

UNCLASSIFIED

Defense Technical Information Center
Compilation Part Notice

ADP010521

TITLE: Emergent Aerospace Designs Using
Negotiating Autonomous Agents

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: Aerodynamic Design and Optimisation of
Flight Vehicles in a Concurrent
Multi-Disciplinary Environment [la Conception et
l'optimisation aerodynamiques des vehicules
aeriens dans un environnement pluridisciplinaire
et simultane]

To order the complete compilation report, use: ADA388284

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, ect. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:

ADP010499 thru ADP010530

UNCLASSIFIED

Emergent Aerospace Designs Using Negotiating Autonomous Agents

Abhijit Deshmukh, Timothy Middelkoop
University of Massachusetts, Amherst
 MIE Department
 Elab Building, Box 32210
 Amherst, MA 01003-2210, United States
 Anjaneyulu Krothapalli
 Dept. of Aerospace Engineering
Florida State University
 2525 Pottsdamme Street, Room 229
 Tallahassee, FL 32310-6046, United States
 Charles Smith
NASA Marshall Space Flight Center
 Bldg 4203 – Room 6145
 Mail Code TD03
 Alabama 35812, United States

Abstract

This paper presents a distributed design methodology where designs emerge as a result of the negotiations between different stake holders in the process, such as cost, performance, reliability, etc. The proposed methodology uses autonomous agents to represent design decision makers. Each agent influences specific design parameters in order to maximize their utility. Since the design parameters depend on the aggregate demand of all the agents in the system, design agents need to negotiate with others in the market economy in order to reach an acceptable utility value. This paper addresses several interesting research issues related to distributed design architectures. First, we present a flexible framework which facilitates decomposition of the design problem. Second, we present overview of a market mechanism for generating acceptable design configurations. Finally, we integrate learning mechanisms in the design process to reduce the computational overhead.

1 Introduction

Design of highly engineered aerospace components involves a variety of tradeoffs, such as cost versus quality, strength versus speed and weight versus stiffness. The need for efficient design methodologies, which capture these tradeoffs, is driven by the requirement to introduce cost effective and reliable products into the market quickly. This is especially true in the aerospace industry, where shrinking defense budgets and international competition are forcing manufacturers to significantly reduce the concept-to-delivery time for new products. In the space vehicle industry the challenges are even greater due to the need to produce extremely reliable designs with limited full scale testing capabilities. In order to achieve these objectives integrated synthesis environments, which incorporate high fidelity analysis, experimental testing, manufacturing and cost information at the design stage, are needed. Such systems enhance the probability of creating high quality products, without cycling through the product redesign iterations [1]. Most of these systems use multi-disciplinary design optimization methods to capture performance tradeoffs and prescribe a configuration with optimal design parameters. However, the complexity of several competing design objectives and diverse nature of the analysis tools makes the design optimization process computationally expensive. As a result,

the design cycle is either extremely long or the designers have to rely on approximations.

The use of multi-disciplinary knowledge at the product development stage presents opportunities for improving the product cost to performance ratios. However, this desired capability, along with the changing customer requirements, places a heavy burden on the design decision makers who must constantly re-engineer products to comply with the customer needs. In this environment it is critical to have a set of integrated tools that can efficiently handle the recurrent design cycles and at the same time manage the complex relations between the different design criteria.

This paper presents a Multi-Agent Design Architecture (MADA) which incorporates both the needs by providing a computational environment suitable for evaluating numerous design scenarios and a distributed solution search method that can handle multiple design criteria. The distributed search methodology uses negotiations between different stake holders, or design decision makers, to guide the search in the design parameter space. An important aspect of this method is the absence of a centralized design optimization module.

This paper is organized as follows. Section 2 gives an overview of the agent based design architecture. We use a high speed civil transport (HSCT) nozzle as an example throughout the discussion. Section 3 details the distributed design search methodology used in MADA. Section 4 discusses learning techniques that can be used to enhance the system performance. Finally, we present a summary of the developments and plans for future research.

2 Multi-Agent Design

Agents provide an efficient architecture for distributed design systems by encapsulating both data and process intelligence into autonomous decision making packages. MADA, shown in Figure 1, is a community of agents which represents a collection of tools, knowledge and procedures required for collaborative design and analysis tasks. We define the portion of the MADA system which transforms input parameters into a complete design as the *design ecology*. While the design ecology can produce a single design instance, the market based decision framework, which we term the *design economy*, searches the design space for a configuration that is acceptable to along all the decision criteria.

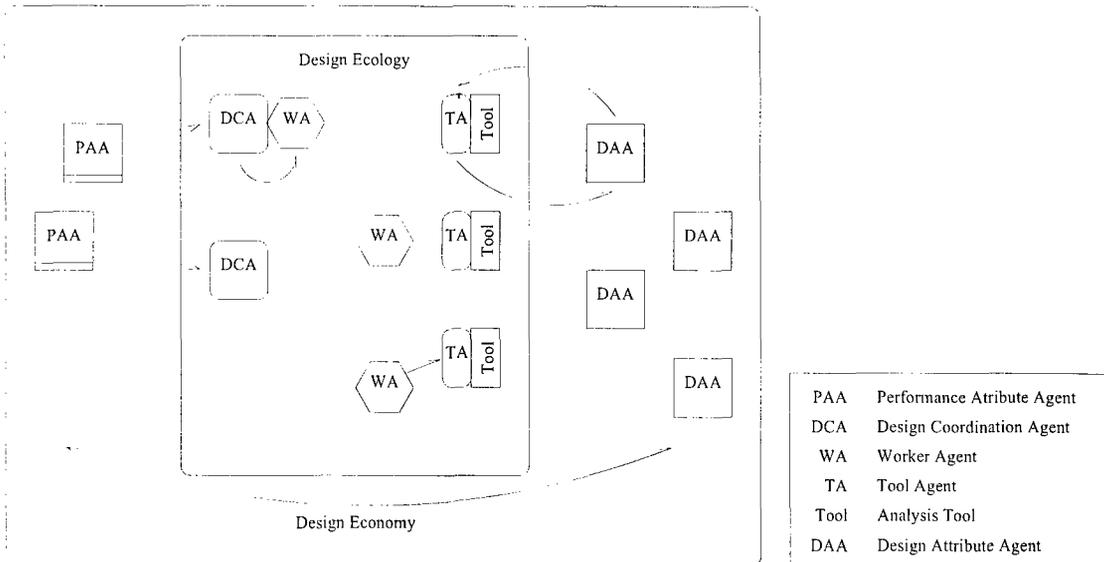


Figure 1: Multi-agent design architecture

It is important to note that the following discussion pertains to parametric design problems. We do not address the issues related to conceptual design in this paper. We consider a design scenario to be a collection of parameters and decisions. Parameters are composed of attribute and value tuples which describe the physical interpretation of the parameter and value it holds respectively. In the HSCT nozzle example, an attribute would be the exit Mach number and when combined with the value of Mach 3.4 it forms a design parameter. The set of attributes can be partitioned according to their dependencies. Thus, it may not be possible to independently control all of the parameters due to relationships between attributes and design constraints. The parameters and attributes are characterized in two groups: design parameters (attributes) and performance parameters (attributes). Design attributes are the characteristics that define a design configuration. In the HSCT nozzle example, wall thickness, material, length and manufacturing processes can be considered as design attributes. Each design attribute is considered to be independent and dependencies in the design attributes can be modeled by intermediate variables. The collection of design attributes is sufficient represent a complete design. The desired product performance characteristics, such as cost and durability, are represented as performance attributes. These attributes are derived from the design attributes, and hence, are dependent variables. The design attributes are transformed into performance attributes by the collective actions of agents in the MADA design ecology. The collection of decisions form the properties of the final design and represent the high level goals of the designers. In a design environment they represent the choices between design alternatives. Parameters and attributes form the structure of the design scenario, while the decisions shape the outcome.

2.1 Design Ecology

The MADA design ecology, shown in Figure 1, contains four distinct entities, 1) design coordination agents, 2) worker agents, 3) tool agents and 4) tools, in addition to the environment in which the agents reside. As mentioned before, the design is represented in a parametric form. The relationships between the parameters and the analysis tools are defined by parameter maps. The relationships between parameters, such as how the cross sectional area of a HSCT nozzle effects the temperature and pressure profile along the axis of the nozzle, can be inferred as the design iterates through several configurations.

The parameter maps are managed by the design coordination agents and the worker agents. The design coordination and worker agents convert the user's goals into manageable and coordinated tasks. The design coordination agents are responsible for determining the processes needed to calculate parameter values and specific tasks for the worker agents. Thus, they provide the necessary intelligence to ensure complete and valid designs. The worker agents use the parameter maps provided by the design coordination agents to satisfy the dependencies and return the result back to the coordination agent. Specialized worker and coordinator agents interact with multiple tools, thereby providing the means for inter-application collaboration.

The tool agents and tool interfaces provide an abstraction layer, which allows the application programs to consume and produce information in a common data and command/communication infrastructure. The tools are individually managed by the tool agents, which are responsible for coordinating requests for information, instantiating and executing the analysis tools and delivering the results. The tool agents have specific information requirements that must be satisfied and these dependencies are managed by

the design coordination agents. Design and analysis tools take independent design attributes and combine them into composite attributes. Tool agents, who manage this process, can also impose constraints at this point and perform simple transformations. Performance tools process design attributes and convert them into performance attributes. This transformation is accomplished by interacting with the design tools and the design coordination agents.

2.2 Implementation

The design architecture is implemented using a multi-layered, multi-agent approach, isolating the complexities of one technology from another. This approach gives the flexibility to change tools and applications without disrupting the entire system. This also gives the environment ability to adapt to the changing technologies without major changes. MADA needs to be able to operate in heterogeneous computing networks and therefore utilizes technologies that facilitate interaction between diverse operating platforms. The MADA environment uses Java as its core language and inherits many key features from it, including the ability to run on any platform that has a Java Virtual Machine (JVM). The tool agents use Remote Method Invocation (RMI) which enables Java applications to invoke methods remotely through lightweight Applications Program Interfaces (API). This allows the agent subsystems to be mobile in non-proprietary ways. Java and RMI by themselves only provide the means in which heterogeneous, mobile agent systems can be built. An agent management system, or Multi-Agent Facility (MAF), on the other hand, provides a common set of resources and services for the mobile agents, such as mobility tracking, data persistence, message passing, naming services and life-cycle support. Individual agents exist in the MAF environment, which provides the required services [2]. The MADA agents communicate information to other agents through a KQML interface [3]. Figure 2 shows a HSCT nozzle being designed using MADA. Additional implementation details of the system can be found in [4].

3 Design Economy

Instead of performing a centralized search on the design space, a distributed methodology is desirable to take advantage of the asynchronous and distributed nature of the underlying design ecology. The search model presented in this paper is based on a collection of individual decision making entities. These entities, or agents, make choices based on limited information and preferences. The choices are formed in a market setting by representing the design and performance attributes as commodities and the stake holders, or decision makers, as consumers.

In addition to the benefits of distributing the computational processes, market based models have also been shown to yield an efficient distribution of stake holder interests through out the system. Thus the combination of design economy and design ecology in MADA provides a

means to balance multiple design criteria while maintaining the complex interactions among the design attributes.

We now formulate the design economy problem so that it can be mapped on to a general equilibrium model. By formulating the design problem in the context of general equilibrium theory we can deploy the techniques used in economics to ascertain the global properties of the design system [5]. The global properties are of interest since they represent the final design configuration generated by the design economy. The aforementioned mapping is defined by three basic entities: 1) consumers, 2) producers and 3) commodities. In the following discussion, we first introduce the concept of general equilibrium and then define the market entities in terms of the design problem.

3.1 General Equilibrium

The concept of general equilibrium has been well studied by economists and is used to model market activity [6, 7, 8]. Economists use equilibrium analysis to model the aggregate behavior of market based systems. Questions about market stability, existence of a solution and other global properties are answered by studying the equilibrium points of the system. In general equilibrium models, we can show that under certain assumptions the market behaves in a predictable fashion, which is desirable for the design search process.

In the economic context, the consumer behavior is dictated by their preferences, which in turn determine their desired set of commodities, or the market bundles. In general equilibrium, the consumers choose market bundles in order to maximize their utility subject to the budget constraints. This desire for goods manifests itself in the market as demand. As the market prices shift due to this demand, the agents try to reallocate their market bundles based on their stated preferences. This iterative activity drives the dynamics of the market economy. The market settles at an equilibrium point when all the consumers are satisfied with their market bundles.

A general equilibrium model consists of the following:

- A fixed collection of k goods or commodities, g^1, \dots, g^k , with an associated price vector \bar{p} in the market.
- A collection of m consumer agents
 - having initial endowment of goods $\bar{\omega} = (\omega_1, \dots, \omega_m)$
 - maximizes stated preferences in the form of a utility function, $u_i = f(\bar{p}, \omega_i \cdot \bar{p})$
- A collection of producer agents
 - with a feasible production set in all feasible production sets $\bar{y} \in Y$, $y_i < 0$ represents inputs, $y_i > 0$ represents outputs
 - maximize profit, $\bar{p} \cdot \bar{y}$

In order to formulate the design problem in terms of the general equilibrium model, we need define consumers, producers and commodities in the design context.

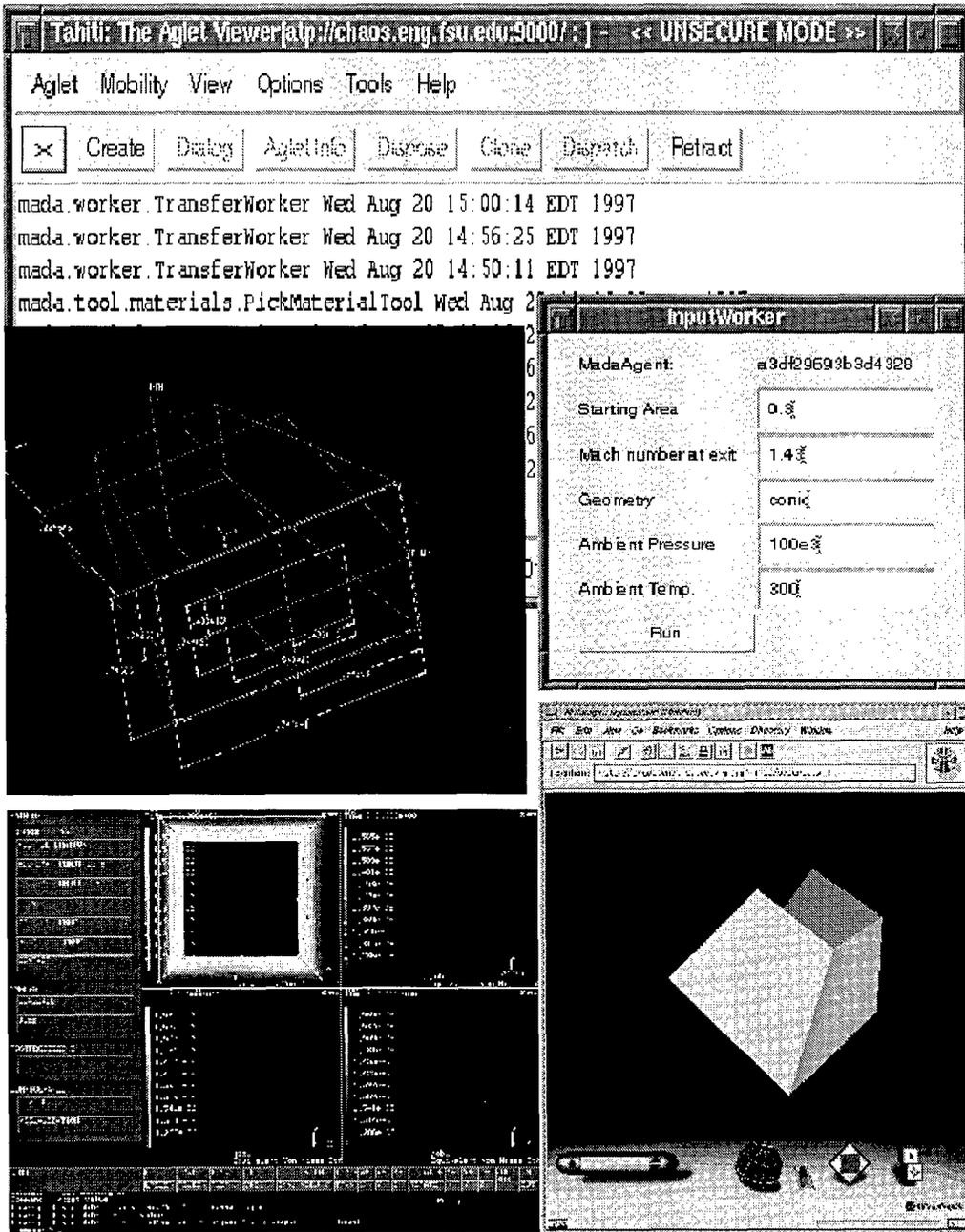


Figure 2: Design of a HSCT nozzle using MADA

3.2 Consumer Agents

Consumer agents are the embodiment of a stake holder in the design process. These agents act in interest of the stake holder they represent. The decision maker's preferences and goals are encoded in the agent using an utility function. Each stake holder's influence on the design is controlled by allocating an initial endowment. This endowment allows us to give different weights to the decision makers in the system.

Definition 1 (Consumer Agent) *Consumer agent i , is a rational economic agent with an initial endowment ω_i that solves the following problem:*

$$\begin{aligned} \max \quad & u_i(\chi) \\ \text{s.t.} \quad & p \cdot x_i \leq \omega_i \end{aligned} \quad (1)$$

where x_i is a set of control commodities, which are explained in the following discussion, and χ is the set of global quantities of the performance commodities, $\chi \equiv f(x_i, g)$, where g is the set of performance commodities.

The consumer agents trade performance measures in order to achieve the highest level of personal utility subject to their resource or endowment constraint. For example, if one stake holder needs a lighter nozzle, it may be willing to allow proportional increase in cost. This activity effectively expresses the consumer's willingness to exchange one performance commodity for another at a marginal rate based on their relative importance. The results in general equilibrium theory ensure that under certain conditions the resulting system will produce an *efficient* or Pareto optimal¹ allocation of performance attributes among the individuals [6, 8]. It is important to note that we are not guaranteed a single equilibrium point or the solution may not be globally optimal according to some aggregate utility metric. However, this problem can be overcome by enumerating through different equilibria in order to ascertain the quality of the design according to a overall utility metric or setting desired utility levels as constraints.

Basing the consumer's utility maximization on the *global* commodity quantity is a departure from the standard general equilibrium model which uses the individual's possession of the commodity to determine the utility. This is done to alleviate the problem of *incremental benefit*, that is, how do you measure the impact of the change in a design attribute on dependent performance attributes and consumer's utility values? In the HSCT nozzle example, if one consumer agent requests a 2% decrease in weight, this decrease may benefit other consumer agents in the economy also. Hence, the utility for an individual consumer needs to be based on the global values of the design and performance attributes. By eliminating the utility proportioning problem the overall system becomes simpler as the consumer agent's utility is based on the global performance measure and not on the fraction of change it is responsible for.

¹It is not possible to increase any one agent's utility without reducing some other agent's utility.

3.3 Performance Commodities

The consumption of commodities drives the market economy and shapes the outcome of the system. Commodities are items that are produced, bought and sold in the market and represent the primary medium through which the consumers interact. In the design economy, the consumers trade performance commodities. The performance commodities are derived from the performance attributes of the design. It is important to note that the consumers do not directly trade in design attributes. Each consumer acquires, or attempts to acquire, performance commodities which will increase its utility value. The price vector, which is based on the demand in the market, governs the cost of trading the performance commodities.

Definition 2 (Performance Commodities) *A performance commodity g^k is a continuously quantifiable real valued performance attribute.*

We consider the commodities in the standard economic sense, *goods*, which generally have positive connotations to the consumer agents. For example, efficiency can be considered a good and having more of this commodity would increase the agent's utility. However, performance attributes do not necessarily conform to this definition. Negative commodities need to be reformulated such that they can be considered as goods. For example, increasing the weight of the HSCT nozzle is generally considered undesirable. Hence, a reformulated commodity could be lightness. Thus, increased consumption of this reformulated commodity would increase the consumer's utility.

3.4 Producer Agents

In order to consume performance commodities they must first be produced. The producers agents consume a set of design attribute as inputs, such as length and material, and convert them into performance attributes, such as weight and cost. The production of performance commodities in MADA is accomplished by performance attribute agents. Formally, this calculation defines the feasible technologies ($y \in Y$) of the producer. The design attributes are managed by the design attribute agents. The performance attribute agents, or producer agents, and the design attribute agents are part of the design economy, as shown in Figure 1.

An interesting situation arises in this formulation when a design attribute (material) creates a positive change in a performance attribute (stiffness) and a negative change in another (weight). It is obvious that if the design attribute were allocated to the two producer agents they would produce conflicting demand patterns. The important issue here is not the allocation of fractional attribute values but possession of control. This conflicting demand for the design attributes is resolved by splitting design attributes into positive and negative control commodities. An intermediate broker (producer) breaks up the design attribute into both positive and negative control commodities. If a producer can increase its profit by lowering a design attribute it acquires more negative control commodities of

the design attribute and vice versa. The producer agents acquire their inputs from the design attribute agents or from their initial endowments.

Definition 3 (Control Commodities) *A control commodity is a guaranteed fractional inclusion (exclusion) of the total value of a design attribute used in the calculation of the performance attributes.*

An agent i can control commodity k in the positive sense by purchasing positive control, c_i^{k+} , or likewise the negative c_i^{k-} . Conservation of control must be maintained, that is $c_i^{k+} + c_i^{k-} = x_i^k$ which represents the range of control.

3.5 Design Commodities

Design commodities are derived from the design attributes and can be classified into two types: singular design commodities and composite design commodities. The amount of a design commodity produced to be converted into a design control commodity depends on the range of a design attribute. Composite design commodities represent a class of dependent design attributes, such as weight and volume. Composite commodities provide a means of controlling aggregate values of dependent variables. The conversion of design attribute values to design commodities is a form of normalization.

In the case of a singular design attribute, a producer agent has the sole initial endowment of the design commodity and produces the associated positive and negative control commodities from it. In the case of composite design commodities, only one producer agent controls each commodity. The producer agents representing singular design commodities use this as their production input.

3.6 Market Dynamics

Once the design problem has been decomposed, the consumer preferences encoded and the market economy defined, the interaction dynamics need to be specified. We use an interaction method called *tatonnement*, which uses an incremental price adjustment process through bidding and auction processes to clear market demand. The market clearing process ensures that supply meets the demand by controlling the prices of the commodities. Other market clearing techniques can also be used to produce similar results. A beneficial property of the *tatonnement* processes is that it can be implemented as a distributed asynchronous protocol which fits well into the MADA distributed environment [9].

The design economy, consisting of the consumers, producers and commodities along with their interaction protocols, evolves through the price negotiation process. This forms the basis of the design search process. The stake holders and their associated consumer agents drive the market with their consumption which cascades down the commodity chain. At every step of the process the production and consumption levels of all commodities are controlled by negotiating their market price through the assistance of the auctioneer and the price bidding processes.

The final market equilibrium represents a balance of the resource usage throughout the system. The equitable distribution of resources results in a robust design as performance commodities are effectively distributed throughout the system. Inequities are represented by a disproportionate amount paid for a commodity by a producer or consumer. Such a point would be represented by market imbalance in a particular commodity.

In general equilibrium the notion of stability has been well studied. The existence of a stable solution, even a unique stable solution, can be guaranteed if the consumers and producers conform to a set of conditions. These conditions are generally based on the convexity of the producer and consumer preferences. In the design context, it is difficult to guarantee these conditions for all the entities. Hence, we empirically test the stability and convergence properties of the systems using our Multi-Agent Diagnostics Convergence Worksheet (MAD-CoW). This worksheet provides a testbed for studying the dynamics of the interaction protocols and the utility values in the market environment. Figure 3 shows an example of three design agents which reach an equilibrium point after 30 iterations.

4 Learning

The market structure of the design environment puts high demands on the analysis software because of the large number of iterations that may be executed before reaching equilibrium. To manage this computational load the tool agents are augmented with a learning mechanism in order to facilitate efficient use of resources. Two mechanisms are currently employed, intelligent tool selection and a output estimation module.

4.1 Intelligent Tool Selection

The parametric nature of the design ecology allows tools with differing capabilities to be substituted into the environment. By allowing multiple tools to be used in the same search process, agents can balance the information fidelity requirements with search time. Analyses performed early in the search processes do not require the same precision as those done later in the search. Hence, faster, lower fidelity tools can be used to focus the search as it progresses. The management of the overall solution quality also important. Using low fidelity inputs makes the use of a high precision analysis tool unnecessary. Managing the entire information quality processes can yield significant computational savings by reducing *fidelity mismatch* and using detailed analysis only when required.

4.2 Neural Network Estimation

Since the market adjustment processes can take many iterations it is desirable to be able to estimate the output of a design tool, thereby reducing the computational overhead.

We present a neural network estimation model for the aerodynamic analysis tool in the HSCT nozzle design, as shown in Figure 4. The aerodynamic analysis tool determines the flow properties inside the nozzle. We presently

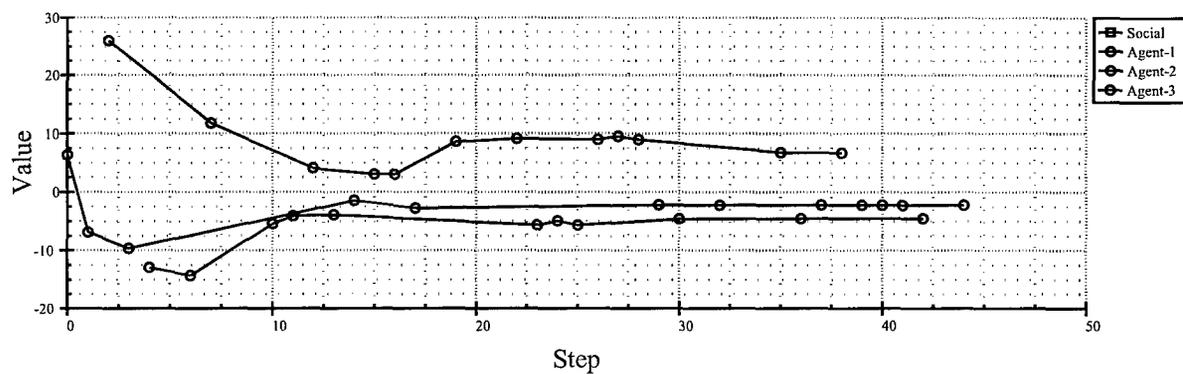


Figure 3: Convergence analysis

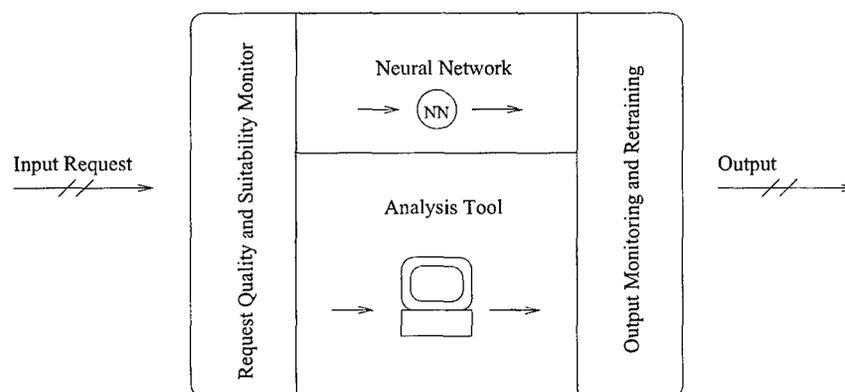


Figure 4: Neural network estimation

use a one-dimensional isentropic flow analysis model, that describes the global characteristics of the flow inside the nozzle with reasonable accuracy. Assumption of sonic flow at the throat gives an area-Mach number relation that relates the Mach number at any location in the nozzle to the ratio of the local nozzle to the sonic throat area. An ideal expansion at the nozzle exit is also assumed in the analysis. The ambient conditions, nozzle pressure ratio (NPR: stagnation pressure/ambient pressure) and the temperature ratio (TR: stagnation temperature/ambient temperature) are specified as the initial nozzle parameters. In practice, the turbine outlet conditions are typically known before the design of a nozzle is initiated. The performance of the nozzle is then characterized in terms of the NPR and TR. For a given throat area and exit nozzle angle, the length of the nozzle can be calculated. Although, the aerodynamic analysis carried out here is relatively simple, the values of most of the important flow parameters inside the nozzle, such as the pressure and temperature on the nozzle walls are captured adequately.

This is a pattern association problem and the network best suited for this is the feedforward network with backpropagation. A feedforward type of neural net is a specific connection structure, where neurons of one layer are only allowed to have connections to neurons of other layers. Backpropagation is a learning algorithm used by nets with supervised learning. A graphical representation of this network can be found in Figure 5, with input nodes, hidden layers and output nodes from left to right respectively. The inputs to the network are 1) nozzle pressure ratio 2) Mach number at exit 3) ambient pressure 4) ambient temperature 5) temperature ratio 6) starting area 7) nozzle angle and 8) number of samples. The outputs generated are 1) maximum temperature 2) length and 3) ratio of starting area to the exit area.

The analysis tool is used to generate the training data set for the neural network. The number of hidden layers, nodes in each hidden layer and the learning rate are experimentally determined. Once the network prediction error is below an acceptable level it is used to estimate the design tool outputs. This significantly speeds up the design search process. The network is verified after a series of runs and the analysis tools are used to update the weight matrix in the net. The decision whether to use the neural net estimation or the analysis tool results can be based on a cost function reflecting the required fidelity and criticality of the results.

Table 1 shows four test results of the neural network for the aerodynamic analysis tool. Best results were obtained when the training was done for 1000 cycles with the structure shown in Figure 5 and the learning rate was set to $0.25(\alpha)$. Note that all inputs to the net were normalized between 0 and 1.

5 Summary

The distributed design environment presented in this paper can be applied to the design of complex products. The distributed nature of the computations and decision making

are well suited for large scale applications. By representing the design search problem in an economic context, we can study the properties of the design configurations generated by this methodology. Although this method does not guarantee an optimal design, it can be used to identify acceptable alternatives.

The future research in this area will focus on testing the architecture on complex assemblies, involving multiple design domains, such as mechanical, electrical and computational systems. The learning tools will also be refined to include the overall design performance prediction capabilities.

6 Acknowledgments

This work was funded in part by Grant # NAG-2-1114 from the NASA Ames Research Center and Grant # 9978923 from the National Science Foundation. The authors acknowledge the assistance of Pravin George in implementing the learning methods in the architecture.

References

- [1] S. H. Thomke, "Managing experimentation in the design of new products," *Management Science*, vol. 44, pp. 743-762, June 1998.
- [2] R. Levy, K. Erol, and H. Mitchell, "A study of infrastructure requirements and software platforms for autonomous agents," in *Proceedings of ICSE-96*, July 1996.
- [3] T. Finin, R. Fritzson, and D. McKay, "A language and protocol to support intelligent agent interoperability," in *Proceedings of the CE & CALS Washington '92 Conference*, June 1992.
- [4] A. Deshmukh, A. Krothapalli, T. Middelkoop, and C. Smith, "Multi-agent design architecture for integrated design systems," *AIAA Journal of Aircraft*, 1999. Accepted with revisions.
- [5] T. Mullen and M. P. Wellman, "Some issues in the design of market-oriented agents," in *Intelligent Agents II: Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996.
- [6] Y. Balasko, *Foundations of the Theory of General Equilibrium*. Academic Press, Inc., 1988.
- [7] J. Quirk and R. Saposnik, *Introduction to General Equilibrium theory and Welfare Economics*. McGraw-Hill, 1968.
- [8] T. Rader, *Theory of General Economic Equilibrium*. Academic Press, 1972.
- [9] J. Q. Cheng and M. P. Wellman, "The WALRAS algorithm: A convergent distributed implementation of general equilibrium outcomes," *Computational Economics*, vol. 12, pp. 1-23, 1998.

Inputs to the Neural Net						Outputs from the Neural Net			Net Error
npr	me	pa	ta	tr	astar	temp	length	aebast	
0.348	0.607	0.857	0.938	0.332	0.278	0.355	0.337	0.641	0.110
0.961	0.605	0.632	0.685	0.388	0.284	0.248	0.319	0.635	0.080
0.792	0.599	0.774	0.800	0.350	0.230	0.288	0.289	0.622	0.076
0.997	0.594	0.443	0.235	0.005	0.112	0.042	0.247	0.599	0.062
0.573	0.562	0.487	0.534	0.284	0.134	0.145	0.231	0.576	0.050

Table 1: Neural network estimation results

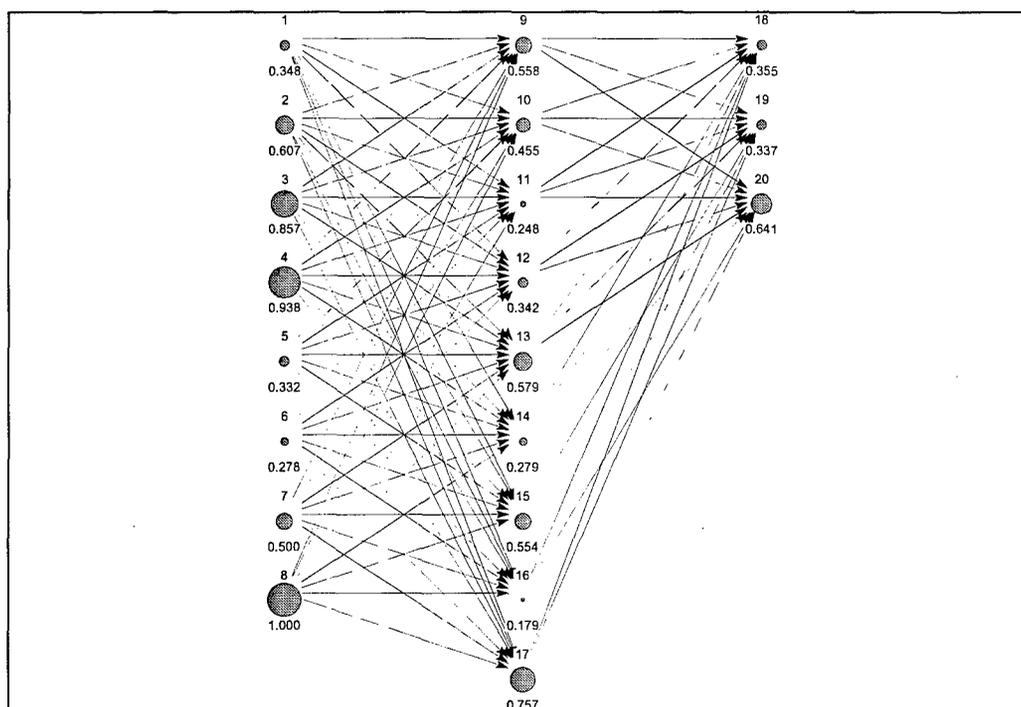


Figure 5: Neural network layout