Comparing Apples with Oranges: Evaluating Twelve Paradigms of Agency

Linus J. Luotsinen, Joakim N. Ekblad, T. Ryan Fitz-Gibbon, Charles Andrew Houchin, Justin Logan Key, Majid Ali Khan, Jin Lyu, Johann Nguyen, Rex R. Oleson II, Gary Stein, Scott A. Vander Weide, Viet Trinh, and Ladislau Bölöni

School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL

Abstract. We report on a study in which twelve different paradigms were used to implement agents acting in an environment which borrows elements from artificial life and multi-player strategy games. In choosing the paradigms we strived to maintain a balance between high level, logic based approaches to low level, physics oriented models; between imperative programming, declarative approaches and "learning from basics" as well as between anthropomorphic or biologically inspired models on one hand and pragmatic, performance oriented approaches on the other.

Instead of strictly numerical comparisons (which can be applied to certain pairs of paradigms, but might be meaningless for others), we had chosen to view each paradigm as a methodology, and compare the design, development and debugging process of implementing the agents in the given paradigm.

We found that software engineering techniques could be easily applied to some approaches, while they appeared basically meaningless for other ones. The performance of some agents were easy to predict from the start of the development, for other ones, impossible. The effort required to achieve certain functionality varied widely between the different paradigms. Although far from providing a definitive verdict on the benefits of the different paradigms, our study provided a good insight into what type of conceptual, technical or organizational problems would a development team face depending on their choice of agent paradigm.

1 Introduction

Researchers have designed a bewildering variety of paradigms for the control of agents. Even if we restrict our inquiry to the case of embodied agents, that is, artifacts which operate either in the physical world or a simulation of it, virtually every paradigm of artificial intelligence, software engineering or control theory was deployed with more or less success. However, wide ranging comparisons of agent paradigms are rare. When new methods and paradigms are introduced, they are compared with only several, closely related approaches which

R.H. Bordini et al. (Eds.): ProMAS 2006, LNAI 4411, pp. 93–112, 2007.

are considered direct competitors of the proposed paradigm. Making or revisiting comparisons between paradigms is a controversial, difficult and hard-to-sell work. One might argue that a researcher might better spend his or her time in designing new paradigms or improving existing ones instead of comparing, say, swarm algorithms with affective computing in the design of embodied agents. There might be people offended by the results, with reasonable claims that the methodology was incorrect, the implementation of the paradigm substandard, or simply, the measured quantity is not relevant to the given paradigm.

The fundamental question, of course, is whether if any of these comparisons make sense. We argue that **if both paradigms A and B can be used in the implementation of the same requirements, then these two paradigms can (and indeed, should be) compared**. That is not to say that the comparison is easy or that it can be reduced to a single numerical "score". Different paradigms have different strengths and weaknesses, and the goal of a comparison study is to shed light on these differences. Although we do not expect definite answers on questions like "which paradigm would eventually lead to an agent passing the Turing test", we can provide insight into lesser but still important questions such as:

- Would the implementation provide adequate performance?
- Can a rigorous software engineering process be applied to the development?
- Can the performance be predicted?
- Can human expertise in the problem domain be transferred to the agent?
- What will the development effort be?
- Will the resulting agent be predictable in its actions?

The remainder of this paper is organized as follows. In Section 2 we present the Feed-Fight-Multiply game, our control problem. We succinctly describe the twelve agents we implemented in Section 3. In Section 4 implementation effort and complexity measurements are presented for each agent. We detail our findings in Section 5.

2 The Feed-Fight-Multiply World

To study the benefits and drawbacks of various agent paradigms, we decided to place them in a virtual environment in which many of the real world challenges are reflected. We did not choose one of the existing environments, because the existing implementations would have skewed the result of the comparison. One requirement towards the environment was the existence of *multiple paths to success*. We expected that agents implemented in various paradigms will have a different external behavior as well. By measuring success as the conformance to a predefined behavior we would have favored some paradigms and disadvantaged others. In addition, having multiple paths to success is a quality of most natural environments and many artificial ones [6].

Upon these considerations, we implemented the Feed-Fight-Multiply (or Mate) game, which borrows elements from turn-based multi-player strategy games and