Spacecraft Reliability and Spatial Navigation

Cyrus F. Nourani
Project_METAAI

1. Introduction

The Double Vision AI computing paradigm's applications to autonomous reliable spacecraft navigation is presented. The autonomous control area is treated with multiagent control, intelligent planning, fault tolerance, and adjustable autonomy. Designs with object co-object pairs is from double vision Computing (Nourani 1995). Programming linguistics concepts for intelligent languages and objects from our papers are applid to present the basic technical definition for computing with object co-object pairs which defines the vision computing syntax and objects. The techniques for abstract implementations by computing agents are further presented for implementing visual computing by AI agents that cooperate n boards for problem solving**.** Eversince writing (Nourani 1995b) there is a new design technique put forth that is an optimistic problem solving paradigm. The paradigm is to design as if nothing could go wrong, then account for all faulty behavior, recoverable activities, and exceptional behavior. It is all taken stock by defining preconditions on co-objects, and agents defined on the co-object that can make intelligent recovery from unusual behavior. This techniques promises to make it easier to design AI systems for complex problems with many objects and thousands of rules. It is quite difficult to have general problem solving paradigms that can 'get to the point' when there are so many exceptions and unusual things that are the last thing on the mind' for a particular problem solving technique Nilsson (1971). The techniques noted in, e.g., Nourani (1999a) allow us to accomplish this in a systematic way. The way design is viewed by our approach is that there are objects actions and relations defining functionality. A system is defined by many communicating pairs of multiagent modules. A preview to a new model-based design for automatic onboard spacecraft visual field prediction based on Morph Gentzen computing is stated.

Keywords: Autonomous Agent Control**,** Double Vision Computing, Multiagent Reliable Computing, Spacecraft Automatic Onboard Vision, Robot Supervision , Multiagent AI Techniques, Visual Field Prediction, Spatial Navigation

2. Autonomy Control

The fastabstract presents multiagent controlled autonomous flight systems where structure for full autonomy, e.g., (AA99 Autnomny Control Wkshp, Seattle, May 99) is designed with specific mathematical

principles. Sensing and fault tolerance is modeled. Planing is carried out with agents trees on multiagent and/or trees. Goal directed behavior for the autonomous controller can be planned. The system is designed to functions with varying degree of independence with intelligent agents. Following the agent models (Genesereth&Nilsson 1987) a agent has an internal state set I, which the agent can distinguish its membership. The agent can transit from each internal state to another in a single step.

3. The Multiagent Multi-board Visual Field

The double vision computing paradigm with objects and agents might be depicted by the following figure. For computer vision, the duality has obvious anthropomorphic parallels. The object co-object pairs and agents solve problems on boards by cooperating agents from the pair without splurges across the pairs. Spatial navigation is stated in brief with the author's Morph Getnzen computing in for example (Nourani 1999b). Our computing by agents and multiboards apply cooperative problem solving techniques on the individual boards, but with generalized paradigms encompassing multiagents cross the boards. (see figure 1).

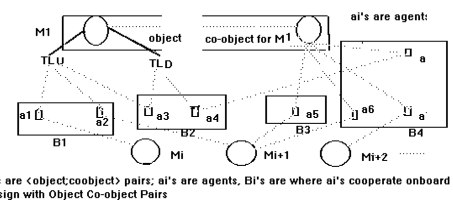

Fig 1 The Multiagent Multi-board Visual Field

4. Multiagent Robot Reliability

The term "agent" has been recently applied to refer to AI constructs that enable computation on behalf of an AI activity Genesereth and Nilsson (1987). It also refers to computations that take place in an autonomous and continuous fashion, while considered a high-level activity, in the sense that its definition is software/hardware, thus implementation, independent, e.g. (Nourani 1995b,98a,99a,). For example, in planning for space exploration, an agent might be assigned by a designed flight system to compute the next docking time and location, with a known orbiting space craft. Robot reliability is an area of crucial importance to which multiagent AI techniques might be applied to solve real problems encountered in fields such as intelligent systems, aerospace, robotics, etc. Each stage has to be approached in ways that define supervision and recovery from faults.

The recovery problems had been studied and are quite important for robot design (Gini's (1983) and (Nourani 1995b). However, since the multiagent AI techniques are recent the present fastabstract on MARS- Multiagent Robot Supervision, is perhaps the first to formally address robot fault and recovery with multiagent AI. System definition is by independent concurrent computing agents. The fastabstract defines FTS by a pair of systems, each consisting of many computing agents. The two systems are mutually synchronized to enable fault and exception handling and recovery in an automatic manner. The fastabstract also presents AI techniques for designing automatic visual flight control and spatial navigation as FTS.

5. The Autopilot Vision

In this section we take the reader through the design stages of a typical autonomous robot flight system for real applications involving many modules with independent functionality. The method of design and implementation by multiagent AI techniques is illustrated by a flight control example. A typical complex problem domain is the design of autonomous space flight system with supervisory functions. The expert is to come up with a set of functions, each corresponding to a space flight module, and to define how the modules interact and function together, by defining some operations amongst the modules. The modules are each complex hardware-software systems, best thought of as a microprocessor with its own running software, that implements the functionality of the module as extrapolated from the design process. Let us view one sample set of modules: M1-M8, each representing a functionality Fi, respectively. The modules are functions defined applying the visual field - figure 1, section 2. M1: Thrust Control; M2: Stage Control; M3: Orbit Selection; M4 : Attitude Control; M5: Flight Deck Control; M6: Sensors; M7: Obstacle Avoidance; M8 Communications; M9: Docking Functions. We present the implementation of one of the modules, M1, for the reader, the function F1 that is to implement M1. To define the thrust control there are a number of parameters that come to play that are hardware implied data or functionality related requirements. These are to be specified and then implemented by AI agents. Each function defined on the object corresponding to M1 is specified with a set of preconditions that imply co-objects to be defined for exception and recovery if the preconditions fail. Similar set of agents are implementing for acceleration, attitude and tilt levels, trajectory violation, and obstacle encounters. In each case the co-object functions and their implementing agents try to compensate such that the precondition to an operation on the object is met. In the figure below objects are represented as <object,co-object pairs>, where the co-object is a copy of the object on which supervision, faults and recovery functions are defined. The functions are implemented by agents ai, where ai are agents implementing velocity, acceleration, obstacle avoidance, or agents that check for limitations of the functions. A set of function symbols in the language, referred to by AF, is the set modeled in the computing world by AI Agents with across and/or over board capability.

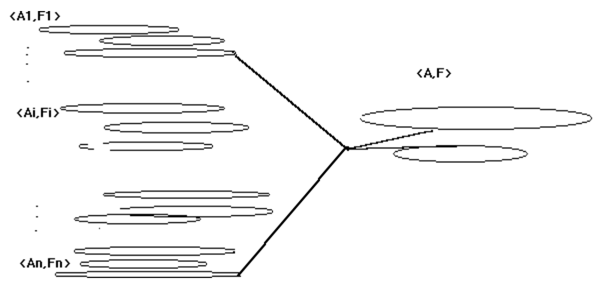

Figure 2 Designs with Intelligent Spatial Modules

Project_METAAI@CompuServe.com
Affiliations METAAI and USA Academic UCSB

References

Genesereth, M. and Nilsson. N.J. (1987). *Logical Foundations of Artificial Intelligence*, Morgan-Kaufmann,1987.

Nourani, C.F. (1999a). Nourani,C.F.,"Multiagent AI implementations an emerging software engineering trend," Engineering Applications of AI, 12(1999) 37-42, Elsevier.

Nourani,C.F.(1995a),"Double Vision Computing,"IAS-4, Intelligent Autonomous Systems,Karlsruhe, April 1995, Germany

Nourani,C.F.(1995b),"Multiagent Robot Supervision," Learning Robots, Heraklion, April 1995.

Gini and Gini(1983) Gini, M and G. Gini, "Towards Automatic Recovery in Robot Programs," Proc. 8th IJCAI, Vol.2, 1983, pages 821-823, Karlsruhe, Germany.

Nourani, C.F. (1998a), Desinging Fault Tolerant Systems with Multiagent AI, FTCS98, Munich, Germany, June 1998.

Nourani, C.F. 1998b, Reliability Engineering With Software Agents, ISSRE, Paderborn, Germany, 1998.

Nourani,C.F. 1999b Autonomous Multiagent Double Vision Spacecrafts, AA99, See papers at the Autonomy Conrol Software Wkshp Track, Seatlle, WA.