# UI on the Fly: Generating a Multimodal User Interface

**David Reitter**
Media Lab Europe
Dublin, Ireland

**Erin Marie Panttaja**
Media Lab Europe
Dublin, Ireland

**Fred Cummins**
University College Dublin
Dublin, Ireland

{reitter,erin}@mle.media.mit.edu

fred.cummins@ucd.ie

## Abstract

UI on the Fly is a system that dynamically presents coordinated multimodal content through natural language and a small-screen graphical user interface. It adapts to the user's preferences and situation. Multimodal Functional Unification Grammar (MUG) is a unification-based formalism that uses rules to generate content that is coordinated across several communication modes. Faithful variants are scored with a heuristic function.

## 1 Introduction

Multimodal user interfaces are everywhere. The use of a keyboard and mouse on a desktop PC is ubiquitous, if not natural. However, the click-then-type paradigm of common interfaces misses the cross-modal synchronization of timing and meaning that is evident in human-human communication. With coordinated output, novice users could get explanations (redundant content) and experienced users could receive additional (complementary) information, increasing the bandwidth of the interface. Coordinated input ("put that there!") speeds up input and relieves speech recognition of notoriously hard-to-recognize referring expressions such as names. If a user interface is generated on the fly, it can adapt to the situation and special needs of the user as well as to the device.

While users are not necessarily prone to make multimodal inputs (Oviatt, 1999), they can still integrate complementary output or use redundant output in noisy situations. Consequently, this paper deals with generating output. We propose a grammar formalism that generalizes decisions about how to deliver content in an adaptable multimodal user interface. We demonstrate it in the context of a user interface for a mobile personal information manager.

## 2 Related Work

Since Bolt's (1980) Put-That-There system introduced cross-modal coordination in multimodal user input, various projects have investigated multimodal input and output methods. Users display a preference for the touch-screen in map-based positioning acts and object selection (Oviatt et al., 1997). WIP (André et al., 1993) and other systems (Feiner and McKeown, 1990; Roth and Hefley, 1993) generate static multimodal documents. In an interactive user interface, however, layout should remain consistent (Woods and Roth, 1988, perceived stability).

SmartKom (Wahlster, 2002) is a recent effort that produces a multimodal user interface, using XML/XSLT techniques to render the output. These are deterministic, which makes soft constraints such as usability hard to implement. SUPPLE (Gajos and Weld, 2004) overcomes this problem in its model of the user and the expected workload for various interfaces, generating a unimodal (graphical) user interface without natural language generation elements. On the integration side, Johnston (1998) presents a unification-based grammar that recasts multimodal signal fusion as a parsing problem.

Our approach employs a non-deterministic grammar to derive variants which are evaluated with a comparatively simple user and situation model according to their utility (information conveyed) and the projected cognitive load imposed on the user. It also removes the requirement inherent in Johnston's system of explicitly defining rules to integrate multimodal information.

In the following, we discuss the grammar formalism used to create output, as well as consistency and adaptation considerations.

## 3 Formalism

In this section, we will explain how the Multimodal Functional Unification Grammar (MUG) allows us to generate content. Our formalism and the associated evaluation algorithm work closely with a dialogue manager. As input, they receive an unambiguous, language- and mode-independent representation of the next dialogue turn.

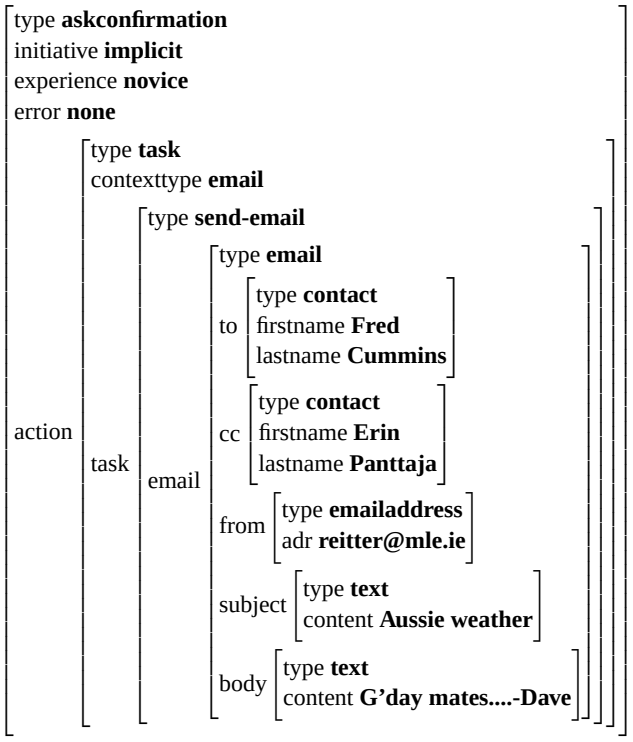



Figure 1: Input representation: confirmation of sending of an email

### 3.1 Dialogue acts as input

Although the semantic input is independent of mode (screen, voice) and language (Portuguese), the input semantics are domain-specific. The representation uses the following types of dialogue acts at the top level: ask for missing information, ask for a confirmation of an action or data, inform the user about the state of objects, or give context-dependent help.

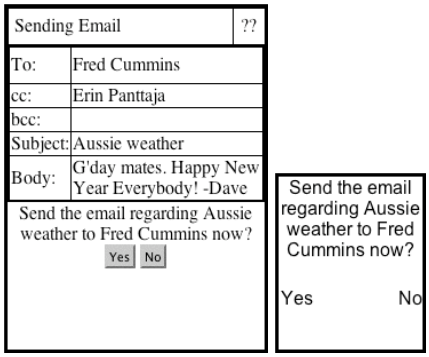
An example is shown in Figure 1. The input-FD specifies type of act in progress (askconfirmation), and the details of the interaction type. It then specifies the details of the current action, in this case, the email that the user is sending.

Furthermore, the dialogue manager may indicate the need to realize a certain portion of an utterance with an attribute *realize*. The input format integrates with principled, object-oriented dialogue managers.

### 3.2 The domain: a personal assistant.

In this example, we have constructed a personal assistant to be used in the domain of sending email messages.

We implemented a MUG for a PDA-size handheld device with a color touch-screen (see Figure 2a). The initial steps to adapt it to a mobile phone (Figure 2b) involved creating a device profile that uses no GUI widgets and associates a higher cost (see Section 5) with the screen




(a) (b)

Figure 2: a) Voice: "Do you want to send the email? Yes or No?". b) Voice: "Send the email regarding Aussie weather to Fred Cummins now?"

output, as the screen is smaller. All devices used have server-driven TTS output capabilities.

### 3.3 The grammar

MUG is a collection of *components*. Each of them specifies a realization variant for a given partial semantic or syntactic representation. This representation may be specific to a mode or general. We call these components *functional descriptions* (FDs) in the tradition of the Functional Unification Grammar (Kay, 1979), from which MUG is derived.

For each output, the MUG identifies an *utterance plan*, consisting of separate constituents in the output. For example, when we ask for missing information ("Who would you like to send the e-mail to?"), the utterance consists of an instruction and an interaction section. Such a plan is defined in a component, as is each more specific generation level down to the choice of GUI widgets or lexicon entries.

MUG is based on the unification of such attribute-value structures. Unification can be seen as a process that augments an FD with additional information. FDs are recursive: a value can be atomic or a nested FD. Values in an FD can be bound to the values in a substructure FD (structure sharing).

To realize a semantic representation $R$, we unify a suitable grammar component FD with each $m$-constituent substructure $F$ in $R$, until all substructures have been expanded. An $m$-constituent is an FD that has an attribute path $m|$cat, that is, which has been designated as a constituent for mode $m$. Note that zero or one grammar components for a given mode can be unified with $F$.

Components from the grammar invoke each other by instantiating the *cat* attribute in the mode-specific part of

a substructure. Figure 3 shows a component that applies to all modes.

There may be several competing components in the grammar. This creates the ambiguity needed to generate a variety of outputs from the same input. Each output will be faithful to the original input. However, only one variant will be optimally adapted to the given situation, user, and device (see Section 5). Our final markup is text for the text to speech system as well as HTML to be displayed in a browser, similar to the MATCH system (Johnston et al., 2002).

The nested attribute-value structures and unification are powerful principles that allow us to cover a broad range of planning tasks, including syntactic and lexical choices. The declarative nature of the grammar allows us to easily add new ways to express a given semantic entity. The information that each component has access to is explicitly encapsulated by an FD.

A grammar workbench allows us to debug the generation grammar. We could improve the debugging process with a type-hierarchy, which defines allowed attributes for each type.

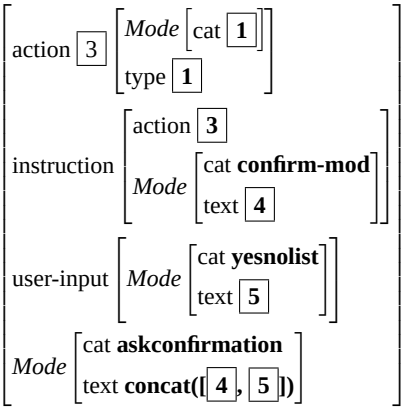



Figure 3: A MUG component that handles the confirmation of tasks or user input. The mode in variable *Mode* may be *voice* or *screen*.

## 4 Planning for Coherence

Coherence is a key element in designing a multimodal user interface, where the potential for confusion is increased. Our user interface attempts to be both consistent and coherent. For example, lexical choice does not vary: it is either 'mobile phone' or 'cell phone,' but it is the same whether it is in text or voice. This is in line with priming effects, which are known to occur in human-human dialogue.

Like humans (McNeill, 1992; Oviatt et al., 1997), our system aims to be coherent and consistent across all modes. We present redundant content, for example, by choosing the same lexical realizations (never mix *cell phone* and *mobile phone*). We present complementary input in linked components. If, for example, a deictic expression such as *these two e-mails* (by voice) requires the e-mails to be put in focus on the screen, it will set a feature accordingly in the complementary mode.

This is possible because of a very simple principle encoded in the generation algorithm: all components realizing one semantic entity must unify. Components may still specify mode-specific information. This is done in a feature named after the mode, so it will not interfere with the realization instructions of a component that realizes the same semantic entity in another mode. The FDs allow us to distinguish information a) that needs to be shared across all output modes, b) that is specific to a particular output mode, or c) that requires collaboration between two modes, such as deictic pronouns. The unification principle replaces explicit integration rules for each coordination scheme, such as the ones used by Johnston (1998), which accounts for the integration of user input.

## 5 Adaptively Choosing the Best Variant

The application of the MUG generates several output variants. They may include or exclude pieces of information, which may be of more or less utility to the user. (When information is being confirmed, it should be fully described, but in later interactions, the email could be referred to as 'it.')

For example, several components applied to the sub-FD for *task* in Figure 1 may depend more on the screen (Figure 2a) or be redundant in screen and voice output (Figure 2b). This allows the system to reflect a low benefit for output on the screen if the user is driving a car or to increase the cost of voice output if the user is in a meeting, or reflect the fact that one doesn't hear the voice output on a mobile phone while reading the screen.

The system adapts to the user's abilities, her preferences, and the situation she is in by choosing an appropriate variant. These properties are scalar, and the resulting constraints are to be weighted against each other in our objective function. Each piece of output is scored according to a simple trade-off: a) realize content where requested, b) maximize *utility* to the user, and c) minimize *cognitive load* in perceiving and analyzing the output.

These constraints are formalized in a score that is assigned to each variant $\omega$, given a set of available Modes $M$, a situation model $< \alpha, \beta >$, a device model $\phi$ and a utility/time trade-off coefficient $\lambda$:

$$s(\omega) = \lambda \sum_{<e,d> \in E(\omega)} u(e,d) + max_{m \in M}(\beta_m t_m(\omega))$$

$$u(e,d) = P(d, \sum_{m \in M} (\phi_m \alpha_m e_{m|realized}), e_{realize})$$

The first part of the sum in $s$ describes the utility benefit. The function $E$ returns a set of semantic entities in $e$ (substructures) and their embedding depths in $d$. The function $P$ penalizes the non-realization of requested (attribute *realize*) semantic entities, while rewarding the (possibly redundant) realization of an entity. The reward decreases with the embedding depth $d$ of the semantic entity. (Deeper entities give less relevant details by default.)

The cognitive load (second part of the sum) is represented by a prediction of the time $t_m(\omega)$ it would take to interpret the output. This is the utterance output time for text spoken by the text-to-speech system, or an estimated reading time for text on the screen.

Further work will allow us to cover the range of novice to experienced users by relying on natural language phrases versus graphical user interface widgets.

## 6 Conclusion

We have demonstrated a formalism that generates coherent multimodal user interfaces, as well its application in a small-screen email client. As the generation algorithm makes use of both hard constraints and scalar scores, it caters for adaptability. We have proven its functionality and efficiency in a series of examples in the context of a dialogue system, where content is generated in real-time for various usage situations and different devices.

Further evaluation will show whether the fitness function can accurately mirror user satisfaction with a given output variant and whether our form of adaptivity is actually an advantage to users on the go. Without a gold standard for a generation system for dynamic multimodal user interfaces to qualitatively compare against, controlled user trials will allow us to evaluate the usability of the interfaces we have created. Task completion times, user frustration levels, and user satisfaction can then be used to evaluate the success of this model of multimodal interactions.

The underlying formalism is intended to be used in creating, using the MUG Workbench, any multimodal system that can be constructed compositionally, using natural language and other auditory and visual components. As possible examples for future applications, we see a multimodal interface that allows mobile users or users with sensory impairments to traverse information-rich social networks, and a kiosk for multimodal, multilingual access to public transportation options.

## 7 Acknowledgement

The authors would like to thank Stefan Agamanolis, Robert Dale, John Kelleher, Kerry Robinson, and the anonymous reviewers. This research was partially funded by the European Commission under the FASiL project, contract number: IST-2001-38685.

## References

E. André, W. Finkler, W. Graf, T. Rist, A. Schauder, and W. Wahlster. 1993. Wip: The automatic synthesis of multimodal presentations. In M. T. Maybury, editor, *Intelligent Multimedia Interfaces*. AAAI Press, Menlo Park, CA.

Richard A. Bolt. 1980. Put-that-there:voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262 – 270, Seattle.

Steven Feiner and Kathleen McKeown. 1990. Coordinating text and graphics in explanation generation. In *Proc. of AAAI-90*, pages 442–449, Boston, MA.

Krzysztof Gajos and Daniel S. Weld. 2004. Supple: Automatically generating user interfaces. In *Proceedings of IUI-2004*, Funchal, Portugal.

M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. 2002. Match: An architecture for multimodal dialogue systems. In *Proceedings of ACL-2002*.

Michael Johnston. 1998. Unification-based multimodal parsing. In *Proceedings of COLING-ACL 1998*, pages 624–630.

Martin Kay. 1979. Functional grammar. In *Proceedings of the Fifth Meeting of the Berkeley Linguistics Society*, pages 142–158, Berkeley, CA.

David McNeill. 1992. *Hand and mind: What gestures reveal about thought*. University of Chicago Press.

Sharon Oviatt, Antonella DeAngeli, and Karen Kuhn. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422. ACM Press.

Sharon Oviatt. 1999. Ten myths of multimodal interaction. *Communications of the ACM*, 42(11):74–81.

Steven F. Roth and William E. Hefley. 1993. Intelligent multimedia presentation systems: Research and principles. In M. T. Maybury, editor, *Intelligent Multimedia Interfaces*. AAAI Press, Menlo Park, CA.

Wolfgang Wahlster. 2002. Smartkom: Fusion and fission of speech, gestures, and facial expressions. In *Proceedings of the 1st International Workshop on Man-Machine Symbiotic Systems*, Kyoto, Japan.

David Woods and Emilie Roth. 1988. Cognitive systems engineering. In M. Helander, editor, *Handbook of Human-Computer Interaction*, pages 1–43. Elsevier, North Holland.