The research topic of *model based systems engineering* is emerging as a rich field for exploration. As an application, systems built out of individual components are interesting to commercial and defense industries. As a discipline, formal modeling is coming into its own, becoming more of a science and less of an art. My experience with modeling and my recent role as an application developer have given me an interesting perspective for research in my career, and I believe that my research should focus on these two major areas:

* **Autonomous systems.** Many systems today can be bought as autonomous off-the-shelf. However, autonomy should not be confused with intelligence, although it frequently is. Intelligence is the ability to adapt/learn, and also to react to stimuli that would otherwise require human intervention; whether or not that reaction is *intelligent* is a function of how the system was engineered.

* **Modeling and metamodeling.** Reliable and commonsense abstractions for programming and control are imperative to building dependable and robust systems and software. Exploration into how this may be done by experts, and passed on to designers, will greatly improve design times and testing confidence.

My anticipated role as a researcher is to use my deep experience with application development, and the "bricks and mortar" of modeling to perform research in the area of how to build intelligent, autonomous systems.

**Research Agenda**

5 YEARS
In this time, I hope to have created several feasibility examples of complex systems which were built using alternative means to hand coding (i.e., through modeling and metamodeling). This will allow me to show the abilities of an alternative to the current paradigm of hard real-time languages for embedded systems (e.g., C++). In addition, I expect to continue to display how to add intelligence to off-the-shelf systems with short timelines by using generative methods (either through models or old-fashioned scripts) which examine the semantics of the domain to determine what the semantics of composed domains should be.

20 YEARS
By this time, the field of engineering will be drastically different. I believe (as I lay out in my teaching statement) that the continued dominance of US universities in engineering education will not be in technical superiority, but in true *engineering* superiority—that is, in the ability to make things work when given a set of components and a goal. My intuition is that straightforward applications will be "farmed out" in the future, meaning that what will make US institutions stand out is their ability to implement complex and highly unintuitive integrations to provide dependable behaviors of systems. In order to do this in a confident way, the paradigm for embedded system design and implementation must be fundamentally changed, and I aim to be a leader in this area by proposing that domain-specific and semantically integrated models and simulations are the way in which this can be achieved.
Autonomous Systems

Autonomous systems—that is, systems that make decisions without an operator—are becoming more common in off-the-shelf as well as custom-built applications and components. Depending on the domain of application, response requirements, and liveliness of the system, the criteria for building intelligent autonomous systems can vary greatly. As such, it is not sufficient to have experience with just control, or just programming, but a wide experience is necessary in design and development of embedded and intelligent systems, as well as experience in how to create such systems in a robust manner.

Intelligent systems should be predictable, meaning that they are expected to react and behave in an anticipated (if not provably correct) manner. In order to deploy such systems research must continue to be explored in the application of robust control, as well as sub-optimal control, in order to find intersection points for certain classes of systems.

Embedded Systems

The vast majority of processors operating in the world today are running in so-called embedded systems. These systems are often software-driven, but hard-coded in that they are expected to behave correctly from the moment they are deployed—that is, there is no automatic update feature for such systems to correct design flaws or to account for bugs discovered. As such, there is a similarity between autonomous and embedded systems that is this common design goal: to turn it loose and have confidence it will run correctly.

Abstractions for building embedded and concurrent systems are woefully out of date, and unfortunately stem more from operating systems (where a user is frequently on hand) than from aeronautical and astronomical systems, where common fixes such as patches and updates are not considered feasible. I am interested in the feasibility of drastically modifying how embedded systems are built, through the same system engineering ideas I plan to apply to intelligent autonomous systems. Most of these concepts stem from my deep experience with domain-specific modeling and metamodeling.

Modeling and Metamodeling

Formal modeling, especially in a domain-specific environment familiar to experts that class of system, is establishing itself as its own science. That discipline, known as Model-Integrated Computing focuses on the development of an Integrated Development Environment (IDE) that is customized for particular domains. This gives the building blocks for the development of intelligence by system experts, rather than by programmers who must first understand the domain and then encode the system. The latter approach frequently results in misunderstandings and miscommunications which may be realized dangerously late in the development process.

Formal modeling can be examined in the abstract and shown to work for all systems that can be built. In this way, it satisfies that requirement of confidence in a final deployed system. Furthermore, the design and implementation of a formal modeling language is often an order of magnitude quicker than traditional programming methods—with reduced time for testing and verification.

Applicable Research Experience

Several application and theory-based experiences show how I am uniquely positioned to enter into this research area with confidence of success.
Software Enabled Control (SEC), DARPA
I joined the SEC project during its final year, and participated in the Capstone Demonstration by participating in the design and implementation of two experiments in the demonstration, each of which was flown on a live aircraft in June 2004 at Edwards AFB.

* Pursuit/evasion of fixed-wing aircraft. This demonstration involved the development of a model predictive controller that tracked a final waypoint, as well as avoided a moving pursuit aircraft through the same controller. The timescale of this project was about 6 months.

* Guarantee of safe UAV landing. This demonstration involved the calculation of safe sets which described the regions of the UAV statespace (3-D position, 3-D attitude, and velocity) from which a UAV could safely land on a runway. We used level-set methods as our model of computation for the reachable sets, and generated them offline using a toolbox. Once we had the level sets, we transformed them into an executable which ran at runtime to give real-time decisions of safety.

Domain-Specific Modeling and Metamodeling
As a graduate student, I designed a forerunner of the current metamodeling language that is used in the Generic Modeling Environment (GME), which is freely available from Vanderbilt University. While there, the metamodeling language was used to design hundreds of domain-specific modeling environments.

During my postdoc, and also during my graduate studies, I taught or team-taught a graduate course on this topic, and am in the midst of co-writing a textbook that describes how this process can be used, both technically and abstractly, to build systems in the language of the domain.

Model-Based Integration of Embedded Systems (MOBIES), DARPA
I participated in MoBIES for several years of my graduate career. My main contribution to the project was the technical development of the hybrid systems interchange format (HSIF), which was a first generation interface between tools in a hybrid systems toolchain. My involvement consisted of maintaining the software release, writing at least 3 of the tool-to-tool translators available, and coordinating between other MoBIES teams who were participating in the toolchain. The next generation of HSIF (hyper) was inspired by this work.

hyper HYBRID SYSTEMS MODELING FRAMEWORK
My involvement with HSIF led me to make several key observations regarding why it failed to become a mature toolbox. Among these were the fact that it was structurally flat (i.e., hierarchical state machines were not allowed), and it’s centralization on tool integration, rather than capability integration; e.g., the toolbox should have integrated simulation and verification, rather than just HyVisual and Checkmate. I took these lessons learnt and am in the process of applying them to the hyper framework, which is aimed at this concept of capability integration.

Research Applications
Research applications in autonomous systems are diverse and interesting to commercial, homeland security, and defense applications. I describe below some immediate goals I have for finding funding groundbreaking research in how model-based system engineering can help to solve these problems.

Healthcare at Home
An emerging—yet extremely important—research application is the use of intelligent systems for care of an aging population. These application fields are rich with possibilities, and ready for a design
process, rather than \textit{ad hoc} productions by competent experts who may be inexperienced in software development for control and autonomy. Example systems include,

* Intelligent embedded sensor platforms, to detect falling, inactivity, or noise
* Integrated homes, to utilize embedded sensors and systems in the infrastructure

\textbf{Homeland Security}

The manpower currently available to perform homeland security tasks is well below that which is required to have a high confidence that illegal activity will be detected. One possible solution to this is the development of intelligent autonomous systems which can reduce the amount of human interaction required. In addition, existing sensors and systems platforms will need to be integrated with new systems with high confidence in the final deployed system.

These systems, when developed, will likely involve aerial, underwater, and stationary platforms and will include vision, sound, and other axes of sensors, all of which will need to be interpreted according to high-level algorithms.