Robot Personality: Representation and Externalization

Stasinos Konstantopoulos [1] and **Vangelis Karkaletsis** [1] and **Colin Matheson**[2]

Abstract. One of the greatest challenges in current human-computer interaction research is the development of natural and affective interfaces. We present personality modelling as the inference of emotion and affect parameters, used by the dialogue management and user interaction components of a robot architecture. We discuss in detail the personality representation and inference aspects of the architecture and also present how the inferred parameters modulate multiple user interface modalities, such as speech and facial expressions.

1 INTRODUCTION

An enduring challenge in human-computer interaction research is the creation of natural and affective interfaces. This involves the development of a computer system that is capable of perceiving and correctly interpreting human behaviour beyond the literal interpretation of an utterance, as well as acting in ways familiar to humans.

We present advanced natural dialogue capabilities which facilitate human-computer interaction, involving multiple modalities such as spoken natural language, gestures, and facial expressions. While the emphasis of the work described here is a personality modelling architecture for generating natural descriptions of objects in a particular domain, there is nothing inherently limiting the application of the modelling architectural to this particular use case.

1.1 Application and Use Case

The personality modelling architecture presented here is being developed in the context of the INDIGO project, which aims to advance human-robot interaction methodology and technology, enabling robots to perceive natural human behaviour, as well as making them act in ways that are more familiar to humans. The INDIGO technologies are integrated in a robot platform which can interact with humans using gestures and spoken natural language, particularly for the purpose of describing its physical surroundings.

The INDIGO platform includes advanced laser scanning and navigation capabilities allowing it to operate for long periods in crowded environments. For interacting with humans, it incorporates robust speech and gesture recognition components for detecting user actions. The robot can dynamically generate and synthesise spoken language responses, but also demonstrate facial expressions via either an animatronic head or a virtual animated face on a display mounted on top of the mobile robot platform.

The INDIGO application is being developed in the field of cultural informatics, configuring the robot as a tour guide interacting with the museum visitors. The robot does not simply follow a pre-determined tour reading out canned exhibit descriptions, but gives visitors the opportunity to influence the tour, asking for more information about exhibits they find interesting or cutting short parts of the tour that they are less interested in; all descriptions are dynamically generated, avoiding the repetition of material, unless drawing a comparison, and weighing various factors to adapt the description to the audience. The material presented is also tailored both to suit the visitor type (child or adult, for instance), and also depends on the interaction history.

In this context, the new approach to personality and emotion modelling in dialogue discussed here is used to influence the output modalities to achieve a more natural interaction with the user. This manifests itself as:

- the same robot behaving differently in different parts of the tour, as different exhibits carry different emotional load and connotations; and,
- different robots behaving differently in the same part of the tour, as both the emotional load and the way it surfaces varies between robots with different personalities.

The INDIGO use case is guiding visitors through an exhibition on the ancient Agora of Athens, introducing the buildings to them before they attend a virtual 3D tour of the Agora hosted at the Foundation of the Hellenic World.[3] The examples used in this paper are drawn from this domain.

1.2 Overview

The underlying knowledge representation formalism for domain knowledge and personality modelling is discussed in Section 2. This is used by a *robot personality* component (Section 3) that infers affect parameters related to a given object in the knowledge base.

Several other components involved in the interaction cycle are briefly described in Section 4 and also shown in Figure 1. In short, user actions (utterances and gestures) are analysed by an array of language and vision tools. The results of the analysis are fused and passed to the *dialogue and action manager* (DAM), which is responsible for generating a robot action in response. The results of the linguistic analysis, in particular, are also passed to robot personality, which calculates the emotional appraisal of the user's utterance.

If the robot action is to be the description of another object, the DAM consults the robot personality module to receive affect parameters associated with possible objects to describe (what comes next in the tour scenario, what the user requested, and so on). These are used by the DAM to make the final decision about the object to be describe, and this choice also generates an emotional appraisal.

[1] Institute of Informatics & Telecommunications, NCSR 'Demokritos', Greece, email: {konstant,vangelis}@iit.demokritos.gr
[2] Human Communication Research Centre, Edinburgh University, U.K. Email: Colin.Matheson@ed.ac.uk

[3] See http://www.tholos254.gr/projects/agora/en/ for the virtual tour, http://meeting.athens-agora.gr/ for the exhibition, and http://www.fhw.gr/ for the Foundation in general.

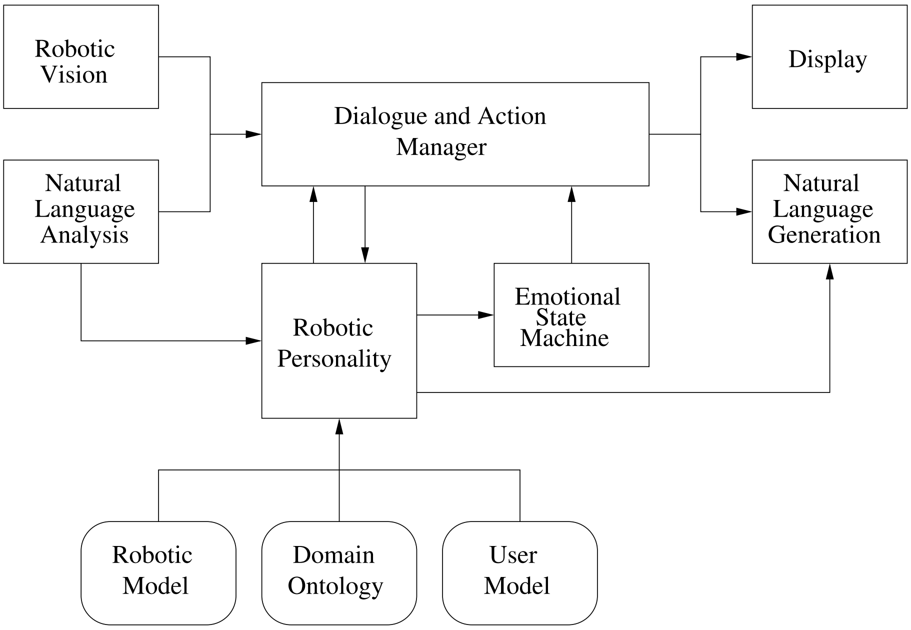

Figure 1. INDIGO system overview. Rounded boxes represent knowledge structures and square ones processes.

Both the user-action and robot-action appraisals are forwarded to the *emotional state machine* (Section 4.3), which updates the *mood* and *emotional state* of the robot and results in an annotation of the robot action. This annotation is needed by speech synthesis and by the animated face or animatronic head to modulate the voice and facial expression of the robot.

Depending on what the robot action is, it is then realized by the natural-language generation and speech synthesis (spoken utterance), the display (animated face expression or textual information), the robot platform (movement), the head (facial expression and/or gesture), or a coordinated combination thereof. Natural-language generation in particular receives from robot personality abstract information that is used to realize the object description.

2 REPRESENTATION AND MODELLING

The domain of expertise of INDIGO robots is the buildings of the ancient Agora, as mentioned above. Knowledge of the domain is represented as an OWL ontology. The *OWL Web Ontology Language* is a W3C recommendation for representing ontologies within the *Resource Description Framework* (RDF).

2.1 RDF and OWL

The *Resource Description Framework* (RDF) [4] is a knowledge representation technology built around subject-predicate-object triples of abstract *resources*. RDF triples assign the *object* resource as the value for the *predicate* property of the *subject* resource. So, for example, the RDF triple:

```
stoa_attalus style pergamene .
```

asserts that the resource `stoa_attalus` has a style property, and that this property has a value of `pergamene`.

RDF Schema (RDFS) defines a vocabulary for asserting that an *instance* resource is a member of a *class* resource, and for placing classes in a subsumption hierarchy, so one can say that `stoa_attalus` is of type Stoa, which is a kind of Building, a kind of ArchitecturalConstruction:

```
stoa_attalus rdfs:type Stoa .
Stoa rdfs:subClassOf Building .
Building rdfs:subClassOf
   ArchitecturalConstruction .
```

from which it follows implicitly that `stoa_attalus` is also has the type ArchitecturalConstruction.

The *OWL Web Ontology Language* [13] is an ontology representation language. OWL defines a vocabulary for expressing logical statements using the logical connectives for conjunction, disjunction, negation, implication, and equivalence as well as quantification over the objects of RDF predications. So, for example, this property restriction:

```
ArchitecturalConstruction
   rdfs:subClassOf PhysicalObject .
ArchitecturalConstruction
   rdfs:subClassOf anon1 .
anon1 rdf:type owl:Restriction .
anon1 owl:onProperty hasArchitect .
anon1 owl:allValuesFrom Architect .
Architect rdfs:subClassOf Person .
```

asserts that the values of the `hasArchitect` properties of `ArchitecturalConstructions` must be members of the `Architect` sub-class of `Person`. In the remainder we will not show anonymous instances, like `anon1` used here to group together the `owl:onProperty` and `owl:allValuesFrom` triples, and present similar groupings with brackets.

2.2 Domain Modelling

The OWL specification defines three increasingly expressive sub-languages, *OWL Lite*, *OWL-DL*, and *OWL Full* [13, Section 1.1]. The INDIGO ancient Agora ontology is within OWL-DL, which is the sub-language covered by *Description Logic reasoning* (see also Section 3).

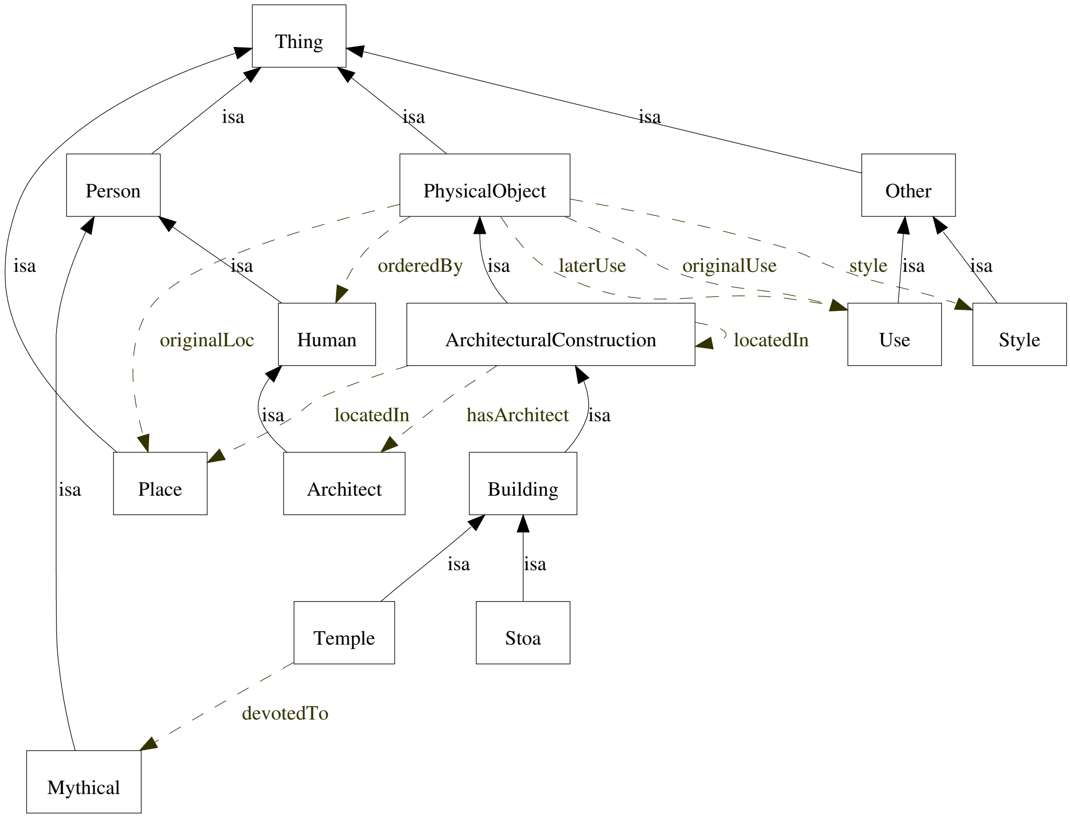

Figure 2. Top level and some characteristic classes of the Agora ontology. The dashed arrows run from the domain to the range class of properties.

The main class of the Agora ontology is PhysicalObject, which contains all exhibits. It comprises two sub-classes, ArchitecturalConstruction and ArtObject. The former has been further refined, as it is more pertinent to the INDIGO use case, and subsumes concepts like Building, Altar, Monument, etc. Buildings are further subcategorized as Stoa, Temple, Library, and so on.

In addition to the main PhysicalObject concept, the ontology defines four top-level concepts used to describe the objects of the former: Person, Time, Place, and Other. The Person class includes both real people and mythical characters. The former are architects, artists, sponsors, and, in general, people involved in the creation of a physical object. The latter are entities depicted in otherwise referred to by physical objects.

The Time and Place sub-ontologies are used to provide geotemporal information on physical objects, such as creation time, original and current location, and so on. They are abstract references, for example `hellenistic_period` or `west_of_agora`, and no inferences can be drawn about their relative location. The rest of the descriptors of physical objects, such as Style, Condition, Use, and Material, are subsumed by the Other top-level concept.

The individual exhibits of PhysicalObject have a large number of properties linking them to descriptors in the other concepts. The domain and range of these properties has been appropriately restricted, implementing reasonable 'sanity checks'. So, for example, devotedTo may only link Temple or Altar instances to Person instances, hasArchitect may only link ArchitecturalConstruction instances to Architect instances, and so on.

In total, the Agora ontology defines 36 classes and 18 properties. At its current state of development, it is instantiated with 94 PhysicalObject instances, described by five Person instances, thirteen Time instances, eight Place instances, and twenty-six Other instances. Figure 2 gives a graphical representation of the top level classes and of some of the classes at deeper levels which are used in our examples.

2.3 User and Robot Modelling

User and robot modelling is based on the RDF schema used by the NATURALOWL generation engine (cf. Section 4.2) and the ELEON authoring tool (cf. Section 2.4). This schema assigns a numerical attribute to properties, individuals, and classes in the domain. In the context of NATURALOWL, this attribute is interpreted as user preference and influences the decision of which ontological elements to draw information from when generating an object's description. ELEON supports NATURALOWL, providing an authoring user-interface within which initial preference values can be provided for the various types of users (lay-person, domain expert, child); these initial user-preference parameters are refined in the course on the interaction.

So, for example, the following RDF statement:[4]

```
style rdf:type Property .
style hasAttribute {
  forUserType Child
  Interest 3 } .
style hasAttribute {
```

[4] The example shown here is simplified for the purpose of focusing the presentation of interest attribution; actual `Property` elements in ELEON and NATURALOWL models comprise several other properties that are not pertinent to the work described here.

```
  forUserType Expert
  Interest 5 } .
```

expresses that the `style` property is attributed with an interest level of 3 for children and 5 for experts. `Property` elements can be further elaborated to be applicable only to properties of instances of a particular class. Consider, for example, the following statement:

```
style rdf:type Property .
style hasPropertyAttribute {
  forUserType Expert
  forOwlClass Temple
  Interest 5 } .
style hasPropertyAttribute {
  forUserType Expert
  forOwlClass Stoa
  Interest 3 } .
```

where `dom` is the namespace of the domain ontology. This fragment states that experts find it more interesting to know the architectural style when discussing temples than when discussing stoas.

In INDIGO, we have extended this user modelling mechanism to describe arbitrary robot attributions to domain objects: a robot profile is essentially identical to a user profile, except that any arbitrary numerical attribution can be provided and not only `Interest`. ELEON is accordingly extended to allow the addition of new attributes. Furthermore, attributes can refer to classes and instances and not only properties.

To demonstrate the above, consider the following robot profile:

```
Temple hasClassAttribute {
  forUserType robot2
  Interest 1 } .
Stoa hasClassAttribute {
  forUserType robot2
  Interest 4 } .
stoa_attalus hasIndividualAttribute {
  forUserType robot2
  Interest 6 } .
style hasPropertyAttribute {
  forUserType robot2
  forOwlClass Temple
  Interest 5 } .
style hasPropertyAttribute {
  forUserType robot2
  forOwlClass Stoa
  Interest 2 } .
```

which is interpreted as a preference for talking about stoas, and in particular `stoa_attalus`; and, furthermore, a reluctance to use `style` to describe stoas, as the latter is more interesting when used to describe temples.

2.4 Authoring

ELEON (Editor for Linguistically Enriched ONtologies) [3] is an ontology editor specifically aimed at creating ontologies that will be used to generate descriptions. It extends the authoring tool described by Androutsopoulos et al. [1].

Using ELEON, persons that have domain expertise but no technological expertise can easily manipulate the domain ontology and the user and robot models, and also author the *micro-plans* used by generation engines to realize concrete descriptions from abstract ontological information.

ELEON presents the domain ontology to authors as a hierarchy of *entity types* (concepts), where *fields* (properties) can be added. When editing an *entity* (individual), the author can add values to the fields of the individual's type and all its super-types. The editor also allows the stipulation of logical restrictions, such as constraining the range of a property's values or the number of values it can have. For example, the author can constrain the locatedIn property to a single value of type Place, while allowing any number of hasArchitect values of type Human. All logical constructs of OWL Lite are accessible through the ELEON user interface, as well as the majority of OWL-DL constructs.

ELEON also allows authors to specify user profiles ('expert', 'child', etc) as well as robot profiles. For each such profile, several parameters such as the desired length of the texts and paragraphs are specified. The ability to aggregate clauses to form longer sentences is also enabled. In addition, for a user type or a robot model, authors can specify how *interesting* each entity type or field is for that user type or robot model.[5]

The author must also create a linguistic model that is used to realize concrete descriptions from abstract ontological knowledge. This comprises a lexicon of surface forms, containing the nouns that realize types and entities and the verbs that realize relations. The authors enter the base forms of the nouns and verbs they wish the system to use, and there are facilities to generate the other forms automatically. [6] Noun entries are linked to entity types and entities, to allow, for example, the system to generate referring noun phrases. For each field of the ontology and each language, the authors have to create at least one clause plan (micro-plan) which specifies how the field can be expressed in that language. The author specifies the clause to be generated in abstract terms, by stating, for example, the verb to be used (from the domain-dependent lexicon), the voice and tense of the resulting clause, and so on.

ELEON exports the domain ontology as OWL and the linguistic, user and robot modelling annotations as RDF statements, as explained above. The NATURALOWL natural language generation engine (cf. Section 4.2) uses these resources to generate descriptions. Work is underway to adapt the METHODIUS NLG engine,[7] to the current formats, as the latter uses a similar micro-plan format developed on the M-PIRO project [9].

One of the most interesting and helpful features of ELEON is that the authoring environment directly displays *previews* of the descriptions. These are created by invoking NATURALOWL from within ELEON. The user can choose a specific entity from the ontology and preview the text produced for the entity in question. When the user makes some modifications, either in the ontology, in the user and robot profiles, or in the linguistic resources, ELEON exports these elements in the background, invokes NATURALOWL for the specified language [6] and requests the description for the selected entity and user and robot profiles (Figure 3).

Another feature of ELEON for supporting ontology authoring is to display error or warning messages in case of logical contradictions, and especially violations of axiomatic restrictions on value cardinality, domain, and range of fields. In order to achieve this, ELEON in-

[5] It should be noted that the robot personality component can use a wider variety of attributes and not just the *Interesting* attribute, but other attributes cannot be authored in ELEON and have to be specified by directly modifying the RDF files. Work is underway to generalize the ELEON user interface to provide access to arbitrary attributes.

[6] Currently for English and Greek only.

[7] See `http://www.ltg.ed.ac.uk/methodius/`

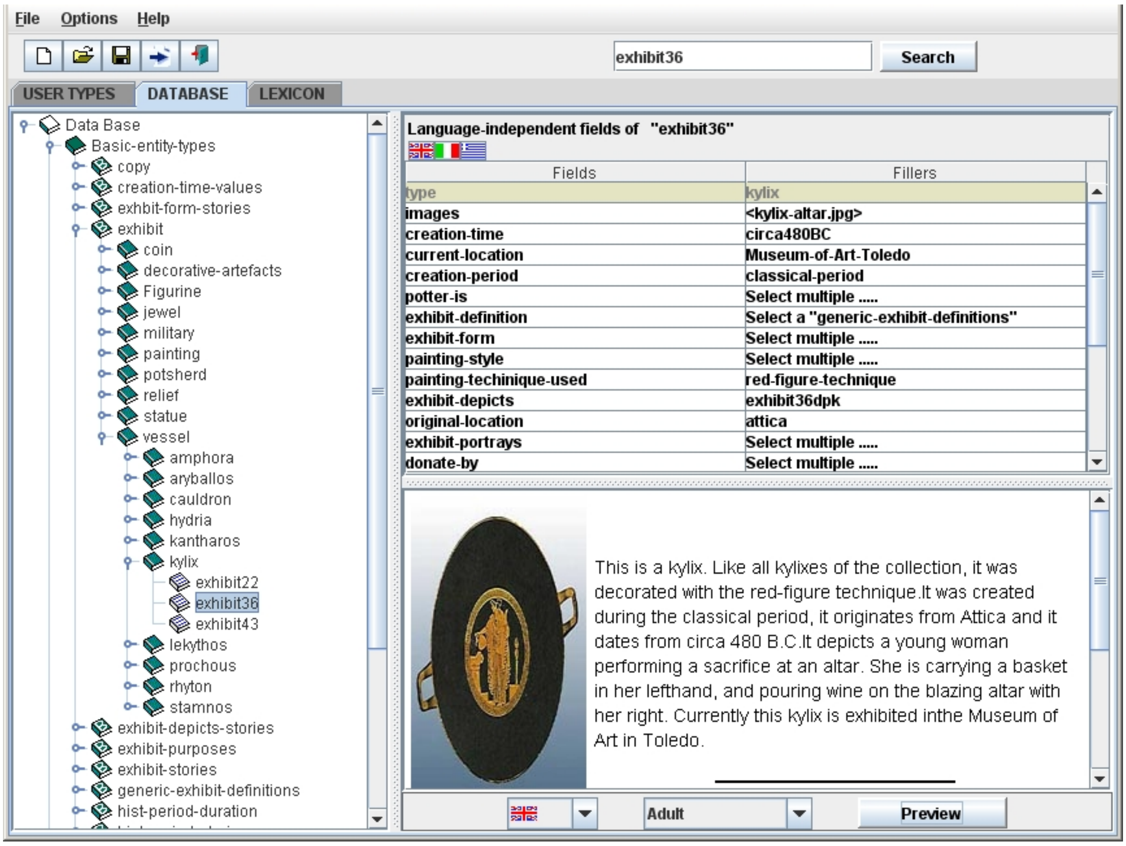

Figure 3. ELEON screenshot, showing a preview of the description of the selected entity.

cludes an interface to the DL reasoning engine RACERPRO.[8]

3 ROBOT PERSONALITY

In INDIGO, robot personality is expressed in a number of ways, influencing dialogue management and natural language generation as well as facial expressions and speech synthesis. Following up on the robot interests example, we will discuss how robot personality calculates the interest levels used by the various INDIGO components.

3.1 Many-valued Description Logics

Inference over OWL models is facilitated by Description Logic reasoners. *Description Logics* (DLs) [2] are a family of first-order logics. Their main characteristic is *decidability*, attained by being restricted to *concepts* (unary predicates, sets of individuals) and *relations* (binary predicates, sets of pairs of individuals).

Many-valued logics in general, and consequently many-valued DLs, extend the 2-valued true-false valuations of logical formulae into many-valued numerical valuations, denoting the *degree* to which formulae hold. Such many-valued models receive their semantics not from set theory, as is the case with 2-valued valuations, but from algebraic *norms* that define the logical connectives.

One of the most widely-used many-valued semantics is based on the Łukasziewicz t-norm and residuum [11], defining conjunction and implication, respectively, as follows:

$$\begin{aligned} \deg(C \sqcap D) &= \max(0, \deg(C) + \deg(D) - 1) \\ \deg(C \sqsubseteq D) &= \min(1, 1 - \deg(C) + \deg(D)) \end{aligned}$$

[8] See `http://www.racer-systems.com/`

where C and D are clauses and $\deg(\cdot)$ is the valuation of a clause. The rest of the logical connectives are then defined so that they verify De Morgan's Laws:

$$\begin{aligned} \deg(C \sqcup D) &= \min(1, \deg(C) + \deg(D)) \\ \deg(\neg C) &= 1 - \deg(C) \\ \deg(C \equiv D) &= 1 - \mathrm{abs}\left(\deg(C) - \deg(D)\right) \end{aligned}$$

The disjunction operator is also called the *t-conorm*.

In INDIGO, many-valued DL reasoning services are provided by YADLR [10]. One of the most attractive, for INDIGO, features of YADLR is that it is a modular system providing multiple alternative reasoning back-ends, one of which being an *instance based* reasoner. Instance-based reasoning directly and efficiently handles queries about instances' belonging to classes, but is less efficient (and in some cases incomplete) when faced with complex class expressions and queries regarding class consistency and subsumption.

3.2 Many-valued DL Representation

We have discussed above how INDIGO involves models in two different formalisms: an OWL representation of the domain, and multiple RDF (user and robot) personality models attributing numerical levels of interest, importance, etc. to individuals, classes, and properties of the domain.

As YADLR supports $\mathcal{SHOIQ}$, a DL that is expressive enough to capture OWL [8], transferring knowledge from the domain ontology to the reasoning service is straightforward.

Personality-related attributes of individuals are captured by normalizing in the $[0, 1]$ range and then using the normalized value as a class membership degree. So, for example, if `interesting` is such an attribute of individual exhibits, then an exhibit with a (normalized) interest level of 0.75 is a member of the Interesting class at a degree of 0.75:

```
(instance stoa_attalus Interesting 0.75)
```

Attributes of classes are given the semantics of a class subsumption at the degree of the attribute. Continuing the previous example, if the class of stoas is interesting at a degree of 0.6, this is taken to mean:

```
(implies Stoa Interesting 0.6)
```

Solving the residuum norm for $\deg(D)$ we see that being a stoa at degree d implies being interesting at degree $d - 0.4$. Or, in other words, Stoa-ness translates to Interesting-ness at a loss of 0.4 of a degree.

It should be stressed that per-instance interests are not exceptions to per-class interests, as Description Logics are monotonic and specific assertions cannot 'override' more general ones. In this manner, in the following knowledge base:

```
(implies Stoa Interesting 0.6)
(instance stoa_attalus Stoa 1)
(instance stoa_attalus Interesting 0.75)
```

the interest factor of 0.75 does *not* override the factor of 0.6 inferred via the stoas-are-interesting axiom. Instead, these two factors (as well as any other alternative paths of proving `stoa_attalus` to be interesting) are disjoined using the t-conorm, yielding a factor greater or equal than the greatest of the conjoined factors, in our example 1.0. This can be thought of as multiple separate reasons why `stoa_attalus` is interesting reinforcing each other.[9]

Finally, personality models also assign numerical attributes to properties, like orderedBy, creationEra or style. These attribute encode the information that a robot guide might, for example, find it more interesting to describe the artistic style of an exhibit rather than provide historical data about it. As a consequence, the interest factor of an exhibit's style will contribute more to the interest of the exhibit, than the interest factor of its creator:

```
(implies (some style Interesting)
         Interesting
         0.8)
(implies (some orderedBy Interesting)
         Interesting
         0.4)
```

where `(some style Interesting)` is the class of things that are related to at least one Interesting instance with the style property. Membership in this class is determined as the maximum interest degree of the *fillers* (value instances) of the exhibit's style property.

To demonstrate how this works, assume the Interesting membership degrees in Table 1. In this example `stoa_attalus` has an interesting style at a degree of 0.8, which is the maximum among the three architectural styles found in the stoa. This contributes 0.6 to the stoa's Interesting-ness itself, whereas its 0.8-interesting orderedBy filler contributes only 0.2.

In the examples so far, the attributes of a given property were identical for each robot for all exhibits. Another capability of the annotation schema used is that attribute values can be restricted to only apply to properties of particular (classes of) exhibits, in which case the property-value restriction is conjoined with membership in the class.

[9] There is no issue about these being independent or not, as Łukasziewicz calculus is *not* probabilistic.

Table 1. Example

Instance	Class	Degree
doric	Interesting	0.8
ionic	Interesting	0.7
pergamene	Interesting	0.3
stoa_attalus	style.Interesting	0.8
stoa_attalus	Interesting	0.6
attalus	Interesting	0.8
stoa_attalus	orderedBy.Interesting	0.8
stoa_attalus	Interesting	0.2

So, continuing our previous example, our robot might find style more interesting for temples than it is for stoas:

```
(implies (and (some style Interesting)
              Temple)
         Interesting
         0.5)
(implies (and (some style Interesting)
              Stoa)
         Interesting
         0.2)
```

3.3 Robot Personality Traits

One of the best established psychological models of *personality traits* is the OCEAN model [5], which defines five elementary parameters: openness, conscientiousness, extroversion, agreeableness, and neuroticism. In INDIGO, we represent personality traits as logical axioms where different personality profiles are modelled as different ways of weighing the attributes of the domain objects. In other words, different robots might share the same interests, but still react differently in similar situations, depending on their personality.

To make this more concrete, the Interesting concept discussed above is, in fact, the RobotInteresting concept, which captures robot interests. Similarly, the UserInteresting concept is defined based on aggregate information over classes of users as well as the current interaction history between the user and the robot (cf. Section 2.3 above). Finally, the AgendaInteresting concept includes at each dialogue state exhibits that are coming up next in a pre-defined tour. The manner of combining the three (as well as, possibly, other attributes) into a single Interesting concept is the expression of a robot's personality.

In INDIGO we are, in fact, mostly interested in defining a personality-influenced Preferred concept, where a high degree of membership promotes an exhibit into the dialogue manager's field of vision when deciding what to describe next. This decision mediates among:

- the tour's pre-established agenda, or scenario, specifying which exhibits should be visited independently of either the robot's or the user's wishes and interests.
- the relevant objective attributes of individual exhibits, such as importance, independently of the agenda of the specific tour being delivered.
- the robot's own interests; and,
- the user's interests.

By pragmatically interpreting the OCEAN parameters, we see that:

- by sticking to the prescribed agenda a robot exhibits a high level of *conscientiousness*.
- by putting user interests above both the agenda and its own interests the robot exhibits *openness* and *agreeableness*.
- by grudgingly abiding to user wishes, the robot exhibits *agreeableness* and *introversion*.

This interpretation can be encoded as a series of logical definitions of robot preferences, as membership of exhibit in the Preferred class.

An introvert robot will have distinct definitions for Preferred and DescribedAtLength; this encodes the intuition that even if it suppresses its own preference when planning the tour, this will surface as a reluctance to give long, enthusiastic descriptions:

```
(implies (and UserInteresting
              AgendaInteresting)
         Preferred)
(implies RoboInteresting
         DescribedAtLength)
```

Naturally, an introvert might also be self-centred and unconscientious, in which case the two concepts will have identical definitions, or it might happen that the robot and user interests coincide with the next tour agenda item, but the point is that Preferred and DescribedAtLength do not necessary coincide.

An extrovert, on the other hand, will externalize preferences while planning, so that Preferred and DescribedAtLength coincide. A conscientious robot that is not open to suggestions will stick to the plan and be happy to elaborate on the relevant exhibits:

```
(implies AgendaInteresting Preferred)
(implies Preferred DescribedAtLength)
```

and an open and agreeable personality will put user interests first:

```
(implies UserInteresting Preferred)
(implies Preferred DescribedAtLength)
```

as opposed to a personality that is closed to suggestions with low agreeableness:

```
(implies RoboInteresting Preferred)
(implies Preferred DescribedAtLength)
```

A conscientious, but closed, personality will balance between its agenda obligations and its own interests, and be closed to user requests:

```
(implies (and RoboInteresting AgendaInteresting)
         Preferred)
```

whereas a conscientious and open personality will balance between its agenda obligation and the user's interests:

```
(implies (and UserInteresting AgendaInteresting)
         Preferred)
```

At the current state of experimentation, as these examples show, the OCEAN parameters are taken as binary values creating a space of 16 possible personalities.[10] An interesting open question in our approach is to see how best to handle numerical OCEAN parameters. One possibility is to use these as relevance factors for the implications. For example, an open and agreeable personality will put user interests first:

[10] It is not clear yet how to represent neuroticism, so only stable personalities with low neuroticism are represented.

$$\left[\text{IS:}\ \left[\begin{array}{l}\text{SHARED:}\ \left[\begin{array}{l}\text{PERSONALITY: robot3}\\ \text{VISITED: }\{\text{o5, o32, o21}\}\\ \text{POSITION: o5}\end{array}\right]\\ \text{DH: [direct(sys,move,o3),ask(usr,see,stoa), \ldots]}\\ \text{LU: direct(sys,move,o3)}\end{array}\right]\right]$$

Figure 4. Part of the IS Structure

```
rule( assertNoMore,
      [
        $/shared/lu/speaker == usr,
        in( $/shared/lu/moves,  utt(no) ),
        fst(  $/shared/dh, act(sys, ask, more ))
      ],
      [
        push( /shared/dh, act(usr, assert, no) ),
        push( /shared/dh, act(sys,goodbye) )

      ] ).
```

Figure 5. Example Update Rule

```
(implies UserInteresting Preferred v)
```

where v is the robot's openness parameter. How to combine parameters in axioms like the one for 'conscientious but closed' will be the subject of future experimentation.

4 PERSONALITY EXTERNALIZATION

In this section we discuss how the parameters inferred by robot personality are used by various components in the INDIGO system. This is the manner in which robot personality is externalized, that is, how it is manifested in its interaction with the user.

4.1 Dialogue Management

The dialogue and action manager (DAM) implements the standard operations of deciding on an interpretation of the user's input and determining what to do and say next. Input in the form of speech and/or gesture is fused into a single representation which is passed into the DAM to be processed. Output in the form of robot speech, robot platform movements, facial expressions and head gestures is then determined and passed on to the appropriate modules.

The DAM is implemented using the TrindiKit, which assumes the *Information-State (IS) and Update* approach to dialogue management [14]. The IS can be used to store any kind of information which is relevant to the dialogue genre, including in the present case static information about the personality models as well as dynamic information such as the dialogue history and the current robot position. The dialogue history is usually represented as a stack, allowing the contents of previous utterances to be stored for later checking, as illustrated below.

An IS can be represented using an attribute-value matrix, as shown in truncated form in Figure 4. When an utterance and/or gesture analysis is passed into the DAM it is processed by a series of typed update rules which specify what the effects on the IS should be. The sequence of rule applications is determined by an update algorithm, which in the case of INDIGO is a simple sequence of rule types. Each individual rule contains a list of conditions and a list of effects to apply if the conditions are satisfied. Figure 5 shows part of a typical rule which handles a 'no' response from the user to a particular question. The conditions state that the speaker was the user, who uttered some form of 'no', and the last thing in the dialogue history was the system asking a 'more' question, which is shorthand for 'would you like to see more buildings'. If the answer is 'no', the dialogue is finished and the system says 'thanks and goodbye', so the rule's effects just note this.

In the museum guide use-case, robot-visitor dialogues have the following general structure:

1. The robot welcomes the visitor, and offers a variety of possible tours from which to choose. The user makes a choice and the tour begins.
2. The robot describes an object, and queries whether the user wants an elaboration of the description, an explanation regarding one of the terms used in the description, or to move on.
3. If prompted to move on, the robot will choose the next item to describe. If the user requests a specific object, the robot might or might not comply with the user's request, depending on robot personality and affinity of the requested object to the tour's goals.
4. The tour agenda might specify junctions where the robot will offer the user a number of sub-tour options. Depending on robot personality, the robot might prefer the user's taking one of the alternatives and actively show so.
5. The tour ends when the agenda runs out of objects to describe or the user indicates that they do not want to proceed.

Robot personality influences points 3 and 4 above, by providing numerical *preference factors*, as explained above. More specifically, when selecting particular objects to describe, the DAM requests the Preferred membership degree for all the options offered to the user and, if any, for the object requested by the user. When selecting which class of objects to turn to next,[11] the DAM requests the Preferred membership degree of the generic member of the class, i.e., of the object that has no other property than those associated with membership in the requested class.

4.2 Natural Language Generation

NATURALOWL [7] is a natural language generation engine mostly geared towards generating natural language descriptions of ontological entities, based on these entities' abstract properties. It traces its origins to technology developed in the M-PIRO project [9] for personalized natural language descriptions of abstract data.

NATURALOWL generates descriptions that are dynamically customized to the current audience. Adaptation is achieved though the *user modelling* mechanism described in Section 2, parameterizing the generation to different user profiles. Furthermore, NATURALOWL uses an, externally provided, *assimilation score* reflecting the current interaction history. This allows NATURALOWL to avoid gratuitous repetition of material and to draw comparisons to previously seen objects.

Comparisons are built around the hierarchical organization of entities into classes: entities of the same class are compared based on the common properties they have. As a result, the generated text can contain information such as:

> It is the *only* building in the Agora which was build in the Pergamene style.

NATURALOWL uses profiles and assimilation to decide which properties of an entity are:

1. interesting or appropriate to the user; and
2. not already used to describe this or other objects.

and, based on the above, selects which properties are to be included in a description. Both interest/appropriateness and assimilation are provided to NATURALOWL by external tools; the former can be either a static user profile or a dynamic user model and the latter can also be from a simple counter to a complex user model.

This mechanism is exploited in INDIGO to vary descriptions of the same object to the same user depending on which robot provides the description. More specifically, instead of basing these parameters on the user alone they are inferred by robot personality by balancing between the user and robot interests (cf. Section 3).

4.3 Emotional Appraisal and Mood

Dialogue actions—user as well as robot actions—have an impact on the *mood* and *emotional state* of the robot, which, in turn, are used to drive speech synthesis and facial expressions. This impact is represented using the OCC model, which defines elementary emotions, such as *joy*, *gratitude*, *pride*, *distress*, *disappointment*, and so on. Ortony [12] provides a full list and describes how dialogue actions map to different points in the OCC space.

In INDIGO, OCC vectors for user actions stem from language analysis to reflect the impact of the *manner* of what the user said. That is, polite or impolite linguistic markers trigger the appropriate OCC appraisal. More pertinent to the work described here is appraisal stemming from robot personality, reflecting robot actions and, indirectly, user requests, that is, the *content* of what the user said; after personality modelling and the DAM have reached a decision on how to react to a user action, this planned robot action is also used to create an OCC appraisal.

Appraisal vectors are used by an emotional-state machine, which updates the robot's emotional state to reflect the latest emotional appraisal received, taking into account the OCEAN personality traits and the previous emotional state. The current emotional state is used by the speech synthesiser and the animated face or animatronic head, which reflect emotional state as voice modulations and facial expressions.

5 CONCLUSIONS

In the introduction we have set our overall goal to be natural interaction with the user, and a specific sub-goal of robot personality modelling that it is manifested as different behaviour by the same robot in different situations as well as different behaviour by different robots in the same situation.

To this end, we approach emotion and personality modelling as a reasoning process where each individual personality is represented as a different many-valued logic program and emotion parameters are an instantiation of the program. From these, numerical parameters are inferred which influence the behaviour of a multi-modal dialogue system.

We have presented here an early state of development of the personality models, where a very small part of the OCEAN personality-modelling space is explored. Our future research will focus on incorporating all five OCEAN parameters and on using them as numerical, instead of binary, parameters.

One of the most attractive features of the approach is the explicit, logical representation of personality aspects, as opposed to creating a variety of dialogue-management update rules to reflect the different

[11] This class can be either a thematic concept of the domain, for example Stoa, or one created for the purposes of a particular exhibition and tour, for example ExhibitsInRoom16.

personalities. This provides a sound infrastructure for experimentation and future research, clarifying the various system components that interact in a dialogue system.

One of the drawbacks of the proposed approach is the volume and detail of robot profile information that needs to be manually authored. To alleviate this problem, work is underway to extend the reasoning-based author support in ELEON to infer as many of the values of emotion modelling attributes and present the inferred values to the author. In this manner, the author can start out with specifying values for wide, upper classes and then only refine the model where necessary.

ACKNOWLEDGEMENTS

The work described here is supported by the FP6-IST project INDIGO. INDIGO develops and advances human-robot interaction technology, enabling robots to perceive natural human behaviour, as well as making them act in ways that are more familiar to humans. To achieve its goals, INDIGO advances various technologies, which it integrates in a robot platform.

For more information on the INDIGO project, please visit the project web site at `http://www.ics.forth.gr/indigo/`

REFERENCES

[1] Ion Androutsopoulos, S. Kallonis, and Vangelis Karkaletsis, 'Exploiting OWL ontologies in the multilingual generation of object descriptions', in *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05), Aberdeen, August 2005*, (2005).

[2] *The Description Logic Handbook: Theory, Implementation and Applications*, eds., Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, Cambridge University Press, 2003.

[3] Dimitris Bilidas, Maria Theologou, and Vangelis Karkaletsis, 'Enriching OWL ontologies with linguistic and user-related annotations: the ELEON system', in *Proc. 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2007), Patras, Greece, October 29–31, 2007*, volume 2. IEEE Computer Society, (2007).

[4] Dan Brickley and R. V. Guha, 'RDF primer', W3C Recommendation (February 2004). `http://www.w3.org/TR/2004/REC-rdf-primer-20040210/`.

[5] P. T. Costa and R. R. McCrae, 'Normal personality assessment in clinical practice: The NEO personality inventory', *Psychological Assessment*, **4**(5–13), (1992).

[6] Arjan Egges, S. Kshirsagar, and N. Magnenat-Thalmann, 'Generic personality and emotion simulation for conversational agents', *Computer Animation and Virtual Worlds*, **15**(1), 1–13, (January 2004).

[7] Dimitris Galanis and Ion Androutsopoulos, 'Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system', in *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG 2007), Schloss Dagstuhl, Germany*, pp. 143–146, (2007).

[8] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen, 'From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language', *Journal of Web Semantics*, **1**(1), 7–26, (2003).

[9] Amy Isard, Jon Oberlander, Ion Androutsopoulos, and Colin Matheson, 'Speaking the users' languages', *IEEE Intelligent Systems*, **18**(1), 40–45, (2003).

[10] Stasinos Konstantopoulos and Georgios Apostolikas, 'Fuzzy-DL reasoning over unknown fuzzy degrees', in *Proc. 3rd Intl. IFIP Workshop of Semantic Web and Web Semantics* (SWWS 07), 29–30 Nov. 2007, Vilamoura, Portugal. Volume 4806 of *Lecture Notes in Computer Science*, pp. 1312–1318, Berlin/Heidelberg, (2007). Springer.

[11] Jan Łukasiewicz and Alfred Tarski, 'Untersuchungen uber den Aussagenkalkül', *Comptes Rendus des Séances de la Societé des Sciences et des Lettres de Varsovie, Classe III*, **23**, (1930). Reprinted in translation in L. Borowski (ed.), *Jan Łukasiewicz, Selected Writings*, North-Holland, 1970.

[12] Andrew Ortony, 'On making believable emotional agents believable', *Emotions in Humans and Artifacts*, chapter 6, pages 189–212. MIT Press, April 2003.

[13] Michael K. Smith, Chris Welty, and Deborah L. McGuinness, 'OWL web ontology language', W3C Recommendation (February 2004). `http://www.w3.org/TR/owl-guide/`.

[14] David Traum and Steffan Larsson, 'The information state approach to dialogue management', in *Current and New Directions in Discourse and Dialogue*, eds., Jan van Kuppevelt and Ronnie Smith, Kluwer Academic Publishers, Dordrecht, the Netherlands, (2003).