## FML Templates

For the communicative behaviour generation the DM has an FML manager component created in the Multi-Manager Flipper Dialogue System (more details are provided in D3.2). This FML manager (1) selects one of the available FML-templates according to the communicative intents to accomplish, (2) fills the available placeholders (each template supports a set of parameters described below and summarized in Table 2: The parameters of an FML Template.) with information extracted from the Information State in Flipper (e.g. a subject in an utterance or an emotion) and (3) sends the template filled in with information to an FML Translator component that in turn produces a regular FML-APML script for Greta.

The **FML Templates** are based and categorized according to the **Dynamic Interpretation Theory** (DIT++) taxonomy of communicative functions[1]. DIT++ describes the communicative functions of dialogue segments, helpful for identifying both verbal and non-verbal intentions. Every dialogue segment can have multiple functions. For example, a response by the agent to a question may indicate that the agent understood there was a question (positive feedback) and replied to the question (e.g., inform as a response to a set question). For each of the relevant communicative functions contained within DIT++, we created an FML template with a subset of parameters as shown in Table 2. An overview of all the DIT++ categories of communicative functions that we used for creating our FML Templates is listed in Table 1.

|  |  |  |
| --- | --- | --- |
| **Class** | **Explanation** | **Behaviour** |
| **Information Transfer** | The class of information-transfer functions consists of all those functions whose primary aim is to obtain or to provide information. The class is further divided into information-seeking and information-providing functions. | We can adjust the amount of non-verbal behaviour for answering and posing questions, such as hand gestures for explanations or head tilts for questions. |
| **Feedback** | Feedback acts provide or elicit information about the processing of the previous utterance(s). | Behaviour for positive (head nod)/negative (frown) feedback during the conversation can be generated for back-channel. |
| **Interaction** | Acts involved without specific semantic content, but structure the dialogue, such as turn-taking, searching for contact, topic management. | Behaviour for example wanting the turn (holding up hand) or seeking contact (gaze). |
| **Social Obligations** | Acts involved with social policies during dialogues. | We have FML templates for saying hello (waving), saying goodbye, thanking, apologizing and expressing gratitude. |

Table 1: The DIT++ categories used for creating our FML Templates.

#### **Templates parameters**

A given template corresponds to a communicative function in the DIT++ category that the agent aims to accomplish. The parameters that the DM can pass in input are defined in Table 2. The *Template TAG* column indicates the element name that appears in the FML template. The *Supp.* column indicates whether the parameter was originally supported by FML-APML already (FML) or if it has been added as a new placeholder as specialization (ARIA). The *Attribute* column indicates an element’s attribute that is changeable (i.e. by the DM). The *Value(s)* column indicate possible values for that attribute (in bold the default one).

|  |  |  |  |
| --- | --- | --- | --- |
| **Template TAG** | **Supp.** | **Attribute** | **Value(s)** |
| **emotion** | FML | type | ***neutral*** | panic | fear | anxiety |sadness |envy | anger | joy | despair | disgust | hate | worried | disappointment | pride | tension | relief | embarrassment | surprise | bored |
| **emotion** | FML | intensity | [0..**1**] float |
| **emotion** | FML | importance | [0..**1**] float |
| **pitchaccent** | FML | level | none | reduced | moderate | strong |
| **pitchaccent** | FML | importance | [0..**1**] float |
| **aria-var** | ARIA | type | name | emotion-verb | emotion-adjective | noun | cause-noun | pronoun | location | topic | user | agent |
| **certainty** | FML | type | **none** | uncertain | certain |
| **certainty** | FML | intensity | [0..**0.5**..1] float |
| **certainty** | FML | importance | [0..**1**] float |
| **voice** | FML | emotion | calm | cross | happy | sad | **none** |
| **aria-alternative** | ARIA | type | static, dynamic, selectable |
| **aria-alternative** | ARIA | name | string (used to link multiple aria-alternative tags by name, applicable only when type is selectable) |
| **alternative** | ARIA | option | <alternative> is a child of <aria-alternative> and the option attribute is valid only for children of selectable aria alternatives |

Table 2: The parameters of an FML Template.

The **emotion** tag has a type, intensity and importance attributes. This element sets a particular emotion that has to be expressed by the agent together with other communicative intents or alone. The importance sets the relative priority compared to other elements in the FML templates and it is used by Greta when transforming the generated FML into BML. The **pitchaccent** emphasizes verbally and nonverbally a certain part of the speech (i.e. dialogue act) in the utterance present in a template. The **certainty** tag allows the DM to specify if an intent should be expressed (via nonverbal behaviours) with certitude or incertitude.

The **voice** tag is supported by FML-APML and it has been originally defined within the Cereproc TTS Engine to synthetize speech with different emotional stances. We included it in the templates as well and its attribute named *emotion* can have 4 possible values as indicated above in the table.

An **aria-var** (i.e. *<aria-var type=”?” />*) represents a placeholder for a single or a couple of words. The whole aria-var element is replaced by the given input that is provided by the DM according to the type attribute. More specifically, *name* is the name of a character, user is the user’s name and agent is the agent’s name.

An **aria-alternative** (i.e. *<aria-alternative type = ”static|dynamic|selectable”>*) is placeholder that supports alternative texts or FML annotations that can be randomly or manually selected according to the type attribute:

· **Static** **alternative**: it contains a list of *<alternative>* child tags that embrace FML ready content. When present, one of the alternative child elements will be randomly selected with a uniform distribution.

Example of static alternative tag:

*<aria-alternative id="alt1" type="static">*

*<alternative>Let me <tm id="tm1"/>explain:</alternative>*

*<alternative>For <tm id="tm1"/>example:</alternative>*

*</aria-alternative>*

· **Dynamic** **alternative**: this placeholder behaves similarly to a static alternative but it allows the DM to pass a list of items (i.e. words) separated by a delimiter that constitute alternatives and one of the alternatives will be randomly selected based on a uniform distribution.

Example of dynamic alternative tag: *<aria-alternative type=”dynamic” />* (N.B.: an input for this could be something like “yes; so true; exactly”)

· **Selectable** **alternative**: this placeholder takes as input from the DM a specific alternative that can then be selected through its option attribute. Moreover, multiple selectable alternatives within an FML Template can be linked together if they have the same name attribute. The result of linking selectable alternatives is that the selected alternative option will be chosen among all linked aria-alternative elements.

Example of two linked selectable alternative tags:

*<aria-alternative id="alt1" name="feedback" type="selectable">*

*<alternative option="positive">Yes sure</alternative>*

*<alternative type="negative">No way</alternative>*

*</aria-alternative>*

*<aria-alternative id="alt2" name="feedback" type="selectable">*

*<alternative option="positive"><some FML-APML tags></alternative>*

*<alternative type="negative"><other FML-APML tags></alternative>*

*</aria-alternative>*

ARIA specific tags can be nested, this means that, for example, it is possible to create nested structures by including within a selectable aria-alternative: (a) aria-var elements or (b) a static aria-alternative or (c) a dynamic aria-alternative (d) a static aria-alternative (nested in a selectable one) might still contain one or more aria-var elements.

#### **FML Translator**

The FML Translator (see Figure 2) has the task of transforming a given FML Template (chosen by the DM) according to the input parameters (e.g. emotions, aria variables) into pure FML-APML for Greta.

These are the steps followed by the FML Translator algorithm for accomplishing this transformation task:

1. Find **selectable** **aria-alternative** elements and replace them with selected option’s alternative content (the option is parameter that the DM gives in input).
2. Find **dynamic** **aria-alternative** elements, randomly choose an alternative among the available ones and replace its content with given input list of items.
3. Find **static aria-alternative** elements, randomly choose an alternative and strip out its contents in the final FML-APML.
4. Find **aria-var** elements and replace those according to the given DM input
5. Find **voice tags** (for Cereproc) and replace emotion attribute OR remove voice brackets if input is not given
6. Find and replace values for **FML** **attributes** (e.g. emotion type and intensity, certainty type and intensity).

[1] Bunt, Harry. "The DIT++ taxonomy for functional dialogue markup." AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts. 2009.