type="name" value="person"
      quantity="single"/>
      <entity xml:id="x2" target="#ne2"
      type="name" value="GPE"
      quantity="single"/>
      <eatRelation source="#x1"
      slotFiller="#x2" type="origin"/>
 </dbox:semanticContent>
</dialogueAct>

<dialogueAct xml:id="dap2TSK1"
     sender="#p2"  addressee="#p1"
     dimension="task"
     communicativeFunction="answer"
     target="#fsp2TSKCV1"
     functionalDependence="#dap1TSK0"
<dbox:semanticContent>
    <entity xml:id="x1" target="#ne1"
    type="name" value="person"
    quantity="single"/>
    <entity xml:id="x2" target="#ne2"
    type="name" value="US"
    quantity="single"/>
    <eatRelation source="#x1"
    slotFiller="#x2" type="origin"/>
 </dbox:semanticContent>
</dialogueAct>
```

Player asks the question concerning the *country* (markable $x2$, named entity $ne2$) of origin of the person in question(markable $x1$ assigned to *you*, named entity $ne1$). We expect an answer of relation type `ORIGIN(x1,?x2)` where $x1$ is the person whose identity need to be guessed and $x1$ the entity slot to be filled. A single slot filler is expected of type GPE, filled in answer with ‘US’.

3.2. Term-based Information Retrieval

The specification of semantic content may include elements from external knowledge bases or ontologies. For example, as a use case, we simulated pre-operative question answering sessions between doctors and patients. As a core part of these medical encounters, Patient Education Forms (PEFs) have to be filled in, and the patient's informed consent form signed. It is of chief importance that the forms are properly understood, and that medical procedures and risks are explained. PEFs contain many medical terms including some in Latin and some as abbreviations. These terms have to be detected and corresponding definitions retrieved from available medical documents. Thus, our approach was to detect medical terms, map them to entries of existing databases and ontologies, and retrieve definitions. For more information concerning the term extraction and application details see (Wolf et al., 2019; Bhatt, 2022).

There is a range of medical knowledge bases, ontologies, standard terminologies, and lexicons. One of the most widely used repositories of biomedical terms is the Unified Medical Language System (UMLS[5], Bodenreider (2004)), which integrates over 2 million names for 900 000 concepts from more than 60 families of biomedical vocabularies, as well as 12 million relations among these concepts. We used MetaMap[6] to find UMLS Metathesaurus concepts and to generate lexical variants of concept names. MetaMap gives a relevance score to each concept. In UMLS, similar terms (`biomedicalTerm`) from different vocabularies are grouped into the same concept (`umlsConcept`) and receive a Concept Unique Identifier (`umlsCUI`). Terms are grouped into semantic groups (`umlsSG`) and semantic types (`umlsST`) through which synonyms and related terms can be accessed. One of the vocabularies integrated into UMLS, which is frequently used for text simplification, is the Consumer Health Vocabulary (CHV, Zeng et al. (2007)), which comprises terms (`chvTerm`) for many common words and phrases used by health care professionals. Another frequently used vocabulary is SNOMED CT, Benson (2012) which consists of a large number of concepts (`snomedctConcept`) from clinical reports. We identified terms related to PEFs in SNOMED CT (84.9% - 95.9%) and in CHV (73.5% - 80.0%). Definitions were mostly retrieved from MedlinePlus[7], an online public health information resource (Schnall and Fowler, 2013).

UMLS concepts, matching CHV and SNOMED CT terms, are integrated with retrieved MedlinePlus definitions as part of semantic content of the BRENNDA (Business pRocess modEls iNtegration iNto Dialogue mAnagement) system (Tarakameh, 2019)):

```
  <xs:schema
xmlns:xsd=
   "http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.brennda.org"
xmlns="http://www.brennda.org"
elementFormDefault="unqualified">
<xs:import
    namespace="http://www.iso.org/diaml"
    schemaLocation="DiAML_Types.xsd"/>
<xs:include
   schemaLocation="DiAML_Containers.xsd"/>

  <xsd:complexType
name="semanticContent">
    <xsd:sequence>
    <xsd:element name="umlsConcept"
    <xs:attribute umlsCUI="xml:id"
    use="required"/>
<xsd:element name="snomedctConcept"
    <xs:attribute SCTID="xml:id"
    use="optional"/>
<xsd:element name="chvTerm"
    <xs:attribute SCUI="xml:id"
    use="optional"/>
<xsd:element
    name="medlineplusDefinition"
    <xs:attribute
    health-topicID="xml:id"
    use="required"/>
</xsd:sequence> </xsd:complexType>
</xsd:schema>
```

The application of this approach gives rise to the following dialogue fragment:

(8) Patient (P1): What is sleep apnea?
System (P2): It is a sleep-disordered breathing

```
<dialogueAct xml:id="dap1TSK13"
sender="#p1" addressee="#p2"
dimension="task"
communicativeFunction="setQuestion"
target="#fsp1TSKCV13"
    <brennda:semanticContent>
        <biomedicalTerm xml:id="biot21"
        target="#ne21"
        umlsConcept="sleepApnea"/>
    </brennda:semanticContent>
</dialogueAct>

<dialogueAct xml:id="dap2TSK27"
 sender="#p2"  addressee="#p1"
 dimension="task"
 communicativeFunction="answer"
 target="#fsp2TSKCV27"
 functionalDependence="#dap1TSK13"
   <brennda:semanticContent>
        <biomedicalTerm xml:id="biot21"
```

[5] https://www.nlm.nih.gov/research/umls/
[6] https://www.metamap.com.
[7] https://medlineplus.gov/

```
            target="#ne21"
            umlsConcept="sleepApnea"
            umlsCUI="C0018787"
            umlsSG="disorder"
            umlsST="diseaseOrSyndrom"
            chvTerm="sleep
                -disordered breathing"
            snomedctConcept="sleepApnea
                             disorder"
            medlineplusDefinition="sleep
                -disordered breathing"/>
        </brennda:semanticContent>
</dialogueAct>
```

In (8), patient (P1) filling in the PEF has difficulty to understand what 'sleep apnea' is. The system extracts the term from the patient's question ($ne21$) and queries the UMLS ontology for CHV and SNOMED CT concepts (synonyms) and the MedlinePlus definition. If a term was found in CHV, it was considered unnecessary to provide other synonyms in a generated answer.

3.3. From Situations to Experiences

Other applications may require richer semantic content to be incorporated in a dialogue semantics than illustrated above. This is, for example, the case of situated (or context-aware) human-computer interactions involving multiple human and artificial `participants` with certain `properties` performing various `roles`, dealing with, referring to, and reasoning about the world within a certain `environment/context` engaged in various `events` that take place in a certain `time` and `space`. The more complex the situation, the richer the content specification is required to describe it.

There have been numerous attempts to define context-aware interactions, most of which are very specific and provide too limited support for situation abstraction. Fully specified semantic representation is hardly possible, and sometimes not even desirable and feasible for maintaining experimental control. One of the most recently undertaken attempts to design annotations which support whole-sentence semantic representation is the Abstract Meaning Representation initiative (AMR, Banarescu et al. (2013)), with an extension for dialogue semantics (Dialogue-AMR, Bonial et al. (2020)). Within this framework, human-robot dialogues are annotated with a speech act[8], tense (`before, now, after`), aspect (`stable, ongoing, complete, habitual, completable`) and semantic role information (PropBank, Palmer et al. (2005)); see example (9) adopted from (Bonial et al., 2020).

[8] The designed tagset to model human-robot interactions comprises a list of 14 mutually exclusive tags.

(9) Commander (C2): Drive to the door

```
(c / command-SA
   :ARG0 (c2 / commander)
   :ARG2 (r / robot)
   :ARG1 (g / go-02 :completable +
   :ARG0 r
   :ARG3 (h / here)
   :ARG4 (d/ door)
   :time (a2 / after
   :op1 (n / now))))
```

To parse and generate DialAMRs, AMR parsers and resources are used, which are steadily growing in number and scope (Zhou et al., 2021; Cheng et al., 2022; Vasylenko et al., 2023).

In the past few years, a number of ISO SemAF annotation schemes have been developed, besides DiAML: Time and Events (ISO, 2012b), Semantic Roles (ISO, 2014), Semantic Relations in Discourse (ISO, 2016), Coreference (ISO, 2019b), Spatial Information (ISO, 2019a) and Quantification (ISO, 2019c). It would be very attractive to include these schemes for modeling situated interactions. An elegant way to incorporate SemAF annotations into dialogue act annotations has been proposed by Bunt (2019), using *annotation schema plug-ins* which make use of a variety of content link structures, e.g. `contentLink` and `emoLink`, for importing elements of one annotation schema into another. Multiple SemAF schemes can be used for content representation by means of the *interlinking* technique (Bunt, 2024).

Dialogue participation involves a range of social and emotional experiences. Human interactions are more than the exchange of information, decision making, or problem-solving; they involve a wide variety of aspects related to feelings, emotions, social status and interpersonal relations.

For developing socially embedded dialogue systems, it has been proposed to model interactive behaviour in terms of *experiences*, i.e. instances of mental states or dialogue context/states (Stevens et al., 2016; Malchanau et al., 2018). Dialogue participants collect interactive experiences and learn from them. An instance may encode all information that influences the interpretation and generation of dialogue contributions, and thus the decision making process: knowledge about domain and partners, participants' preferences and attitudes, emotional state and social status, and this list is far from exhaustive. Although there are no theoretical limitations on instance size, the application efficiency is the highest when the state representation is relatively compact. A very complex state representation may make state tracking and instance retrieval very costly. There should be no problem with using incomplete instances, since humans also have to deal with partially available, am-

Holder: slot type	Possible Values
doctor: strategy	competence | warmth
doctor: expertise	low | moderate | high
doctor: importance	low | moderate | high
doctor: framing-effect	threat | risk | benefit
doctor: preference	(im)possible | (un)desired | (in)abile mandatory | urgent
patient: strategy	avoiding | hesitant | submissive | biased | cooperative | aggressive | resistant
patient: expertise	low | moderate | high
patient: importance	low | moderate | high
patient: framing-effect	threat | risk | benefit
patient: preference	(im)possible | (un)desired | (in)abile mandatory | urgent
patient: readiness	low | moderate | high

Table 1: Instance contents concerning participants' strategies and preferences.

biguous and/or vague information, imperfect understanding and limitations of working memory.

We designed instance-based LICA[9] agents that are involved in doctor-patient interactions, where an imbalance is observed in the knowledge and relationship between interlocutors, due to social, professional and personal factors. Agents simulate patients of different personalities, motivational and emotional dispositions. Interacting with LICA agents, doctors are trained to identify strategies that are optimal for specific patients, i.e. positively affect patient's preferences for a certain treatment.

Some important strategies concern pragmatic aspects such as use of indirect speech acts for politeness or to express interest, respect, support and empathy; or qualified functional aspects concerning affected behaviour in order to build a trustful relationship through the development of rapport and responsiveness to a patient's emotions ('Appeal to Warmth', (Fiske, 2018). Other strategies concern the quality of arguments presented in health intervention utterances ('Appeal to Competence'): (1) information provided, e.g. expert language use and appeal to authority; (2) attitudes towards proposed interventions and its outcomes: costs, appeal to importance and call for readiness; and (3) targeted framing effects, e.g. presentation of options in positive or negative terms (survival rates or mortality rates for a treatment).

Both relevant functional aspects and semantic content are encoded in an instance represented as a set of slot-value pairs. Table 1 presents a template encoding beliefs concerning domain knowledge, the participants' preferences, and the persuasion strategy being pursued.

The domain selected for our use case concerns the treatment of diabetes. To generate health interventions of various types, medical claims and evidence were collected from PubMed abstracts[10], viz. 32 claims and 64 supporting and attacking evidence statements. Keywords and phrases were extracted using the KeyBERT model, (Grootendorst, 2020) and the term banks UMLS and CHV were queried to compute the level of expertise, the framing effects, and the applied strategy. Importance, readiness, preference and framing effects were modulated. On the basis of previous research (Guenoun and Zlatev, 2023; Lapina and Petukhova, 2017), features were selected for linguistic modulations. These concern *appeal* (competence/warmth), *text length* (long/short), *framing* (risk/benefit), *lexical complexity* (complex/simple), *concreteness* (numbers/textual delivery) and *grammatical voice* (passive/active) (Wan Ching Ho and Petukhova, 2024).

Below is an excerpt from the XSD scheme specifying LICA semantic content; a dialogue fragment example which makes use of LICA content specifications is presented in (10) of Appendix 7:

```
  <xs:schema
xmlns:xsd=
    "http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.lica.org"
xmlns="http://www.lica.org"
elementFormDefault="unqualified">

  <xs:import
   namespace="http://www.iso.org/diaml"
   schemaLocation="DiAML_Types.xsd"/>
<xs:include schemaLocation=
    "DiAML_Containers.xsd"/>

  <xs:simpleType name="Holder">
    <xs:restriction base="xs:token">
    <xs:enumeration value="doctor"/>
    <xs:enumeration value="patient"/>
    </xs:restriction>
</xs:simpleType>

  <xs:simpleType name="Strategy">
    <xs:restriction base="xs:token">
    <xs:enumeration value="competence"/>
    ...
    <xs:enumeration value="hesitant"/>
    </xs:restriction>
</xs:simpleType>
...
<xs:simpleType name="Readiness">
    <xs:restriction base="xs:token">
    <xs:enumeration value="low"/>
    <xs:enumeration value="moderate"/>
    <xs:enumeration value="high"/>
    </xs:restriction>
</xs:simpleType> </xsd:schema>
```

4. Value for Real-World Applications

The multi-scheme design presented in this paper has a number of advantages, as well as limitations. One of the advantages is that such a design splits up large annotation efforts into small(-

[9]Learning Intelligent Conversational Agents.
[10]https://pubmed.ncbi.nlm.nih.gov/

er) tasks that are more manageable for human annotators and automatic labeling systems. This positively affects annotation quality and costs: it increases annotation consistency and accuracy, it improves scheme usability in terms of inter-annotator agreement, and it potentially decreases annotation time[11].

Another advantage is that task-specific annotations can be straightforwardly reused by other applications. For instance, labeled data can be used for adaptation or knowledge distillation of pre-trained large models, which significantly improves their performance on a variety of up-/downstream tasks. Applications based on clear use-case semantics are easier to evaluate and their performance can be directly compared to other existing systems or models.

A limitation of this approach is that semantic information that is not captured in annotations needs to be modelled inside an application and often remains somewhat hidden. This is for instance the case of hidden layers as used in modern neural systems that are responsible for learning intricate structures in data which are not explicitly annotated. This makes neural networks a powerful but black-boxed tool with limited explainability and interpretability of the system's behaviour. Another limitation is that the collection of semantic information from multiple annotation projects runs into danger to be less interoperable and challenging to fuse.

Advantages and limitations put the designer in a position to carefully weight pros and cons for their design scenarios, with trade offs between semantic expressiveness and precision on the one hand and simplicity of its application on the other.

5. Discussion and Conclusion

This paper explores dialogue use cases of varied semantic complexity: slot-filling supporting question answering, term-based information retrieval, and complex situation and experience specifications. Functional aspects of dialogue contributions are modelled using the ISO 24617-2 dialogue act annotation standard and specified in DiAML. Semantic content is represented in an appropriate way for a specific dialogue application.

Applications require an interpretation framework, either utilising explicit knowledge representation techniques or relying on an intuitive interpretation scattered implicitly across application code. Specifying annotation schemes for semantic content in a formal way, e.g. in XSD format, opens opportunities to share annotations among different applications and tools.

In this paper we have proposed a way to integrate a wide range of domain/application-specific annotations with the domain-independent ISO 24617-2 scheme specified in DiAML. The ISO annotation standards developed within SemAF can be integrated in a similar manner. For all components, XML schema definitions (XSD) refer to external XML schemes. More than one XML schema can be included or imported within an XML schema, as we showed using 'namespaces'. XSD has the important advantage that it can be used to validate the contents of an XML document, as well as to generate code within an application design.

We will distribute the designed XSD domain-independent DiAML and Application Schemes on the DialogBank[12], a collection of dialogues annotated according to ISO 24617-2 standard. A full package of gold standard dialogue act annotations, XSD schemes, primary data, and documentation is available for the Metalogue Multi-Issue Bargaining Corpus in the LDC catalogue.[13]

6. Acknowledgments

The authors are also very thankful to anonymous reviewers for their valuable comments.

7. Bibliographical References

Laura Banarescu et al. 2013. Abstract meaning representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*.

Tim Benson. 2012. *Principles of health interoperability HL7 and SNOMED*. Springer Science & Business Media.

Abhinav Bhatt. 2022. Personalized medical arguments generation. Master's thesis, Saarland University.

Olivier Bodenreider. 2004. The Unified Medical Language System (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Claire Bonial et al. 2020. Dialogue-AMR: abstract meaning representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*.

[11] For example, the Real Time Factor (RTF) can be estimated - amount of time spent on annotations given the amount of dialogue data. RTF 10 means that an annotator spent 100 minutes annotating 10 minutes of real dialogue, e.g. speech and video.

[12] https://dialogbank.lsv.uni-saarland.de/

[13] https://catalog.ldc.upenn.edu/LDC2017S11

Harry Bunt. 2014. Annotations that effectively contribute to semantic interpretation. In *Computing Meaning: Volume 4*, pages 49–69. Springer.

Harry Bunt. 2015. On the principles of semantic annotation. In *Proceedings of the 11th Workshop on Interoperable Semantic Annotation (ISA-11)*.

Harry Bunt. 2019. Plug-ins for content annotation of dialogue acts. In *15th Workshop on Interoperable Semantic Annotation (ISA-15)*.

Harry Bunt. 2024. Combining semantic annotation schemes through interlinking. In *20th Workshop on Interoperable Semantic Annotation (ISA-20)*.

Ziming Cheng, Zuchao Li, and Hai Zhao. 2022. Bibl: AMR parsing and generation with bidirectional bayesian learning. In *29th International Conference on Computational Linguistics*.

R.L. Costello. 2006. Ml schemas: Best practices. multi-schema project: Zero, one, or many namespaces?

Susan T Fiske. 2018. Stereotype content: Warmth and competence endure. *Current directions in psychological science*, 27(2):67–73.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Bushra S Guenoun and Julian J Zlatev. 2023. Sending signals: Strategic displays of warmth and competence. *Working Paper 23-051*.

ISO. 2012a. *Language resource management – Semantic annotation framework – Part 2: Dialogue acts. ISO 24617-2*. ISO Central Secretariat, Geneva.

ISO. 2012b. *Language Resource Management-Semantic Annotation Framework (SemAF)- Part 1: Time and events. ISO 24617-1*. ISO Central Secretariat, Geneva.

ISO. 2014. *Language Resource Management- Semantic Annotation Framework (Se mAF)- Part 4: Semantic roles ISO 24617-4*. ISO Central Secretariat, Geneva.

ISO. 2016. *Language resource management - Semantic annotation framework (SemAF) Part 8, Semantic relations in discourse ISO 24617-8*. ISO Central Secretariat, Geneva.

ISO. 2019a. *Language resource management - Semantic annotation framework- Part 7: Spatial information (ISO-Space). ISO 24617-7*. ISO Central Secretariat, Geneva.

ISO. 2019b. *Language Resource Management-Semantic Annotation Framework (SemAF)- Part 9: Reference annotation framework (RAF) ISO 24617-9*. ISO Central Secretariat, Geneva.

ISO. 2019c. *Language Resource Management: Semantic Annotation Frame work (SemAF)-Part 12: Quantification. ISO 24617-12*. ISO Central Secretariat, Geneva.

Hye-Kyeong Ko and Minho Yang. 2017. An effective xml schema conversion technique for improving xml document reusability using pattern list. *International Journal of Internet, Broadcasting and Communication*, 9(2):11–19.

Valeria Lapina and Volha Petukhova. 2017. Classification of modal meaning in negotiation dialogues. In *Proceedings of the 13th Joint ISO-ACL Workshop on Interoperable Semantic Annotation (ISA-13)*.

Stefan Larson and Kevin Leach. 2022. A survey of intent classification and slot-filling datasets for task-oriented dialog.

Andrei Malchanau, Volha Petukhova, and Harry Bunt. 2018. Towards integration of cognitive models in dialogue management: designing the virtual negotiation coach application. *Dialogue & Discourse*, 9(2):35–79.

Bonan Min and Ralph Grishman. 2012. Challenges in the knowledge base population slot filling task. In *LREC*, pages 1137–1142.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Volha Petukhova et al. 2018. Understanding questions and extracting answers: Interactive quiz game application design. In *Human Language Technology. Challenges for Computer Science and Linguistics*, pages 246–261. Springer.

James Pustejovsky and Nancy Ide, editors. 2017. *Handbook of Linguistic Annotation*. Springer.

Janet G Schnall and Susan Fowler. 2013. Medlineplus. gov: quality health information for your patients. *AJN The American Journal of Nursing*, 113(9):64–65.

Christopher Stevens, Harmen de Weerd, Fokie Cnossen, and Niels Taatgen. 2016. A metacognitive agent for training negotiation skills. In *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM 2016)*.

Bahar Tarakameh. 2019. Integration of business process models into dialogue management: patient self-assessment use case. Master's thesis, Saarland University.

Pavlo Vasylenko et al. 2023. Incorporating graph information in transformer-based amr parsing. In *Findings of the Association for Computational Linguistics: ACL 2023*, page 1995–2011.

Clara Wan Ching Ho and Volha Petukhova. 2024. Towards generation of personalised health intervention messages. In *Proceedings of the 1st Workshop on Patient-oriented Language Processing (CL4HEALTH), LREC-COLING 2024)*.

Martin Wolf et al. 2019. Term-based extraction of medical information: Pre-operative patient education use case. In *Conference on Recent Advances in Natural Language Processing*.

Qing Zeng et al. 2007. Term identification methods for consumer health vocabulary development. *Journal of medical Internet research*, 9 (1).

Jiawei Zhou et al. 2021. Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Appendix: LICA dialogue fragment

(10) Doctor (P1): You should minimise alcohol intake
Doctor (P1.1): Alcohol intake may place people with diabetes at increased risk for delayed hypoglycemia
Doctor (P1.2): Persons using insulin or insulin secretagogues can experience delayed nocturnal or fasting hypoglycemia after alcohol consumption.
Doctor (P1.3): Moderate alcohol consumption has minimal acute and/or long-term detrimental effects on glycemia with type 1 or type 2 diabetes.

```
<dialogueAct xml:id="dap1TSK1"
sender="#p1" addressee="#p2"
dimension="task"
communicativeFunction="inform"
target="#fsp1TSKCV1"
    <lica:semanticContent>
        <claim xml:id="claim1"
        target="#fsp1TSKCV1"
        topic="alcohol intake"
        preference="mandatory"
    </lica:semanticContent>
</dialogueAct>


<dialogueAct xml:id="dap1TSK1.1"
sender="#p1" addressee="#p2"
dimension="task"
communicativeFunction="inform"
target="#fsp1TSKCV1.1"
    <lica:semanticContent>
        <evidence xml:id="evidence1.1"
        target="#fsp1TSKCV1.1"
        relation="#claim1"
        stance="support"
        topic="alcohol intake"
        expertise="moderate"
        importance="high"
        preference="mandatory"
        framing="risk"
        strategy="competence"
    </lica:semanticContent>
</dialogueAct>
<rhetoricalLink dact="#dap1TSK1.1"
rhetoRelatum="#dap1TSK1"
relType="justification"/>

<dialogueAct xml:id="dap1TSK1.2"
sender="#p1" addressee="#p2"
dimension="task"
communicativeFunction="inform"
target="#fsp1TSKCV1.2"
    <lica:semanticContent>
        <evidence xml:id="evidence1.2"
        target="#fsp1TSKCV1.2"
        relation="#claim1"
        stance="support"
        topic="alcohol intake"
        expertise="high"
        preference="mandatory"
        importance="high"
        framing="risk"
        expertise="high"
        strategy="competence"
    </lica:semanticContent>
</dialogueAct>
<rhetoricalLink dact="#dap1TSK1.2"
rhetoRelatum="#dap1TSK1"
relType="justification"/>

<dialogueAct xml:id="dap1TSK1.3"
sender="#p1" addressee="#p2"
dimension="task"
communicativeFunction="inform"
target="#fsp1TSKCV1.3"
    <lica:semanticContent>
        <evidence xml:id="evidence1.3"
        target="#fsp1TSKCV1.3"
        relation="#claim1"
        stance="support"
        topic="alcohol intake"
        expertise="high"
        preference="mandatory"
        importance="moderate"
        framing="risk"
        expertise="high"
        strategy="competence"
    </lica:semanticContent>
</dialogueAct>
<rhetoricalLink dact="#dap1TSK1.3"
rhetoRelatum="#dap1TSK1"
relType="justification"/>
```