Design Patterns for Research Methods: Iterative Field Research

Kevin S. Pratt
kpratt@cs.tamu.edu
HR Bright 333
Texas A&M University
College Station, TX
77843

Abstract

For the last two decades the idea of design patterns has been a useful abstraction for computer scientists and programmers. As computer scientists, and scientists of all fields, are more than just programmers, we can apply the patterns concept to more than just program design. Indeed, the meta-creative processes and research methods which generate the code can also be viewed through the patterning abstraction to identify research method patterns and the contexts where they can be applied. One example of a research pattern is Iterative Research. Two examples of this Iterative Research method will be presented: the first investigating the vehicle, interface, and team CONOPS for small Unmanned Aerial Systems (sUAS) used during Urban Search And Rescue (USAR) operations, and the second working to develop a multi-operator team HRI metric and robot usability evaluation method.

Introduction

In 1977 the concept of design patterns was first introduced by Alexander et al. (1977) as a way to think about the architectural elements of buildings and other urban structures. A design pattern was initially described as a generalized, reusable design element which can be used to solve recurring problems. As with the individual patterns themselves the concept of design patterns is itself a reusable abstraction and has since been extended to computer science and program design (Gamma et al. 1994) and HCI (Tidwell 2005). Patterns can be applied to more than just design elements however: just as there are common challenges in design, there are recurring problems in the creative work which drives design. By considering the process of research as a set of contexts and matching patterns we, as a group of researchers, make it possible to build on not only each others results, but our processes as well.

As an illustration of research patterns two examples of the *Iterative Research* pattern will be presented. The similar to its sibling, Iterative Design, Iterative Research is a methodology that focuses on multiple, short cycles between the field, development, and lab testing. The first example details an ongoing research project into the vehicle, sensor, operator interface, and team Concept of Operations (CONOPS) for small, man-packable Unmanned Aerial System (UAS) used in Urban Search And Rescue (USAR) operations. The second discusses the development of a multi-operator team HRI metrics and robot usability evaluation method for analyzing the performance of operators and robots in USAR operations.

To support these examples **Design Patterns** will review the concept of design patterns as used in architecture and programming, then **Patterns in Research** will expand the concept of patterns to include the research process itself, following this **The *Iterative Research* Pattern** will describe the pattern while **Examples** will detail the two case studies, and finally **Discussion/Conclusion** will provide additional commentary and a supporting set of potential research patterns.

Design Patterns

In 1977 Design Patterns as an idea was first presented by Alexander et al. in the book *A Pattern Language* (1977). This initial description of the concept focused on the observation that many medieval cities had similar useful and aesthetically pleasing features, even though there had been little collaboration between the various architects and builders. Presented with similar needs for their cities the architects tended to respond in similar manners (this bears strong resemblance to the concept of parallel evolution used in evolutionary biology). Framed in a computer science metaphor, the pattern employed by the architects can be viewed as a function which generates a plan, be it the design of a door or the layout of the high street, based on the local factors (materials, available land), which serve as the inputs to the function. The implementation (*I*) is a function of the pattern (*P*) employed and the conditions, χ, of the specific instance.

$$I \approx P(\chi) \tag{1}$$

Speaking of computer science, it was not even the original application of design patterns to architecture, but the application to computer programming that they are best known for. Design patterns as a concept were popularized by Gamma et al. (also known as the Gang of Four) in *Design Patterns: Elements of Reusable Object-Oriented Software* (1994). The authors identify three categories of patterns, Creational, Structural, and Behavioral, and go on to provide several examples of each. Well known patterns include the

Factory, a creational pattern for generating run time specified objects, Singleton, a creational method which only permits one instance of a class to be invoked at a time, and the behavioral Iterator for accessing elements of arbitrary type from a collection (1994).

Patterns in Research

In addition to the individual patterns themselves, Gamma et al. also inadvertantly identified another pattern: the concept of design patterns itself. Just as they were able to translate the meta-pattern of design patterns from the old context of architecture to the new context of coding, the concept of patterns can be applied to the context of research and creative work.

To begin we will revisit the definition of patterns proposed by Gamma et al. (1994). In Section 1.1 of *Design Patterns* they present the following definition.

1. The **pattern name** is a handle we can use to describe a design problem, its solutions, and consequences in a word or two. Naming a pattern immediately increases our design vocabulary. It lets us design at a higher level of abstraction. Having a vocabulary for patterns lets us talk about them with our colleagues, in our documentation, and even to ourselves. It makes it easier to think about designs and to communicate them and their trade-offs to others. Finding good names has been one of the hardest parts of developing our catalog.
2. The **problem** describes when to apply the pattern. It explains the problems and its *context*[emphasis added]. It might describe specific design problems such as how to represent algorithms as objects. It might describe class of object structures that are symptomatic of an inflexible design. Sometimes the problem will include a list of conditions that must be met before it makes sense to apply the pattern.
3. The **solution** describes the elements that make up the design, their relationships, responsibilities, and collaborations. The solution doesn't describe a particular concrete desing or implementation, because a pattern is like a template that can be applied in many different situations. Instead, the pattern provides an abstract description of a design problem and how a general arrangement of elements (classes and objects in our case) solves it.
4. The **consequences** are the results and trade-offs of applying the pattern. Though consequences are often unvoiced when we describe design decisions, they are critical for evaluating desing alternatives and for understanding the costs and benefits of applying the pattern.

   The consequences for software often concern space and time trade-offs. They may address language and implementation issues as welll. Since reuse is often a factor in object-oriented design, the consequences of a pattern include its impact on a system's flexibility, extensibility, or portability. Listing these consequences explicitly helps you understnd and evaluate them (1994).

To show that the pattern concept is extensible to research patterns the second and third elements in this list: Context (problem) and the matching Pattern (solution) must apply to research methods as well. We begin by expanding on our definitions of Context and Pattern.

Context, in the design pattern sense, describes the condition or situation when a specific pattern should be applied. A context should be a prototypical description of a set of problems that is encountered repeatedly across a range of projects. Depending on the context in question there may also be a set of conditions that help identify the context (or potentially diferentiate between similar contexts). Using the Iterator pattern (1994) as an example, the context in this case would be that you have a collection of items and you need to step through the collection visiting each item once. This context could be applied to an array, a tree, a linked-list, or a hash table and can also be used across a range of data types in the collection: individual bits, float, char, string, or binary blobs. These specifics, collection type and data type, don't matter to the context, but instead are the inputs to the pattern to determine the specifics of the implementation.

And this pattern is of course the other critical element of the concept. For a pattern to exist there must be elements of the solution that are common across all instances of the context. This commonality often occurs in the functionality of the solution to the problem context. In our Iterator example the commonality is the need to step through all the items of a collection which, again, is independent of the collection or data type.

With the elements of the design pattern concept identified we can now evaluate if these elements are present in the process of conducting academic research. With the elements identified above the question becomes: Are there recurring problems in research, and are their reusable solutions to these problems?

Certainly each researcher could construct their own definition of the research process: for the sake of argument however, consider research as a derivative of the scientific method. The canonical definition of the scientific method consists of the following elements.

- Observation
- Hypothesis
- Experimentation
- Analysis
- Interpretation
- Evaluation

If we agree that in some sense conducting research is the process of implementing the scientific method, then we have a reasonable case for the following definition of research.

> *Research* is the process of generating a research question and hypothesis, devising and conducting an experiment, and evaluating the results to answer the original question and confirm or deny the hypothesis.

Visible in this very definition are the contexts necessary for the design patterns paradigm. Generating research questions is a context. Constructing experiments is a context and conducting experiments is a context. Processing data and evaluating results is also a context. These are only a few of the potential contexts: investigating each of these steps, or a particular field of research (such as HRI or field robotics) will undoubtedly reveal addition more precise contexts.

Indeed, our collective participation in this symposium, *Experimental Design for Real-World Systems* implies that we all collectively believe there to be recurring problem contexts and reusable solution patterns that we can share and benefit from. Beyond this *de facto* existence proof the best way to show that research patterns exist is to describe one such pattern, *Iterative Research*, which is presented in the following section.

The *Iterative Research* Pattern

Expanding on the definitions of research patterns presented in earlier sections, this section will describe the Iterative Research pattern. While this pattern undoubtedly can be succesfully applied to numerous contexts, it is presented here as a method to drive research question generation in AI, field-robotics, and man-machine interface research. The iterative research pattern consists of four primary steps: field observations, problem identification, technology development, and field testing. Each of these elements is detailed below. Figure 1 illustrates these four segments of the pattern and their relationship to each other. As the diagram shows, the first step in applying the pattern is to observe the target application, these observations are then used to help identify the research problem(s). A solution to the selected problem is then developed, and tested in the original domain where the problem was identified. These tests are then observed and the next iteration begins.

While most robotics researchers follow this advice as a matter of course, it is worth beginning the description of the Iterative Research pattern with an aside to restate another research recommendation.

> 'In any field, but especially robotics, have an real-world application.'(Scassellati 2008)

Even though your research interests may be machine learning, HRI, or mechatronics having a specific application to focus on, be it child autism therapy, stroke victim rehabilitation, or USAR robotics will help focus and guide the research. Having such an application provides an environment to initiate the observing segment of the pattern.

Beginning with this real-world application the first step of the Iterative Research pattern is field observations. These field observations are used to identify needs within the chosen task. As a research project moves through its life cycle the nature of field observations will of course change. When a project first begins however, these observations can be as basic as following the subject matter experts as they conduct their work. In USAR robotics this might be watching a search team work through a set of exercises, or for rehabilitation or therapy work this might be watching a set of doctor-patient sessions. Having first hand knowledge of the

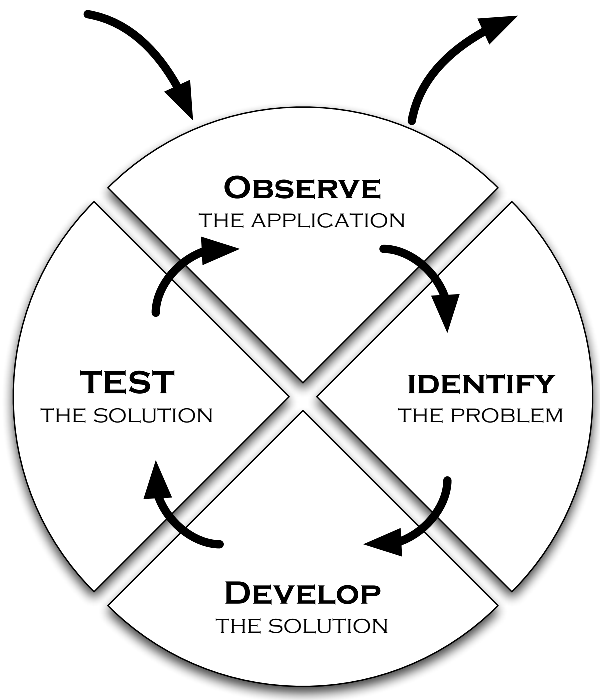

Figure 1: Diagram of the Iterative Research pattern. The four segments of the pattern: Observe, Identify, Develop, and Test are shown in their cyclic relationship.

task domain allows us as researchers and scientists to ask the appropriate questions and identify the most critical technical needs (and hence research areas) for those working in our domain. As individual researchers continue work in a specific field application and work on subsequent series of projects in the same domain, this initial set of observations can be considered a type of gate condition: the domain familiarity this generates may already exist as a result of previous work in the area.

The second step in the pattern is to use the field observations to identify the research problem. As a research project is iterated over this straightforward step can take on many different implementations. In the first iteration, this task may take the form of team debriefings and reviewing the notes from the initial observations. Involving the subject matter experts consulted during the field observations can serve as a verification of observations made by the research team. As is often the case, multiple potential research topics may present themselves and consultation can also help prioritize or refine these multiple topics. As a project moves into later iterations this step becomes focused on the tests and experiments conducted in the previous loop. The results of these tests can be analyzed for both success and their fulfillment of the issues or problems they were designed to address. If the experiments are successful a new problem (or sub-problem) can be identified for the next iteration, or if work remains to be done on the topic at hand another iteration can be taken to address the problem with a different approach.

With the specific research problem or question identified the next step is to actually do the development work. The nature of this step of course depends on the application and the type of research. It may be developing a new vision sensor and obstacle detection algorithm, building a new robotic arm and hand unit for a wheelchair, or prototyping a new interface for robot control. Given the selected field application and the engineering, as opposed to purely theoretical, heritage of robotics this step will nearly always involve some element of actual development. Early iterations through the cycle may use Wizard of Oz studies or other simulations to help explore a breadth of topics, but as a project matures through multiple iterations this step will encompass more active development.

The final step in the pattern after the development has been completed is to return to the original setting and test the new work. The intent is of course that these tests show the newly developed elements completely address the problem selected in this iteration. Just because what is intended and what actually happens are not always similar does not mean this step fails if testing shows the problem is not sufficiently resolved. Though it does not validate the particular solution, such additional testing and observation of the target application is equally useful in helping to refine the nature and scope of the problem in question as well as presenting new hypothesis for potential solutions. As these tests are being conducted they are of course being observed, thus beginning a new iteration by either refining the current problem or identifying new questions that can be pursued.

As the name implies, the intent is that the pattern be applied several times in a row, using multiple iterations of the loop to address a problem rather than a single set of Observe, Identify, Develop, Test. Indeed without multiple iterations, this degenerates into a monolithic waterfall development model.

Examples of *Iterative Research* in field-based robotics research

With an understanding of what the Iterative Research pattern is, this section presents two examples of how the pattern can be applied to actual research scenarios. The first research example discusses the development of the vehicle, interface, and team CONOPS for small Unmanned Aerial Systems (sUAS) used during Urban Search And Rescue (USAR) operations. It was primarily conducted using field observations of USAR and structural inspection of buildings damaged by Hurricane Katrina in 2005. The second example presents the creation of multi-operator team HRI metrics and robot usability evaluation methods. The focus of this work was on testing the developed metrics at robotic disaster response field exercises.

In this first line of research, the use of Small Unmanned Aerial System (sUAS) in USAR operations, field observation has been used to identify initial research areas for further study and to evaluate initial results in these areas. Additionally, it is anticipated that field work will be crucial in evaluating the final results of this work: only by closing the loop and returning to experiments similar to the original flight operations that helped generate the research questions will we be able to show these questions have been addressed. As described in Pratt (2007), Murphy, Pratt, and Burke (2008), and Murphy et al. (2006) this research began by studying extended deployments of teleoperated sUAS in the disater areas affected by Hurricanes Katrina and Wilma in 2005. As these were the first documented deployments of sUAS in civilian applications (for both USAR and structural inspection tasks), the primary purpose of this research was to establish a baseline for the CONOPS of these vehicles as well as identify reasearch areas that would improve the capability of the vehicles.

As described in (Pratt 2007), a typical operation began with the team arriving on-site with the damaged multi-story commercial structure. The team, which included several domain experts, then conducted an evaluation of the structure to decide the mission for the operation, which was followed by a safety breifing to account for the particular hazards of the site. One or more individual flights were then flown at the site to satisfy the goals of the mission. Once the flights were completed, the team conducted a mission debriefing to evaluate both team and vehicle performance during the mission. Additional team debriefings were conducted at the end of each day as well as at the conclusion of the expedition. It was these debriefings with our team of domain experts that served as the basis for the majority of the research findings. Given the dynamic and uncontrolled nature of the field research it would have proved exceedingly difficult to collect a statistically significant number of flights or enough quantitative performance data to properly evaluate the vehicle and the team. Post-operation debriefings were the only available method for analyzing performance during these flights.

This research exhibits several of the elements of the Iterative Research pattern described above. First, even though the members of the research team are AI, behavioral robotics, and HRI researchers there is a selected application and focus for the research; a focus which by its very nature requires field observations and demonstrations for successful execution. As with much observational research, this inherently means that more traditional lab based methods were unavailable, we were still able to collect valuable data: indeed, as an initial evaluation tool opinion and qualitative analysis by domain experts was a most productive research method. From this research we were able to draw four primary conclusions about sUAS operations in urban environments (one of these was that omni-directional obstacle avoidance was critical for vehicles in cluttered three-dimensional environments) (Pratt 2007) as well as identify required team member roles and the minimum team size required for successful operations. As this omni-directional obstacle avoidance technology is developed, a return to similar field experiments will be critical in evaluating the initial conclusions surrounding the usefulness of such capabilities.

Figure 2 shows both the vehicle used as well as the conditions encountered during this research. The figure shows the iSensys Imaging Platform 3 (IP3) flying into the Hard Rock casino to inspect where the casino barge had been pushed into the building by the storm (the grey object in the frame is the aft end of the barge, listing hard to port, where it has come to rest against the building). The frame also illustrates the hazard cluttered flight domains pose to the vehicle. Within 2m of the helicopter are: the roof, the floor, a trash compactor, two vertical steel beams, caution tape (additionally hazardous as a moving obstacle), and wires and ceiling tile supports hanging from the ceiling (also mobile).

Figure 2: The sUAS helicopter entering a damaged structure to inspect an I-beam for structural integrity.

A second line of inquiry similarly reliant on field work has been the development of robot and team performance metrics. As described by Burke et al. (2008), the 2007 NIST Rescue Robot Evaluation Exercise provided an opportunity for the initial field testing of a General Robot Usability Questionaire as well as the Human-Robot Team Effectiveness Incident Log and Ratings Scale. In contrast to the sUAS work above, this research operates at a more meta-investigative level: not asking how to build a better robot, but how to evaluate robots so that we can identify when a better robot has indeed been built. Rather than starting in the field, however, this research began as analytical work in the lab, and progressed to field research as an initial evaluation of the methodology to help refine the questionaire and ratings scale for future revision and use. This is not to say that observation and identification of the research question were not a part of the process. The PI for this research, Dr. Burke, has observed numerous other USAR exercises as an HRI researcher, allowing this project to benefit from earlier observations without an individual set of its own.

During this research the questionaire was provided to USAR professionals who were attending the NIST exercise and the rating scale and incident logs were used by the researchers to help monitor the operators as they moved through the exercises and robots presented. This research was intended to be a pilot study for all three of these measures, but as this was not the primary purpose of the exercise we were not able control the appropriate conditions to generate a successful pilot study. Which is not to say that we were not able to generate useful findings from this initial testing of these measures. While an insufficient number of responses were collected to qualify this as a full pilot study for these measures, the responses we did receive allowed us to capture several instances of multi-operator team coordination (both single and multiple robot variants) as well as identify initial conditions of the exercies (team coordination was implicitly and occassionally explicitly discouraged due to the setup of the experimental scenarios) which would need to be modified for succesful future research on multi-operator robot teams. As this research progresses field research will continue to play an important role in the progress of the project: as the tools are prepared for pilot tests, and eventually used as operational HRI team metrics, it will be critical to consistently evaluate the methodologies during field trials.

Figure 3 shows an example of the type of behavior that can be captured with the Human-Robot Team Effectiveness Incident Log and Rating Scale (notably the Incident Log which allows for notation and classification of such sporadic and ad-hoc behaviors). The operator of the second robot, wearing the backpack, uses his video display to point out to the two operators of the first robot a compartment that they missed in their search pattern. Only the first robot was involved in the scripted exercise, but they had failed to locate all the targets in the train, until the second operator/robot (not part of the exercise, just practicing operating the robot nearby), was able to provide the exproprioceptive viewpoint needed to locate the missed compartment and target.

Figure 3: Spontaneous multi-operator, multi-robot cooperation in searching a train car. This illustrates the type of behavior targeted by the Human-Robot Team Effectiveness Incident Log and Rating Scale

Discussion / Conclusion

The previous two examples of USAR sUAS and HRI team performance metric research illustrate two cases where a cyclic set of tasks begins to appear. Does this prove the existence, validity, or usefulness of the Iterative Research pattern? Does the iterative pattern confirm the concept of patterns as a useful abstraction for research methods overall? Of course not. Patterns are but one potential abstraction for discussing research methods and practices: as with all abstractions, it is only as useful as others find it applicable to their work.

There is, however, reason to believe that patterns can function as an abstraction for research methods. As we have seen from *A Pattern Language: Towns, Buildings, Construction* (1977) and *Design Patterns: Elements of Reusable Object-Oriented Code* (1994) patterning itself is a functional abstraction, and can be applied across multiple fields. Indeed, in a recent article (2009a) and at a MacWorld 2009 PULSE presentation (2009b) Merlin Mann examined the potential of 'Design Patterns for Creative Work'. Using the Symposium as an opportunity to examine experiment design and research methods across multiple academic disciplines, we can look to find other research patterns and investigate the validity of patterning as an abstraction of research and experimental methodologies.

As an initial exploration of patterns, several potential research patterns are presented below. The definition of a pattern established by Alexander (1977) is used to present the *Dichotomous Pairing*, *Increasing Fidelity*, and *Divide and Conquer* patterns.

Dichotomous Pairing

Problem The problem for research in real-world systems is that research on these systems must show that the work is both theoretically sound as well as practically functional. As researchers we must prove to our peers that both our engineering and algorithms have merit.

Solution The solution (pattern) for this dilema is of course to show proofs of our system on both of these fronts. Research that effectively employs this pattern provides both a theoretical validation, typically by proof or simulation, and an engineering validation, which often takes the form of field trials or demonstrations of the system. Much of robotics research, particularly systems research, lies at the curious confluence of theoretical and engineering derived scientific discplines. Though a rather routine, banal pattern its commonality throughout the literature does speak somewhat to the usefulness of the pattern.

Consequences Pursuing a research problem on both fronts will of course require increased resources. For some research problems the majority of the useful results may be generated in one of these steps and not the other. As resources are of course limited it is important for researchers to verify that this dual approach will yield the results necessary before employing the pattern.

Increasing Fidelity

Problem Systems research projects can be expansive, multi-faceted projects that require an extensive amount of net work to complete. Such a large amount of work and time that in many cases it would be prohibitive to invest that amount of time before generating or publishing results.

Solution Instead of tackling large projects in one pass they can be broken down into several segments which yield an incremental refinement towards the result. Initial portions of the project use smaller, lower-fidelity systems to reduce large risk elements and provide preliminary results. Later segments of the project then use the early results to focus the development of the full, high-fidelity system on methods that have been initially vetted with the low-fidelity systems. By creating multiple versions of incrementally increasing fidelity time and resources can be focused only on the most promising variants of a system.

Consequences As with any simulation, researchers must carefully vet their assumptions used in the low fidelity systems. If the assumptions underlying the initial systems are fallacious there can be little hope for accurate results in later work.

Divide and Conquer

Problem Mirroring the problem of the *Increasing Fidelity* pattern, the fundamental issue for *Divide and Conquer* is the size and complexity of systems research projects. Systems research is exemplified by multi-disciplinary research requiring work across numerous subjects and specialties.

Solution Complementary to the horizontal seperation of *Increasing Fidelity* parceling the problem into multiple layers, the boundaries can instead be vertical, decomposing the problem into its constituent research areas. This may seem antithetical to the canonical systems approach of combining disciplines, but it is not strict decomposition, breaking the system into its components, but instead chosing to focus current work only on an individual research area, gradually combining results as they are completed. Indeed, this could be combined with *Increasing Fidelity* to intensifying the realism of one element at a time.

Consequences As with any variable isolation type system, decomposing a system in this fashion will destroy the interaction between variables. Researchers must be precise in selecting their variable boundaries, or at a minimum, be cognizant of the impacts it will have comparred to a more holistic system approach.

Summary

This paper presents the concept of Research Patterns as an abstraction for organizing and discussing research methods, notably for systems and field based research. The concept of research patterns, derived from design patterns in software development, holds that there is a set of common, recurring problems in research and that while the details may differ, there is a set of common solutions which can be used to address these problems. As an example, the Iterative Research pattern was presented, along with two real-world uses of the pattern. The pattern states that field observations of the targeted application can be used to generate research questions, and then these questions can be addressed by rapid cycling through development, field testing, and beginning again with field observations. Other potential examples of research patterns were also discussed, notably *Dichotomous Pairing*, *Increasing Fidelity*, and *Divide and Conquer*.

References

Alexander, C.; Ishikawa, S.; Silverstein, M.; Jacobson, M.; Fiksdahl-King, I.; and Angel, S. 1977. *A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series)*. New York: Oxford University Press, USA.

Burke, J.; Pratt, K. S.; Murphy, R. R.; Lineberry, M.; Taing, M.; and Day, B. 2008. Toward developing hri metrics for teams: Pilot testing in the field. In *HRI 2008: Metrics Workshop*.

Gamma, E.; Helm, R.; Johnson, R.; and Vlissides, J. M. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software (Addison-Wesley Professional Computing Series)*. Addison-Wesley Professional.

Mann, M. 2009a. The problem with "feeling creative". http://www.43folders.com/2009/01/02/feeling-creative.

Mann, M. 2009b. Towards patterns for creativity. http://clips.43folders.com/post/69662786/merlins-macworld-slides, Macworld PULSE.

Murphy, R. R.; Griffin, C.; Stover, S.; and Pratt, K. S. 2006. Use of micro air vehicles at hurricane katrina. In *Proceedings of IEEE Workshop on Safety Security Rescue Robots*.

Murphy, R. R.; Pratt, K. S.; and Burke, J. 2008. Crew roles and operational protocols for rotorary-wing micro-uavs in close urban environments. In *Human-Robot Interaction*.

Pratt, K. S. 2007. Analysis of vtol mav use during rescue and recovery operations following hurricane katrina. Master's thesis, University of South Florida, 4202 E. Fowler Ave. Tampa, FL 33620.

Scassellati, B. M. 2008. Social robots and human social development. In *HRI Pionners Keynote Address*.

Tidwell, J. 2005. *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media, Inc.