# Creating Tangible Interfaces by Augmenting Physical Objects with Multimodal Language

David R. McGee and Philip R. Cohen
Center for Human Computer Communication
Department of Computer Science and Engineering
Oregon Graduate Institute
Portland OR 97006 USA
+1 503 748 1602
{dmcgee, pcohen}@cse.ogi.edu
http://www.cse.ogi.edu/CHCC/

## ABSTRACT

Rasa is a tangible augmented reality environment that digitally enhances the existing paper-based command and control capability in a military command post. By observing and understanding the users' speech, pen, and touch-based multimodal language, Rasa computationally augments the physical objects on a command post map, linking these items to digital representations of the same—for example, linking a paper map to the world and Post-it™ notes to military units. Herein, we give a thorough account of Rasa's underlying multiagent framework, and its recognition, understanding, and multimodal integration components. Moreover, we examine five properties of language—generativity, comprehensibility, compositionality, referentiality, and, at times, persistence—that render it suitable as an augmentation approach, contrasting these properties to those of other augmentation methods. It is these properties of language that allow users of Rasa to augment physical objects, transforming them into tangible interfaces.

## Keywords

Human factors, augmented reality, mixed reality, multimodal interfaces, tangible interfaces, and invisible interfaces

## 1 INTRODUCTION

In air traffic control centers, military command posts, and hospital emergency rooms, life-and-death decisions must be made quickly in the face of uncertain information. In these high stress environments, professionals put a premium on safety, timeliness, team cohesiveness, and mutual awareness. Consequently, controllers, officers, and doctors have chosen tools that are robust, malleable, physical, and high in resolution; often these tools begin with a paper and pencil. Moreover, professionals discard computational tools that do not have these properties. We assert that these properties must be supported when "upgrading" physical tools to computational ones in order for them to be effective and accepted. Indeed, it is now possible, using the language of the task, to augment real world artifacts, creating digital tools as robust, malleable, and portable as paper and other physical artifacts.


*IUI'01,* January 14-17, 2001, Santa Fe, New Mexico, USA.


## 2 OVERVIEW

This paper describes a design for augmenting environments that minimally perturbs existing work practices by allowing users to take advantage of their current tools and processes, in order to obtain benefits from digitization. By *augment*, we mean to manipulate a physical artifact so that it is publicly, and in most cases permanently, recognized to represent or denote something else. This kind of natural augmentation is an activity that humans perform every day. We often use common physical objects to represent other entities and their relations when the 'true' objects are not nearby: (1) we sketch out a map on a napkin, so that guests can find their way to our party; (2) we use a pencil and coffee cup to describe what happened in an accident on the freeway between a delivery truck and a motorcycle; (3) we array Post-it™ notes on a white board or other surface during planning and other tasks.

Based on our observations of command posts in the U.S. military, we identified a set of constraints on the design of computing systems for command and control. In a recent paper, we showed that contemporary object tagging (i.e., augmentation) approaches fail to meet these constraints [10]. Here we discuss why language, because of its generativity, compositionality, comprehensibility, and referentiality meets each criterion, making it an especially attractive augmentation method. Finally, we present a detailed description of Rasa—a system that understands the language officers use in military command posts to naturally augment physical objects on a paper map and updates digital command and control systems in response to the use of that language.

## 3 WORK PRACTICE

At Ft. Leavenworth, Kansas, and at other military bases, we observed commanders and their subordinates engaging in command and control of armed forces. The photograph in Figure 1 was taken during an especially frenetic period in the command post.

On the left is a rear-projected SmartBoard™ and on the right is a Sun Microsystems workstation. Several other systems, not captured in the photograph, are in the immediate foreground. On each is one of the Army's latest command and control ($C^2$) software systems. These systems are multi-year, high-dollar investments created to aid the commander and his staff by providing situational awareness. Notice, however, that no one is using these systems during this critical phase of work. Rather, the commander and his staff have chosen to use a set of tools other than the computerized ones designed for this task. They have quite purposefully turned their backs on computer-based tools and graphical user interfaces, preferring to use instead an 8-foot-high by 6-foot-wide paper map, arrayed with Post-it notes (Figure 2).



其实也是一样的，相当于给一个physical object匹配了一个software层面的意义，而且是customizable的
physical interface不在局限于button和各种系统本身的东西，而是任何可以被识别的物理系统都可以成为这个系统的physical interface

**Figure 1. State of the art military command and control systems in action.** *Photo courtesy of William Scherlis.*

**Figure 2. What commanders prefer.** *Photo courtesy of William Scherlis*

This large map has the same two-fold purpose as the $C^2$ systems above: (1) to depict the terrain, its occupants (military units consisting of soldiers and machinery), their position, and capabilities; and (2) to overlay that information with a graphical rendition of the daily plan for the force. A symbol representing a unit's functional composition and size is sketched in ink on each Post-it. As unit positions arrive over the radio, the Post-its representing these units are moved to their respective positions on the map. The users establish associations between the Post-it notes and their real-world counterparts with a standardized, compositional language, used since Napoleon's time, which is capable of denoting thousands of units.

Officers keep the map up-to-date by constant communication both up and down the units' organizational hierarchy. It is hoped that the computerized systems will reduce the analog communication flow by providing situational awareness digitally at all levels. However, because the computational interface lacks certain properties that are essential for decision-making in this environment, the officers continue to choose paper tools in favor of their computational alternatives. They do so because paper has extremely high resolution, is malleable, cheap, lightweight, and can be rolled up and taken anywhere. Additionally, people like to handle physical objects as they collaborate. As officers debate the reliability of sensor reports and human observation to determine the actual position of units in the field, they jab at locations on the map where conflicts may arise. They also pick up Post-it notes and hold them in their hand while they debate a course of action.

Rasa is designed to combine the advantages of both paper and computation. Furthermore, it does so without substantial task overhead by perceiving the pre-existing augmentations resulting from the users' interacting multimodally with these physical objects. Consequently, users can continue to employ familiar tools and procedures, which in turn create automatic couplings to the digital world. In other words, the physical artifacts become the computational interface. Moving a Post-it note on the map moves it in the digital system. In the next section, we examine other methods that have been used to augment physical environments.

### 3.1 Augmenting physical objects

Researchers use two methods to sense physical objects in an augmented environment:

- Sensing their physical properties (weight, shape, color, size, location, etc.);
- Affixing sensable properties to them, hereafter called simply "tags" (e.g., bar codes, glyphs, radio-frequency identifier tags).

In either case, the purpose is to determine which objects in an environment are being physically manipulated and to perform a corollary computational manipulation. We considered every available tagging approach in both categories when we began to design Rasa, but none of them met the physical constraints (e.g., power and size issues) needed to augment a Post-it note. Moreover, none fulfilled the design constraints provided below and discussed in detail in [10].

| | |
|---|---|
| ***Minimality Constraint*** | Changes to the work practice must be minimal. The system should work with user's current tools, language, and conventions. |
| ***Human Performance Constraint*** | Multiple end-users must be able to perform augmentations. |
| ***Malleability Constraint*** | The meaning of an augmentation should be changeable; at a minimum, it should be incrementally so. |
| ***Human Understanding Constraint*** | Users must be able to perceive and understand their own augmentations unaided by technology. Moreover, multiple users should be able to do likewise, even if some are not present either physically or temporally. |
| ***Robustness Constraint*** | The work must be able to continue without interruption should the system fail. |

With respect to these constraints, none of the previous tagging approaches (1) provides a persistent representation of the augmentation that would make it robust to the frequent system failures (e.g., communications, power); (2) is decipherable without computational aid (i.e., if tag readers or other systems fail, no one can understand what the tag means); (3) has a natural way for the end-user to create the augmentations; (4) could be introduced into this work practice without substantial change.

In the next section, we discuss how language overcomes these deficiencies, examining its properties that commend it as a tool for augmenting the meaning of physical objects.

## 4 LANGUAGE

As we discuss language, we mean an arrangement of perceptible "tokens" that have both structure and meaning. This definition is meant to subsume both natural spoken and written languages, as well as diagrammatic languages such as military symbology. In this sense, language is generative, compositional, comprehensible and referential. Moreover, depending on whether it is written or spoken, it can be permanent or transitory, respectively. These attributes make language a particularly suitable candidate for creating augmentations.

By *referentiality*, we mean that we can quickly and radically change the shared meaning that an object has in a community of use by bestowing a new name or new meaning upon it. In our work practice, we can generate placeholders for things like groups of people, simply by saying "suppose this is Charlie Company," and pointing at a rock or by drawing a symbol of Charlie Company on a Post-it note. Not only is the Post-it note now meant to represent Charlie Company for the creator, but any observer, including a computational one, can *comprehend* the "augmentation" of the Post-it in virtue of the shared language.

By *generativity*, we mean that language allows us to express completely new concepts by combining tokens from our token set in an original way, according to a generalized grammar. Our ability to establish the types of referential relationships described above is part of our generative capacity to use language. Furthermore, because of the compositionality of the semantic interpretation, listeners have a reasonable chance of comprehending the meaning of the each new utterance, given its context.

The *comprehensibility* of utterances is aided by the *compositionality* of language in general, and of the military symbology language in particular. This language consists of shapes to indicate friend (rectangle) or foe (diamond), a set of lines, dots, or "X's" to indicate unit size (squad to army), and a large variety of symbols to indicate unit function (e.g., mechanized, air defense, etc.) denoted by combinations of meaningful diagrammatics. For example, an armored reconnaissance unit's symbol is a combination of the marks used for armor and for reconnaissance. In virtue its compositional nature, the symbol language can denote thousands of units, the types of personnel and machinery in them, and the unit's place among the thousands. Because of this compositionality, soldiers are able to comprehend and generate complex concepts in terms of their parts, which is essential for the collaborative planning that is a principal component of this decision environment.

Moreover, since written language is *permanent*, it leaves behind a persistent trail when paper-based augmentations are incrementally applied. Spoken language, on the other hand, is convenient when the user would prefer a lack of persistence. For example, rather than update the permanent shared understanding conveyed by the map, spoken language is often used to name or refer to objects casually when collaborators are co-present and working synchronously.

None of these characteristic properties of language—its generativity, compositionality, referentiality, comprehensibility, and its robustness to failure in the written form—is present in previous augmentation approaches. One might imagine designing a user interface to accompany the creation and reading of tags that offered these capabilities. However, their robustness in the environment would be questionable. Moreover, any change that causes more than minimal disruption to these users' highly learned behaviors, such as the introduction of computer interfaces, will likely be resisted.

In summary, the language used in the command post offers an ideal means for augmenting physical objects like its map and Post-it notes. However, in order to take advantage of the users' reliance on language, a system must be capable of understanding it. In the following section, we present Rasa—a system that enables multimodal understanding of spoken and gestural language in such augmented environments—derived from our earlier work on QuickSet [4].

## 5 DESCRIPTION OF USE

When the user first sets up Rasa in the command center, he unrolls his map and attaches it to a SmartBoard or other touch-sensitive surface (see Figure 3). Users can register any type of map (e.g., paper map, satellite photograph, drawing) to its position in the real world by tapping at two points on it and speaking the coordinates for each. Immediately, Rasa is capable of projecting information on the paper map, or some other display, from its digital data sources. For example, Rasa can project unit symbology, other map annotations, 3D models, answers to questions, tables, etc.

**Figure 3. Users collaborating with Rasa**

Figure 4 provides an example of the information flow in Rasa. As a user receives a radio report identifying an enemy reconnaissance company, (1) he draws a symbol denoting the unit on a Post-it. Simultaneously, he can choose to modify the object with speech. For instance, he draws the reconnaissance company unit symbol in Figure 4 and at the same time gives the unit the name "*Advanced guard*" via speech. (2) The system performs recognition of both speech and gesture in parallel, producing multiple hypotheses. (3) These hypotheses are parsed into meaning representations, and (4) are submitted for integration. (5) Some time later, the user places the Post-it on a registered map of the terrain at position 96-94. (6-8) This gesture is recognized and parsed, then also submitted for integration. After successful fusion of these inputs, Rasa says, "*Confirm: Enemy reconnaissance company called 'advanced guard' has been sighted at nine-six, nine-four.*" The user can then cancel the action in response to a system error. If it is correct, the user need not confirm. Confirmation or cancellation can occur with speech, or users can create buttons multimodally, again using

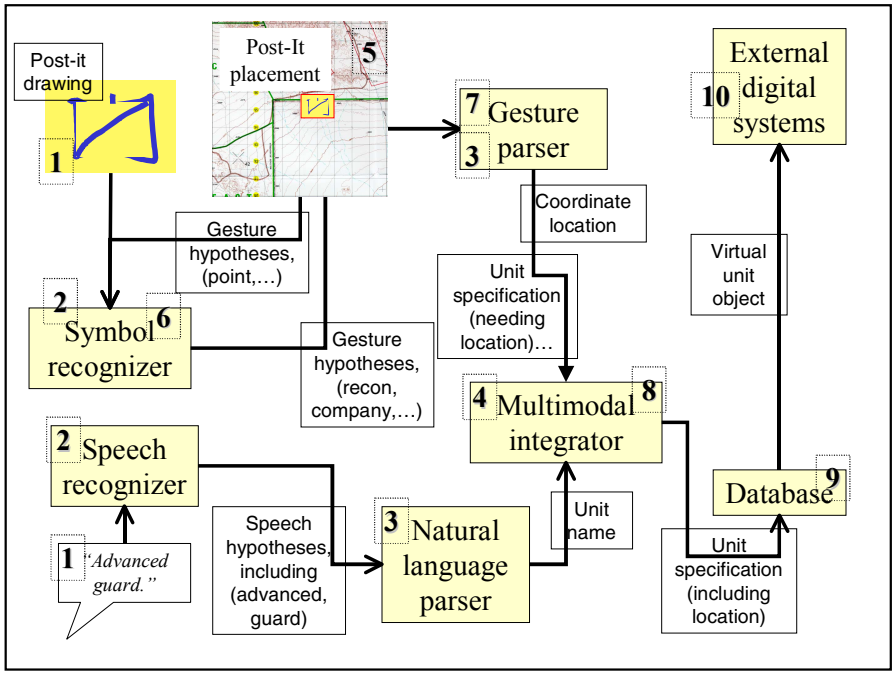


**Figure 4. Example of data and process flow in Rasa**

Post-it notes, for confirming and canceling actions. Further action by the user, such as the placement of another unit or the movement of one, implies that the prior command should be confirmed [9]. After confirmation, the unit is inserted into a database, which triggers a message to external digital systems.

The next section describes the system architecture that makes this type of augmented, multimodal interaction possible. Following this description, we will examine this example of Rasa's use in further detail.

## 6 ARCHITECTURE

Rasa consists of autonomous and distributed software components that communicate using an agent communication language in the Adaptive Agent Architecture (AAA) [7], which is backwards compatible with the Open Agent Architecture (OAA) [3].

### 6.1 Agent Framework

The AAA is a robust, facilitated multi-agent system architecture specifically adapted for use with multimodal systems. A multi-platform Java agent shell provides services that allow each agent to interact with others in the agent architecture. The agents can dynamically join and leave the system. They register their capabilities with an AAA facilitator, which provides brokering and matchmaking services to them.

Rasa's understanding of language is due to multiple recognition and understanding agents working in parallel and feeding their results via the AAA facilitator to the multimodal integration agent. These agents and the human computer interface agents are described below.

### 6.2 User Interfaces

With Rasa, users first draw on a pad of Post-it notes affixed to a Cross Computing iPenPro™ digital pen tablet. A paper interface agent, running on the computer system connected to the tablet, captures digital ink, while the pen itself produces real ink on each note. However, there is no computer or user interface visible, other than the Post-its.

In addition to the Post-its, map annotations can be created multimodally and then projected onto the paper map or a separate surface. As new units are placed on the map, a colored shadow indicating their position and their disposition (friendly or enemy) is overlaid on the paper map. A table of all units present and their combat strength is projected next to the map as visual confirmation of the status of each unit. This type of table is often found next to the map tool in command posts. "Control measures," such as barbed wire, fortifications, berms, etc., can be drawn directly on the map's plastic overlay. The resulting military icon for that object is projected directly onto the map.

Conceptually, Rasa's user interfaces act as transparent interaction surfaces on the paper. Whenever the user touches the map and the touch-sensitive surface beneath it or uses the pen on the iPenPro tablet, she is interacting with Rasa. As she does, messages describing the digital ink left behind on those surfaces are sent to the facilitator for distribution to the relevant agents.

One additional invisible interface is used, text-to-speech, which can be enabled or disabled on command as needed. Each interaction with Rasa produces not only visual confirmation if a projector is present, but also audible confirmation.

### 6.3 Recognizers

Interaction with any of these paper surfaces results in ink being processed by Rasa's gesture agent. At the same time, speech recognition is also enabled. Each recognizer provides input to the integrator. These agents and their abilities are discussed below.

#### *6.3.1 Gesture*

Rasa's gesture agents recognize symbolic and editing gestures, such as points, lines, arrows, deletion, and grouping, as well as military symbology, including unit symbols and various control measures (barbed wire, fortification, boundaries, axes of advance, etc.) based on a hierarchical recognition technique called Member-Team-Committee (MTC) [17].

The MTC weighs the contributions of individual recognizers based on their empirically derived relative reliabilities, and thereby optimizes pattern recognition robustness. It uses a divide-and-conquer strategy, wherein the members produce local posterior estimates that are reported to one or more "team" leaders. The team leaders apply weighting to the scores, and pass results to the committee, which weights the distribution of the teams' results. Using the MTC, the symbology recognizer can identify 200 different military unit symbols, while achieving a better than 90% recognition rate.

#### *6.3.2 Speech*

Rasa receives its spoken input from a microphone array attached to the top of the SmartBoard directly above the map, or from wireless, close-talking microphones. The speech agent uses Dragon Systems NaturallySpeaking or other Microsoft SAPI-compliant engines, which are continuous, speaker-independent recognizers, though training can be used to increase their accuracy. The recognizers use context-free grammars, producing n-best lists for each phrase. Rasa's vocabulary is approximately 675 words, and the grammar specifies a far greater number of valid phrases.

Once each recognizer generates a set of hypotheses, both the speech and gesture recognizers forward to their respective parsers the results, their scores and time stamps.

## 6.4 Parsers

### *6.4.1 Natural language*

A definite-clause grammar produces typed feature structures—directed acyclic graphs (DAGs) of attribute value pairs—as meaning representation. For this task, the language consists of map-registration predicates, noun phrases that label entities, adverbial and prepositional phrases that supply additional information about the entity, and a variety of imperative constructs for supplying behavior to those entities or to control various systems.

### *6.4.2 Gesture*

The gesture parser also produces typed feature structures, based on the list of recognition hypotheses and probability estimates supplied by the gesture recognizer. Typically, there would be multiple interpretations for each hypothesis. For example, a pointing gesture has at least three meaningful interpretations—a selection is being made, a location is being specified, or the first of perhaps many point locations is being specified. In the next section, we will examine how these multiple interpretations are weighed in the multimodal integrator.

## 6.5 Multimodal Fusion

Rasa's multimodal integration technology uses declarative rules to describe how the meanings of input from speech, gesture, or other modalities must be semantically and temporally compatible in order to combine. This fusion architecture was preceded by the original "Put-That-There" [1], and other approaches [6, 12, 13].

In Rasa, multimodal inputs are recognized, and then parsed, producing meaning descriptions in the form of typed feature structures. The integrator fuses these meanings together by evaluating any available integration rules for the type of input received and those partial inputs waiting in an integration buffer. Compatible types are fused, and the candidate meaning combination is subject to any constraints specified in the rule (e.g., spatial, temporal, etc.). Successful unification and constraint satisfaction results in a new set of merged feature structures. The highest ranked semantically complete feature structure is executed. If none is complete, they all wait in the buffer for further fusion, or contribute to the ongoing discourse as discussed below.

### *6.5.1 Typed Feature Structure Unification*

Semantic compatibility and fusion is assured via unification over typed feature structures [2]. Unification determines the consistency of two or more representational structures, and if consistent, combines them. This type of unification is a generalization of term unification in logic programming languages, such as Prolog. Typed feature structure unification requires feature structures to be compatible in type (i.e., one must be in the transitive closure of the subtype relation with respect to the other). The result of a typed unification is the more specific feature structure in the type hierarchy. Typed feature structure unification is ideal for multimodal integration because it can combine complementary or redundant input from different modes, yet it rules out contradictory inputs.

In the next section, we describe how Rasa's multimodal fusion works (described fully in [5]). Following that we will present a short example of multimodal fusion in Rasa. Then we will examine how Rasa currently supports human-computer discourse.

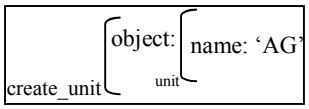


**Figure 5. Typed feature structure from spoken utterance "*Advanced guard*."**

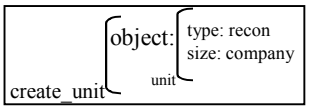


**Figure 6. Typed feature structure resulting from drawing recon company.**

### *6.5.2 Example of Multimodal Fusion in Rasa*

To demonstrate how multimodal fusion works in Rasa, let's return to the example given above, in which an officer adds a new unit to Rasa's augmented map. The user's speaking the unit's name ("advanced guard") generates a typed feature structure of type `create_unit` with an `object` attribute. The `object` attribute contains a feature structure of type `unit`. The `unit` feature structure contains one attribute, called `name`. The `name` attribute, in this example, stores the value for the name Advanced Guard, 'AG' (Figure 5). To name a new unit, the name should be uttered while drawing a symbol that specifies the remaining constituents for a unit, such as the reconnaissance company symbol shown in Figure 4. This symbol is recognized as a reconnaissance company and ultimately assigned the feature structure in Figure 6. This feature structure is similar to the one shown in Figure 5, except that it specifies the `type` and `size` attributes of the unit feature as `recon` and `company`, respectively.

Rasa's fusion approach uses a multidimensional parser, or *multi-parser*, based on chart parsing techniques from natural language processing. Edges in the chart are processed by declarative multimodal grammar rules. In general, these rules are productions LHS ⇐ DTR1 DTR2 … DTRn; daughter features are fused, under the constraints given, into the left-hand side. The shared variables in the rules, denoted by numbers in square brackets, must unify appropriately with the inputs from the various modalities, as previously described.

One of Rasa's multimodal grammar rules, see Figure 7, declares that partially specified units (`dtr1` and `dtr2`) can combine with other partially specified units, so long as they are compatible in type, size, location, and name features, and they meet the constraints. It is expected that this rule will fire successfully when the user is attempting to create the particular unit using different modalities synchronously. `dtr2` is a placeholder for gestural input (note the location specification) and `dtr1` for spoken input, but this need not be the case.

After successful fusion, any constraints are then satisfied using a Prolog meta-interpreter. For example, the timing constraints for this rule (the "`overlap or follow`" rule specification) guarantee that the two inputs will temporally overlap or that the gesture will precede the other input by at most 4 seconds. This constraint is based on empirical, wizard-of-oz experiments with multimodal map interfaces [14].

Figure 7 demonstrates partial application of the rule and shows that after fusion the left-hand-side is still missing a location feature for the unit specification. In the next section, we will describe how Rasa uses completeness criteria to notice this missing feature and query the user for it.

## 6.6 Discourse in Rasa

The completeness criterion and constraint satisfaction work together to allow each feature structure to carry along a rule that specifies what features are needed to complete the structure. In this way, during feature structure evaluation, rules can fire that instruct Rasa

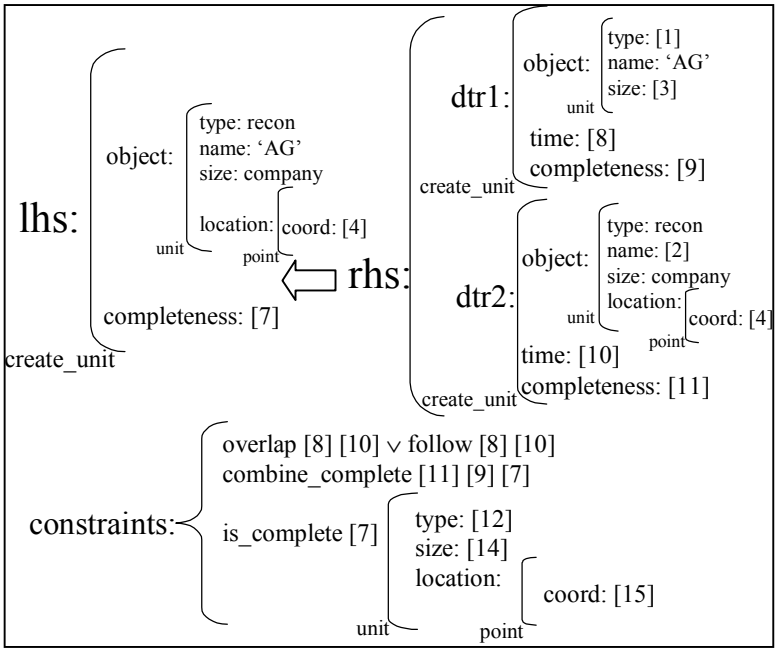


**Figure 7. Multimodal grammar rule for partial unit fusion.**

to formulate subdialogues with the user or another agent to request the missing information, similar to Smith's "Missing Axiom Theory" [15].

For example, Figure 7 shows the completeness criterion for units. This feature structure captures the attribute names for each of the attributes that must have a value before the feature structure is complete. In the example shown here, the criterion stipulates that every unit must have a unit feature, with values for *type, size*, and *location*. The location feature must have a feature-structure of type *point*. This point feature must have a value for its *coord*.

This information can then be used by Rasa to produce queries when one of the values is missing. After a tunable delay, Rasa asks the user for the position of those units that are fully specified, except for a missing location feature. Users respond by placing the Post-it note on the map or by disconfirming the operation and throwing the Post-it away. For example, if the Post-it note representing the reconnaissance company has not been placed on the map within 10 seconds, Rasa would respond by saying "Where is the reconnaissance company called 'Advanced guard'?"

Discourse rules can be declaratively specified in Rasa to promote mixed-initiative, collaborative dialogue. Remaining applications for this capability are left for future work.

## 7 DISCUSSION

We have recently conducted a field test of Rasa with nine Marine Corps Personnel from the First Marine Expeditionary Force (I-MEF) in order to evaluate its performance and assess its ability to co-exist with paper tools and practices and to overcome the issues of robustness, resolution, portability, etc., described earlier. We will discuss the detailed results of this experiment in a forthcoming paper. In general, the subjects agreed on several points:

1. Work was able to continue without pause when the computing system (i.e., Rasa) failed.
2. Digital information was quickly recoverable once Rasa was restored.
3. Rasa was as easy or easier to use than paper and preferred over paper.
4. Rasa did not impede performance of the task.

## 8 RELATED WORK

Several recent approaches have successfully augmented paper in novel ways [8, 11, 16]. However, none of these approaches treats the existing language of work as anything other than an annotation. These systems can capture the augmentations, but cannot understand them. Consequently, though they augment paper, and even support tasks involving written language, they are unable to take advantage of its properties as a means of augmenting the physical objects, as Rasa has done.

For example, Mackay et al. [8] have analyzed why certain paper-based artifacts in air traffic control environments have been impossible to replace with computerized artifacts. They have even suggested augmented design alternatives for flight strips in air traffic control centers. These strips, though suitably capturing the handwritten annotations for remote collaboration and storage and certainly robust to failure, do not capture the meaning of the annotations for digital processing, database update, and the like, and therefore could certainly benefit from something like Rasa's multimodal understanding capabilities.

## 9 CONCLUDING REMARKS

We have presented Rasa, an environment where physical objects are computationally augmented by a system that observes users' work. Rather than force the user to adopt radically new ways of working, we instead propose to augment the pre-existing tools in her environment, bringing computing to her and her tools, rather than the other way around.

We have shown how language has properties that are especially suited for augmenting artifacts. In virtue of the referentiality of language, users create augmentations; in virtue of its generativity, these augmentations are potentially never before seen; in virtue of its compositionality, a large set of augmentations are possible; because this compositional language is is shared, the author and others can comprehend them immediately; finally, in virtue of its persistence, the augmentation remains understandable in the face of failure.

Rasa leverages these benefits of language in understanding the augmentations that officers in command posts place on Post-it notes and paper maps, resulting in the coupling of these placeholders with their digital counterparts. This coupling creates a computational environment that not only retains the current team-oriented work practice in its entirety, but also retains the advantages of paper—most especially robustness to computing failure.

We chose this approach because we had no alternative. Users have set aside their computational aids and, with good reason, have resorted to paper tools instead. Our view is that they should not have to choose between the two sets of advantages. Rather we can, and often should, begin to augment their choice of tools in a way that leaves them and the work changed only by the benefits offered by digitization.

## 10 ACKNOWLEDGEMENTS

This work was supported in part by the Command Post of the Future Program, DARPA under contract number N66001-99-D-8503, and also in part by ONR grants: N00014-95-1-1164, N00014-99-1-0377, and N00014-99-1-0380. The Adaptive Agent Architecture infrastructure was supported by the DARPA CoABS program and

US Air Force Research Laboratory under contract number: F30602-98-2-0098. Thanks to our colleagues Sanjeev Kumar, Sharon Oviatt, Misha Pavel, Richard Wesson, Lizhong Wu, and everyone who contributed to this work. The opinions and any errors that appear herein are the authors'.

## 11 REFERENCES

[1] Bolt, R.A., "Put-That-There": Voice and gesture at the graphics interface. *Computer Graphics*, 14(3 1980): 262-270.

[2] Carpenter, R., *The logic of typed feature structures*. Cambridge England: Cambridge University Press, 1992.

[3] Cohen, P.R., Cheyer, A., Wang, M., and Baeg, S.C. An open agent architecture, in the *Proceedings of the AAAI Spring Symposium,* 1994, AAAI Press. Reprinted in *Readings in Agents,* Huhns, M. and Singh, M. (eds.), Morgan Kaufmann Publishers, San Francisco, 1-8.

[4] Cohen, P.R., Johnston, M., McGee, D.R., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. QuickSet: multimodal interaction for distributed applications, in the *Proceedings of the International Multimedia Conference,* 1997, ACM Press, 31-40.

[5] Johnston, M. Unification-based multimodal parsing, in the *Proceedings of the International Joint Conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics,* 1998, ACL Press, 624-630.

[6] Koons, D.B., Sparrell, C.J., and Thorisson, K.R., Integrating simultaneous input from speech, gaze, and hand gestures, in *Intelligent Multimedia Interfaces*, M.T. Maybury, Editor. AAAI Press/MIT Press: Cambridge, MA, 1993, 257-276.

[7] Kumar, S., Cohen, P.R., and Levesque, H.J. The Adaptive Agent Architecture: Achieving Fault-Tolerance Using Persistent Broker Teams, in the *Proceedings of the International Conference on Multi-Agent Systems,* 2000.

[8] Mackay, W.E., Fayard, A.-L., Frobert, L., and Médini, L. Reinventing the familiar: Exploring an augmented reality design space for air traffic control, in the *Proceedings of the Conference on Human Factors in Computing Systems,* 1998, ACM Press, 558-565.

[9] McGee, D.R., Cohen, P.R., and Oviatt, S. Confirmation in multimodal systems, in the *Proceedings of the International Joint Conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics,* 1998, ACL Press, 823-829.

[10] McGee, D.R., Cohen, P.R., and Wu, L. Something from nothing: Augmenting a paper-based work practice with multimodal interaction, in the *Proceedings of the Conference on Designing Augmented Reality Environments,* 2000, ACM Press, 71-80.

[11] Moran, T.P., Saund, E., Melle, W.v., Bryll, R., Gujar, A.U., Fishkin, K.P., and Harrison, B.L. The ins and outs of collaborative walls: Demonstrating the Collaborage concept, in the *Proceedings of the Conference on Human Factors in Computing Systems,* 1999, ACM Press, CHI'99 Extended Abstracts, 192-193.

[12] Neal, J.G. and Shapiro, S.C., Intelligent multi-media interface technology, in *Intelligent User Interfaces*, J.W. Sullivan and S.W. Tyler, Editors. ACM Press, Frontier Series, Addison Wesley Publishing Co.: New York, New York, 1991, 45-68.

[13] Nigay, L. and Coutaz, J. A generic platform for addressing the multimodal challenge, in the *Proceedings of the Conference on Human Factors in Computing Systems,* 1995, ACM Press, 98-105.

[14] Oviatt, S.L., Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction*, 12(special issue on "Multimodal interfaces" 1997): 93-129.

[15] Smith, R.W. Integration of Domain Problem Solving with Natural Language Dialog: The Missing Axiom Theory, in the *Proceedings of the Innovative Applications of Artificial Intelligence Conference,* 1992, AAAI Press, 270-270.

[16] Want, R., Fishkin, K.P., Gujar, A., and Harrison, B.L. Bridging physical and virtual worlds with electronic tags, in the *Proceedings of the Conference on Human Factors in Computing Systems,* 1999, ACM Press, 370-377.

[17] Wu, L., Oviatt, S., and Cohen, P., Multimodal integration - A statistical view. *IEEE Transactions on Multimedia*, 1(4 1999): 334-341.