The Evaluation of Adaptable Multimodal System Outputs

Erin Marie Panttaja
Media Lab Europe
Dublin, Ireland

David Reitter
Media Lab Europe
Dublin, Ireland

Fred Cummins
University College Dublin
Dublin, Ireland

{erin,reitter}@mle.media.mit.edu
fred.cummins@ucd.ie

Abstract

Adaptable multimodal systems are difficult to test. We present a methodology for evaluating parallel multimodal output which is generated in response to a specific set of user, device and situation constraints. Our method involves the ranking of many potential output variants using a fitness function, and selecting well-differentiated variants for user testing. We focus on the generation of multiple variants of user interfaces for small-screen graphical devices with natural language voice output, within a system we term UI on the Fly.

1 Introduction

Coordinated multimodality, adaptivity and automatically-generated interfaces are relatively new paradigms in human computer interface design. Rather than sequentially employing modes to convey information to the user, several modes are used redundantly or complementarily. Sound-enabled interfaces are a simple example for coordinated multimodality, voice-enabled SALT[1] documents another. Many research prototypes represent dynamically generated user interfaces, which can be adapted to the user's special needs in a given situation, for example, if the user cannot pay much attention to the screen while performing maintenance operations or driving a car. Natural language plays a central role in such interfaces, not only in voice output, but also in visual user interfaces adapted to devices like mobile phones that have only limited input options.

These advanced systems are notoriously difficult to test, as they change their behavior dynamically and unpredictably. As systems begin to follow new interface paradigms, evaluation metrics will have to change to take into consideration additional learning time on the part of the user. In addition, test systems are often limited in their functionality, and may depend on full implementation of a dialogue system that may not be available during testing. We circumvent some of these problems by focusing on a system with an *adaptable situation model* which remains fixed during each test case.

There are many different measurements for describing a 'good' system. Does it function within the specification of its design document? Can it be used by its target group? Is it accessible to the hard of hearing? To the blind? To those with motor impairments? Even accessibility is hard to define. A system may be technically accessible without being usable. Does it allow users to complete the tasks they set out to complete? Are these tasks useful in their daily lives? Do they enjoy the system? Do they trust the system?

In our example case, the system performs the automated, parameterizable generation of a user interface with a visual component and text-to-speech voice output for sending email. The system relies on a grammar of hierarchical components to define the display. The generation algorithm and the components ensure that the output is consistent across multiple devices. The design choices that the algorithm makes are also based on the prediction of utility and cognitive load that a possible output variant will have. We describe the underlying formalism in Section 4.

In what follows, we will present a general methodology for the evaluation of multimodal system outputs which we believe is capable of potential application to a wide variety of evaluation problems. We illustrate the method as it is currently being applied to a specific application (an email client), along with a concrete formalism which easily supports the method by the generation of multiple output variants. Full evaluation results will be presented at the workshop.

2 Recent work

It can take a very long time, on the order of years, to find out if users will really use a system or accept a new paradigm. This acceptance may be dependent on (or impeded by) other factors (e.g. issues with documentation, trust, advertising, cost...)

[1] Speech Application Language Tags, *www.saltforum.org*

(Reiter and Dale, 2000). It is not surprising that a user who has years of experience using a two-dimensional graphical user interface with a keyboard and a mouse will seldom find that a novel interface with 3D graphics and coordinated natural language interaction is a better way to input commands and data, at least at first.

In many projects related to natural language or multimodal dialogue, evaluation is ignored altogether. Experiments with human users are often used mainly as part of the system design process. (as in Feiner and McKeown, 1988). Many of these systems are research prototypes that apply to a limited domain or a limited number of interesting test cases. A user-based evaluation is only feasible once the system is sufficiently stable to allow users to access it over time. While this is an eventual goal, preliminary evaluation will prevent wasting time on substandard user interfaces.

When it comes to the evaluation itself, there are a variety of quantitative measures (time to perform, accuracy, percent agreement of assessments) and qualitative ones (user perceptions of utility, ease of use, and naturalness). (Maybury and Wahlster, 1998)

Qualitative measures also include the study of think-aloud protocols and observation of users. These techniques obviously require a stable and even robust system to be available. At earlier design stages, a cognitive walk-through as well as heuristic evaluation against rules-of-thumb (Cockton et al., 2002) can provide guidance.

Evaluation based on user models employs a simulated user that behaves under, ideally, the same limitations and strategies that a human user would demonstrate. GOMS (goals, operators, methods and selection rules, Kieras, 2002) is a methodology that allows the formalization, even before a system can be used, of elements of a user interface in terms of the knowledge required from a user. A GOMS model seems inappropriate for an adaptable system that may dynamically change the operators available to the user. *Adaptivity involves a constantly changing system model*. Its benefits become clear only in the context of a user under certain external limitations - such as those imposed by parallel, unrelated tasks like driving a car or participating in a conversation.

In SUPPLE, Gajos and Weld (2004) discuss a number of different evaluations of their system. Efficiency evaluations of the generation algorithm show the effect of certain proposed optimizations. Gajos and Weld propose judging the quality of the user interfaces by comparing the system's decisions to those made by human designers under similar constraints regarding the available user interface widgets.

Subjective testing asks a user or designer for their impression and judgment of a system. Reiter and Dale (2000) discuss having experts evaluate both automatically-generated and hand-generated examples. In Comfort (2002), Knight et al. evaluate wearable UIs on the bases of emotion, attachment, harm, perceived change, movement, and anxiety. This set of criteria was generated by multidimensional scaling, and could be adapted for use with other mobile (but not necessarily wearable) devices.

The NASA-TLX system (Hart and Staveland, 1988) is a measure of subjective workload. It has users rate a human-machine environment based on mental demands, physical demands, temporal demands, their own performance, effort, and frustration. A weighted superposition of these features, based on relative ratings given by the user, leads to less between-rater variability than do one-dimensional ratings.

Direct testing compares metrics that are directly related to the interface itself, such as task completion time or success rates. Walker et al. (1997) score dialogue systems with a combination of dialogue success measure and various utterance-related costs. Dialogue success depends on whether slots for a dialogue are correctly filled. Costs for normal utterances and repair moves are counted separately and, like the dialogue success, are weighted using multiple linear regression, with user satisfaction as an external factor. The advantage of this approach is that dialogues may then be scored without an explicit user judgment.

Beringer et al. (2002) modify the framework significantly in order to evaluate free dialogues with their multimodal system, where users are given a much less specific task which cannot be described in terms of necessary and optional slot-filler pairs.

Indirect testing examines things like walking speed or ability to concentrate on outside tasks. Pirhonen et al. (2002) use the percentage preferred walking speed that a user is able to maintain while using the device to evaluate usability.

When doing evaluations, it can be very difficult to compare results from different systems (Bontcheva, 2003). It is important to ensure that both the baseline and adaptive versions of the system are generating in real time.

3 Evaluation

In this section, we propose an evaluation methodology. We expect very similar methods to be ap-

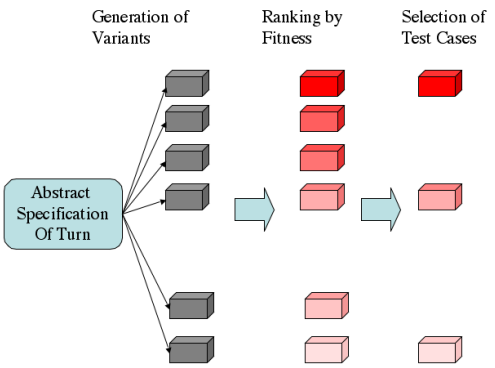


Figure 1: Generating test cases for user evaluation

plicable to dynamic human computer interfaces that generate output based on a number of constraints and define a fitness function to rank solutions. This is true whether or not these systems are multimodal, and without regard to the degree or specific instantiation of the multimodality.

3.1 Method

Our method requires the definition of a candidate fitness function with which multiple candidate output variants can be ranked. The fitness function estimates the projected utility of a variant depending on factors defined by the system designer, and is intended to capture the relevant features of the dialog context, device constraints, user preferences and situation-specific elements. Figure 1 illustrates the process whereby an abstract specification of the dialog turn is received from the dialog manager. This is used to generate many candidate output variants which differ in their informational density and the distribution of information across modes. The fitness function ranks these from best to worst, allowing well-differentiated test cases to be selected from among the best, middle and worst cases for user evaluation in an experimental situation.

Without a gold standard generation system for dynamic multimodal user interfaces to compare against, controlled user trials will allow us to evaluate the usability of the interfaces we create. Key to our approach is the use of sufficiently discriminable output variants (as provided by the ranking) to ensure that we capture a range of user reactions, and can thus gauge the suitability of the fitness function.

Multimodal output generation has been the focus of various grammar-driven generation algorithms, such as COMET (Feiner and McKeown, 1998) or SUPPLE (Gajos and Weld, 2004) which optimize text and graphics layout in documents for print or display on various screens. In dialogue systems, coordinated multimodality can be found in some embodied conversational agents (e.g. Cassell et al., 2000; Wahlster, 2002). Essentially, these systems form intelligent multimedia-interfaces.

3.2 A note on adaptable systems

We make a distinction in this paper between adaptive systems and adaptable ones. Both adaptive and adaptable systems present novel challenges, as user expectations may be confounded, and interface consistency needs to be maintained despite variation in surface realization. We see a role for adaptable systems where an *information bottleneck* arises, e.g. because of the use of a small screen device, situational constraints, or changing user preferences. In these cases, we address the problem of adapting the output to the situational demands by generating multiple variants, and selecting among them based on a fitness function which takes these constraints into account. Adaptive systems, on the other hand, change over a longer time scale to match the user's (or group of users') needs or skills. Our current methodology does not involve sufficient testing time to experiment with such adaptivity, though it could be modified to do so.

3.3 System of ranked variants

Prior to the design of the system, we have identified several areas where we can parameterize the output. The *device model* specifies capabilities of the end-user devices, in particular the screen size and interaction options such as a touch screen or variable buttons as used in many cell phones. The *user model* reflects preferred multimodal interaction (and signal integration) patterns.

The *situation model* reflects external constraints imposed on the interaction with the device. These constraints originate from ambient noise, the users' cognitive workload, manual workload (as in cooking, driving), and sensory workload (watching a movie, walking, listening to a talk).

Our evaluation method controls the adaptation models in order to reflect carefully chosen real-life situations. The more adaptation parameters there are, and the more values that are under consideration, the greater the number of experiments needed to gain sufficient data to show a significant effect of the system's design choices. Over long periods of time, user model adaptation can be problematic, as the system and user may adapt to each other reciprocally. For these practical reason, we decided to vary only the situation and device models.

In an adaptable system in which multiple variants are generated and scored, the scoring metric (see Section 4.2) can be tested by creating versions of the system for the user to interact with. Each version is based on a particular user model, situation model, and device model, and compares the best-rated, worst-rated, and, optionally, one mid-ranked option (according to the fitness function) for each situation. In this manner, the fitness function can be evaluated, as a high degree of discriminability among the variants presented to the user is assured.

Both subjective and objective measures of interface usability can then be used to assess whether the fitness function can boost user satisfaction with a given output variant and, indeed, whether adaptivity is of advantage at all to users in a specific situation. Task completion times, task completion rates (recognition of incorrect system responses), user frustration levels, and user satisfaction are all candidate variables for evaluating the fitness function.

3.4 Scales

There are several different scales on which one can measure a given test. A scale may be absolute or comparative. It may test things subjectively, directly, or indirectly. It may compare different instantiations or different underlying reasons for adaptivity.

In absolute testing, we ask "is this a good user interface?" This is a difficult question to answer. In general, the testing of a user interface comes down to comparing it with other systems. Sometimes that means testing comparison with similar interactions between humans, and other times it may mean comparison with the behavior of a simulated system in a Wizard of Oz scenario (See Section 3.5).

In comparative testing, different output variants, or different versions of the same system are compared to find the relative merits of the systems in the eyes of either users or human designers. This tends to be easier to control, as it can be ensured that the systems are, in fact, comparable.

Even multimodality has multiple scales. A user may use the screen or sound for output, and may use touch screen, keyboard, or voice for input. Multimodal interfaces may also use gesture, haptics, or even smell as a mode for interaction.

Stressors on the user may be internal, as when the user is trying to pay attention to a meeting while checking for an emergency email message, or external, as in a noisy restaurant where one cannot escape distraction. A user may have limits placed on them, based on how public or private their setting is, which may be changed significantly by personal and cultural issues.

An ideal system evaluation would test each relevant metric individually.

3.5 WOz

Wizard of Oz (WOz) testing has an important role to play in creating adaptable systems. It can give an indication of what kinds of interfaces are needed and how those interfaces will be used without the initial cost of building a whole system. However, over reliance on WOz testing can be dangerous: some aspects of a WOz simulation may not be replicable in the actual application (e.g. near-perfect speech recognition). In the special case of evaluating adaptable systems, it can be difficult to ensure sufficient consistency in work of the wizards to ensure that the only differences between trials are those demanded by the adaptation.

3.6 Methodology

We create situations in which we can limit the user's attention to various modalities and collect information on user satisfaction using the NASA-TLX scale (Hart and Staveland, 1988), task completion time, and task success rates.

For mobile systems, the ability for the user to use the system even when distracted is key. To this end, the testing will involve the user being distracted from the requested task. Undistracted usage would parallel a user at his desk or working in some other quiet, non-distracting environment. This situation could serve as a control. For a system which includes a screen display with auditory output and pen and voice input, one form of distraction would be auditory in nature, as that found in a crowded restaurant, while listening to the radio, or while in a meeting and visual and tactile distractions, as found in a meeting, while cooking, or while walking down the street. These situations, of course, must be customized to the aims of the particular modalities of the system in question.

We divide the testing into two phases, for ease of understanding. The first phase tests the fitness function's ability to choose the best of the interfaces for a given situation. This would mean selecting (see Figure 1.) the best, middle, and worst cases for each situation.

The second evaluation phase allows users to use the ideal variant for each situation in other situations. This means evaluating whether the fitness function really does select the optimal design for each situation correctly, as well as determining whether there are distinct ideal adaptations for each situation.

These two sets of tests are very similar. In most cases, the total variant list for all three scenarios will be the same. But the worst-case interface for a user who is subject to auditory distraction may be an unequivocally bad interface, rejected by both users and the fitness function.

In the next section, we describe the application of this methodology to a specific case study: UI on the Fly.

4 UI on the Fly

In this section, we outline our multimodal generation system, which has a grammar and a fitness function at its core. The system is currently undergoing a full evaluation.

4.1 MUG

Multimodal functional Unification Grammar is a non-deterministic grammar (Reitter et al., 2004) that generalizes decisions about how to deliver content in a multimodal user interface. A grammar in this formalism specifies an adaptable user interface using natural language. The formalism is an extension of functional unification grammar (Kay, 1979; Elhadad & Robin, 1992) that ensures content coordination in the different modes. The formalism allows for the generation of multimodal user interfaces.

The application of a MUG yields several solutions that are faithful to the original specification and consistent and coherent across the different output modes. But only one of these solution is considered the best one – according to a fitness function, which incorporates the user, situation and device models.

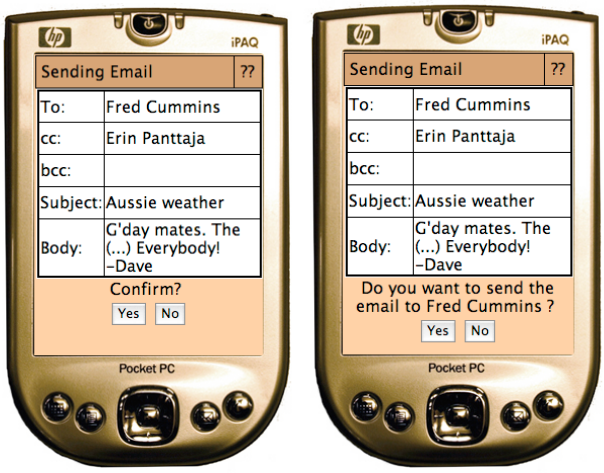
We demonstrate it in the context of a limited-domain user interface for a mobile personal organizer.

MUG is a set of *components*. Each of them specifies a realization variant for a given partial semantic or syntactic representation, similar to a rule in a production grammar. The components are attribute-value structures. The generation algorithm chooses components from the grammar and unifies them iteratively with the original input specification, thereby instantiating several layers of output planning and surface form realization.

While allowing for cross-modal consistency, the attribute value matrices 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 (for example, deictic pronouns).

There may be several competing components in the grammar for a particular job, all of which unify with a given partial semantic input. This translates to a design choice the system has to make. Design choices are never made individually: They often depend on other choices. For example, choosing to render the full subject line of an email in a display variant on a small screen device might not leave enough room for the (more important) name of the recipient. The system therefore evaluates the variant as a whole[2].

Design variants are ordered according to the outcome of a fitness function. The best variant is is that optimally adapted to the given situation, user, and device (see Section 4.2).



(a) (b)

Figure 2: a) Voice: "Send the email regarding Aussie Weather now?". b) Voice: "Send the email now?"

MUG enables some feedback to the dialogue system about which parts of the dialogue semantics were actually realized in a given situation, as addressed by (Wahlster, 2002). A mode-specific attribute (*realized*) is instantiated by the grammar for each semantic entity that has been incorporated in to the output in the given mode.

4.2 Fitness function

Finding the best solution to the hard constraints defined by the grammar can be seen as optimization problem. What do we optimize? In other words, what is a *good* solution?

There are different approaches to formulating the scoring function, and usually there are several con-

[2]Practically, best-first / A* search algorithms may be used to optimize the search for an optimal solution. But that has no consequences for the evaluation.

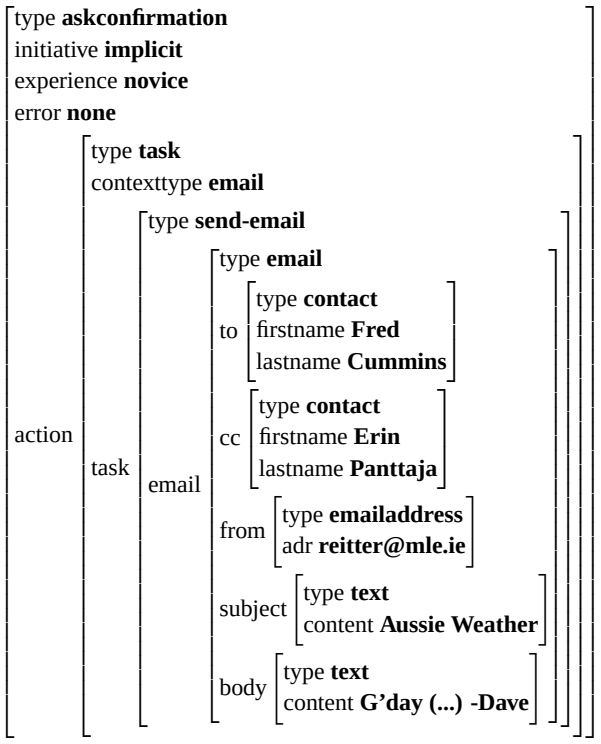


Figure 3: Input representation: confirmation of sending of an email. E-Mail body text abbreviated.

siderations that are weighted. In SUPPLE (2004), Gajos and Weld predict the effort a user has to make in order to reach each element of an interface. Such a user-model driven fitness function still leaves the designers with many choices – for example, whether the cost for each user interface element should also depend on the maximum-likelihood probability for its actual use.

In UI on the Fly, we generate simpler, but multimodal interfaces for small-screen devices. The number of elements shown on a screen is small, and the user interface widgets defined by the MUG do not differ greatly in the time it takes to operate them. We see major cost differences, however, in the degree to which the voice modality is used (it takes time to listen to system speech). We therefore model the utility of a particular multimodal output as a combination of reading / listening time versus the benefit of presenting important information.

By default, we try to be as helpful as possible, with information that is deeply embedded in the semantic structure receiving lower priority than higher elements. Redundant information, that is, information that is presented in both modes, does not receive a double benefit. Information that needs to be presented according to the assumed dialogue management component leads to a heavy penalty if it is left out during generation stage.

The trade-off lies in the cost of the output, which is estimated in terms of the cognitive load imposed on the user, who needs to read new text on the screen or listen to the voice output.

These constraints are formalized in a score that is assigned to each variant ω, given a set of available Modes M, a situation model $< \alpha, \beta >$, a device model ϕ:

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

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

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 request is encoded in the *realize* attribute, the actual realization feedback is given in the mode-specific attribute *realized*.

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 equals the utterance output time for a text spoken by the text-to-speech system, or an estimated reading time for text on the screen.

The a utility/time normalization coefficient λ can be manually estimated or learned from a corpus. If the evaluation setup is used, λ will be acquired from a separate training partition of the data.

5 Evaluating UI on the Fly

In carrying out preliminary evaluations of UI on the Fly, we need to bear in mind that a) it is not a complete dialogue system, but b) it should be evaluated with human subjects.

One of the ideas behind UI on the Fly is that local decisions about the generation of multimodal output may incur a local cost, but benefit the dialogue. For example, a certain output may contain more information and be, thus, longer. But in turn, the system can save an additional confirmation step. Such decisions can only be evaluated in the context of a full dialogue.

The core component of the generation system is the fitness function described in Section 4.2.

To evaluate whether the prediction of cognitive complexity is realistic, we will measure task completion time for a predefined task that involves sending an e-mail. We will compare the performance of a system that chooses the output variant deemed optimal against one that always chooses a mid-ranking output variant.

5.1 Recreating usage situations in the laboratory

In an attempt to broaden the range of respondents in this evaluation, it will be built as a web page to be used in the user's own office or home. This will allow us to test the system with a variety of different computer-literate users.

The evaluation of the system in the laboratory recreates the important mode-specific characteristics of a range of hypothetical situations.

The users will be given a computer-game task as an auditory, visual, and tactile distraction. This will be a flash program in the testing web page. In order to ensure that the user is paying attention to the game, their score will be recorded. The time they are allowed for each turn will also be limited.

This will not exactly mirror the target task of walking down a busy street, but will simulate some of the distraction and cognitive load.

5.2 Devising tasks

Each user will be asked to send one email message using the system, while performing the distraction task. This message will be selected from a bank of three messages. They will not be using a full dialogue system: each turn of the task will be represented by an output turn from the system, then a corresponding input from the user. Errors by the user will be ignored by the testing system (though recorded for the evaluation).

The tasks will all be web site-based, but will simulate usage of either a small screen cell phone or a PDA-device (see Figure **??**).

Each task will involve approximately five system turns, and the two selected variants will be the first and middle option from a pool of 30-90 variants created by the system for each turn.

Some of the test turns will involve mistakes on the part of the system. Whether or not the user catches these mistakes will be recorded as the user's error detection rate.

There will be three different tasks, and two different devices. There will be 10 users of each task, for a total of sixty users.

5.3 Measuring quality

We will be collecting several different kinds of information from each user. We will start with a user questionnaire, to establish their background and that their system is sufficient for the experiment.

For each system turn, we will record the task completion time and whether the task was completed successfully.

After each task, we ask the user, how appropriate the system output was in the given situation (user satisfaction). By pairing user satisfaction ratings for different utterance types we can show, whether the fitness function and the user satisfaction data show a significant correlation, and whether the situation-specific adaptation has a significant effect on the user satisfaction.

6 Conclusion

We have discussed several approaches to the evaluation of adaptable, multimodal dialogue systems and their output generation components. We have presented a case study giving a preliminary outlook of how to evaluate a concrete instantiation of such a system under realistic constraints. Meaningful evaluation, even of a single subsystem with limited functionality, is feasible.

This methodology can be applied to any system that uses a fitness ranking to choose the optimal interface to present to a user. Each parameter (situational, user, or device) added to the system will, of course, increase the number of tests (and users) required, of course, but each additional constraint can be easily be compared against tests already completed.

User distraction levels and different device models are not, in themselves, applicable to every multimodal system, but each system will have its own set of constraints that will be used to define the output variants generated and the fitness function used to select the optimal variant.

Acknowledgements

This research was in part funded by the European Commission under the FASiL project, contract number: IST-2001-38685.

References

Nicole Beringer, Ute Kartal, Katerina Louka, Florian Schiel, and Uli Türk. 2002. PROMISE - a procedure for multimodal interactive system evaluation. In *Proceedings of the LREC 2002 Workshop on Multimodal Resources and Multimodal Systems Evaluation*, Athens, Greece.

Kalina Bontcheva. 2003. Reuse and challenges in the evaluation of NLG systems. In *Proceedings of the EACL-2003 Workshop on Evaluation Initiatives*, Budapest, Hungary.

J. Cassell, J. Sullivan, S. Prevost, and E. Churchill. 2000. *Embodied Conversational Agents*. MIT Press, Cambridge, MA.

G. Cockton, D. Lavery, and A. Woolrych. 2002. Inspection-based evaluations. In J. Jacko and A. Sears, editors, *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*. Lawrence Erbaum Associates, Mahwah, NJ.

Michael Elhadad and Jacques Robin. 1992. Controlling content realization with functional unification grammar. In R. Dale, E. Hovy, D. Roesner, and O. Stock, editors, *Proceedings of the Sixth International Workshop on Natural Language Generation*, pages 89–104. Springer Verlag. Lecture Notes in Artificial Intelligence.

Steven K. Feiner and Kathleen R. McKeown. 1998. Automating the generation of coordinated multimedia explanations. In Mark T. Maybury and Wolfgang Wahlster, editors, *Intelligent User Interfaces*. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

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

S. G. Hart and L. E. Staveland. 1988. Development of a multi-dimensional workload rating scale: Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati, editors, *Human mental workload*. Elsevier, Amsterdam, The Netherlands.

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

David Kieras. 2002. Model-based evaluations. In J. Jacko and A. Sears, editors, *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*. Lawrence Erbaum Associates, Mahwah, NJ.

J. F. Knight, C. Baber, A. Schwirtz, and H. W. Bristow. 2002. The comfort assessment of wearable computers. In *Proceedings of the Sixth International Symposium of Wearable Computers*, Seattle, Washington.

M. T. Maybury and W. Wahlster, editors. 1998. *Intelligent User Interfaces*. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

A. Pirhonen, S. A. Brewster, and C. Holguin. 2002. Gestural and audio metaphors as a means of control for mobile devices. In *Proceedings of ACM CHI2002*, Minneapolis, Minnesota, USA. ACM Press, Addison-Wesley.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

David Reitter, Erin Marie Panttaja, and Fred Cummins. 2004. UI on the fly: Generating a multimodal user interface. In *Proceedings of HLT-NAACL-2004*, Boston, Massachusetts, USA.

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.

M. Walker, D. Litman, C. Kamm, and A. Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In Philip R. Cohen and Wolfgang Wahlster, editors, *Proceedings of the ACL-EACL-1997*, pages 271–280, Somerset, New Jersey. Association for Computational Linguistics.