Project Summary

Software has become an extremely important part of today's technology driven world. Although necessary, thorough qualitative and quantitative evaluation of software is a challenging and expensive task. Once developed, evaluation solutions (typically test suites and benchmarks) are generally inflexible, representing only a sample of the application space, and are difficult to maintain as applications evolve. The nature of static evaluation methods is that they can not keep up with application innovation, usually designed only as an afterthought. The objective of this proposal is to address software evaluation issues by creating a dynamic strategy based on *artificial markets* of computational challenges. Fundamental to the design of artificial markets is the concept of a supported (or unsupported) constructive belief. Building on ideas from interactive proofs and program checking, dynamic market evaluation fosters innovation through frequent feedback, assisting in the discovery and testing of new algorithms and the production of high quality software — ultimately increasing the overall algorithmic knowledge of a given domain.

Given a concise description of a computational problem domain, X, it will be possible to create an *artificial market* around X, $AM(X)$. Software responsible for the creation of the market will generate a starting *agent* and a trust-worthy *market administrator*. Teams and individuals competing within the problem domain will improve their starting agent by participating in regular, automatic, administrator run competitions. Throughout the course of a competition, each agent offers and solves computational *challenges*, and is rewarded for both offering hard problems in X, and solving other agent's challenges effectively. Agents are ranked based on their performance within competitions and the corresponding teams use the log of a competition to test their agents, discover hard problems for their agent to offer, and gather ideas to improve their algorithms.

Challenges in the market represent algorithmic beliefs involving an agent's ability to provide problems that other agents will find difficult to solve. Similar to an interactive proof session, the belief's structure defines a protocol for agents to exchange problems and solutions. An agent's profit or loss from a given challenge provides a measure of support for or against the represented beliefs, and in turn demonstrates the quality of an agent's algorithms. The market is designed such that agents with the best algorithms make the most profit.

Intellectual Merit: The main contributions of this project are (1) to study the design of artificial markets of computational challenges with specific objectives; (2) show that artificial markets of computational challenges are a useful dynamic benchmarking tool for algorithms and software; and (3) show that they drive innovation, helping to both find better algorithms and improve understanding of their development. The overall goal is to help program officers, managers, and professors with the evaluation of research prototypes, competing software packages, and student programs, respectively, and to help designers improve their algorithms using feedback from an artificial market. The properties of the underlying constructive belief system will be well understood, and are used in the design of particular artificial markets to drive algorithmic innovation in specific areas by providing targeted feedback. The development of complex agents, administrators, and game generators will be supported by a number of generic programming tools, libraries, and design patterns that push the state-of-the-art. The resulting languages and methodologies are independent of the problem domain and contribute to programmer productivity and software quality.

Broader Impacts: The impact of this project will be through advancing discovery and understanding of how to solve computational problems and providing an exciting platform for interdisciplinary teaching and learning. A generic, parametrizable artificial market supports the improvement of algorithmic techniques for a wide variety of problems and an experiential teaching curriculum. Students become engrossed in learning the art of software development, algorithmic analysis, and modeling, through the refinement of their agent and frequent competitions. Distributed competitions will be run over the Internet allowing agents from different schools, companies, and even countries to compete within a common computational domain. The resulting algorithms, improved software, and useful tools, libraries and techniques will benefit software developers, scientific communities, and the general public as they make their way into various products and services.