A Marketplace of Design Agents for Distributed Concurrent Set-Based Design[1]

H.V.D. Parunak*, A.Ward[+], M.Fleischer*, and J.Sauter*
*Industrial Technology Institute [+]Ward Synthesis
van@iti.org

Abstract

The RAPPID project (Responsible Agents for Product-Process Integrated Design)[2] is developing agent-based software tools and methods for using market place dynamics among members of a distributed design team to coordinate set-based design of a discrete manufactured product. This report begins with an overview of the RAPPID vision, in which the components being designed (represented by their designers) buy and sell the characteristics they wish to assume. It describes the entities that interact in the market economy and outlines the market protocols through which trades are made.

1. RAPPID Overview

A designer seeks to embed a set of *functions* (e.g., optical, electromechanical, control) in an artifact with specified *characteristics* (e.g., weight, color, complexity, materials, power consumption, physical size). The functional view drives most designs, since it distinguishes the disciplines in which engineers are trained and in support of which design tools are available. Conflicts arise when different teams disagree on the relation between the characteristics of their own functional pieces and the characteristics of the entire product. Some conflicts are within the design team: How much of a mechanism's total power budget should be available to the sensor circuitry, and how much to the actuator? Others face design off against other manufacturing functions: How should we balance the functional desirability of an unusual machined shape against the increased manufacturing expense of creating that shape?

It is easy to represent how much a mechanism weighs or how much power it consumes, but there is seldom a disciplined way to trade off weight and power consumption against one another. The more characteristics are involved in a design compromise, the more difficult the trade-off becomes. The problem is the classic dilemma of multivariate optimization. Analytical solutions are available only in specialized and limited niches. In current practice such trade-offs are sometimes supported by processes such as QFD (Quality Functional Deployment) or resolved politically, rather than in a way that optimizes the overall design and its manufacturability. The problem is compounded when design teams are distributed across different companies.

RAPPID uses a marketplace to set prices on each characteristic of a design. Agents representing each component buy and sell units of these characteristics. A component that needs more latitude in a given characteristic (say, more weight) can purchase increments of that characteristic from another component,

---

[1] Forthcoming at ISPE/CE97: Fourth ISPE International Conference on Concurrent Engineering: Research And Applications, Troy, Michigan, August 20-22, 1997.

[2] RAPPID is sponsored by the Rapid Design Exploration and Optimization (RaDEO) program (formerly MADE) of DARPA, and is administered through the AF ManTech program at Wright Laboratories under the direction of James Poindexter. In addition to the authors, the project team includes Steve Clark, Mike Davis, Bob Matthews (all ITI) and Mike Wellman (University of Michigan). The RAPPID prototype is being tested on the design of military land vehicles at the U.S. Army's Tank and Automotive Command (TACOM) in Warren, MI, with the support of the Technology Integration Division.

but may need to sell another characteristic to raise resources for this purchase. In some cases, analytical models of the dependencies between characteristics may help designers estimate their relative costs, but even where such models are clumsy or nonexistent, prices set in the marketplace define the coupling among characteristics.

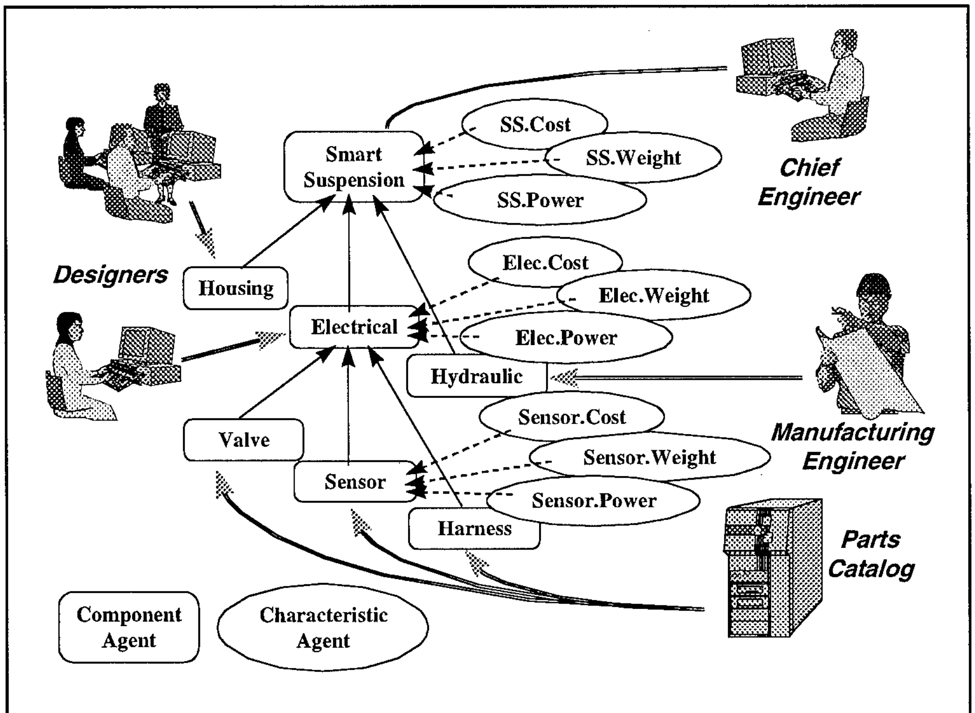

Figure 1: The RAPPID Design Ecology for a New Tank Suspension

Figure 1 shows a design decomposed into Component Agents (rounded rectangles), each with one Characteristic Agent (ovals) for each dimension in the design space. For example, the "SS.Weight" Characteristic might represent the constraint that the entire product weigh between 5 and 10 kg. The topmost Component represents the complete product (in this case, a new suspension for a tank), and is the concern of the Chief Engineer, who reflects the Customer's requirements in the initial allocation of design space. The bottom-most Components are either custom-manufactured or (in the Figure) selected from an on-line Parts Catalog. Designers, who typically have responsibility for intermediate levels of the product tree, propagate the constraints from the top and bottom of the tree toward each other. Each Component (either automatically or under guidance from its Designer) buys and sells allocations on its Characteristics to and from other Components.

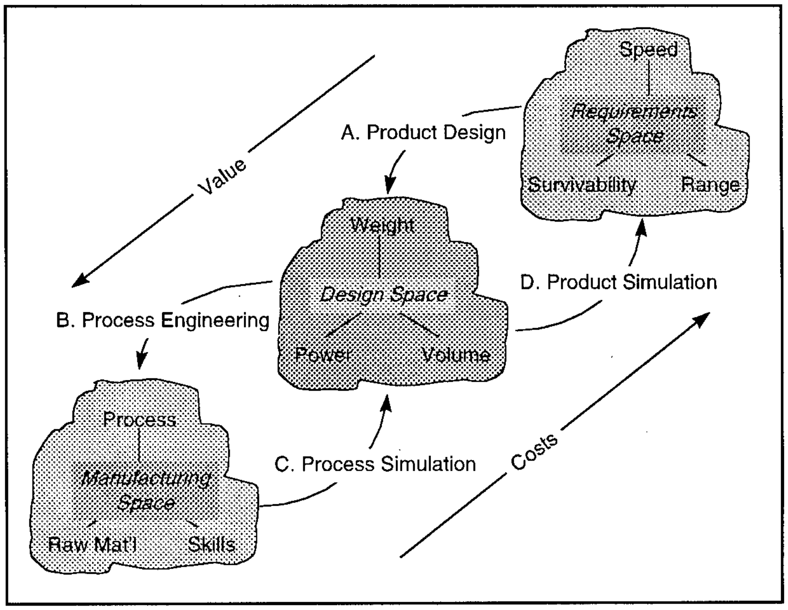

Figure 2: : IPPD as a Mapping Among Spaces

A product exists not only in Design Space (characterized by features such as weight, power, and shape), but also in Manufacturing Space (characterized by such features as Process, Raw Material, and Operator Skill) and Requirements Space (characterized by behavioral features such as Speed, Range, and

Survivability). As illustrated by the curved arrows in Figure 2, the problem of Integrated Product-Process Design (IPPD) can be viewed as one of mapping among these spaces. Value is defined by the customer, starting in Requirements Space, and flows down to design and then to manufacturing. For example, through battlefield simulations, the customer can determine the incremental value of another 5 kph of vehicle velocity, or another 20 km of range. Cost is defined by manufacturing flows up through design to the customer. For example, the designer (with input from manufacturing) determines the cost of providing an increment of velocity or range. A marketplace in these three spaces supports mapping in both directions among them. The current version of RAPPID focuses on a marketplace in the design space. Later research will extend the market approach to the neighboring spaces.

2. Basic Concepts

RAPPID rests on three basic concepts: markets as a mechanism for coordinating distributed decision-making, set-based design, and the use of computerized tools in partnership with humans rather than as a replacement for them.

2.1 Market-Based Control

Researchers addressing distributed problems in a wide range of domains have recently begun to turn to market-based mechanisms for coordination. [Clearwater 1996] offers a convenient collection of applications to fields as diverse as computer network control, memory allocation, factory scheduling, pollution management, and air-conditioning load balancing. In all of these areas, competitors for scarce resources can efficiently express their needs in terms of a common currency, and the balance between supply and demand can set prices for those resources that rationalize the distribution of resources across competitors.

[Wellman 1995] reports some early experiments in market-based automatic configuration of a manufactured product. RAPPID extends these methods to a hybrid system in which humans as well as computers can participate, and supports a set-based approach to design.

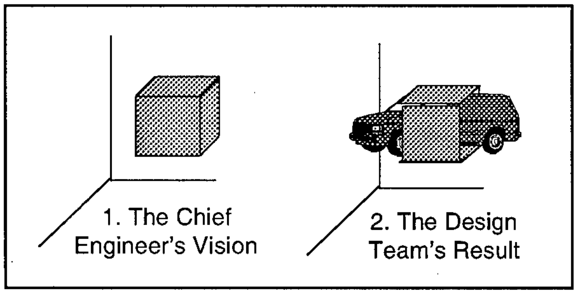

Figure 3: Building a Design to Fit a Design Subspace

2.2 Set-Based Ideas

A product's characteristics can be thought of as dimensions of a Cartesian space within which the product is defined. Traditionally, designers seek to *build* a design to fit a predefined subspace of characteristics, without knowing in advance whether any acceptable design fits (Figure 3).

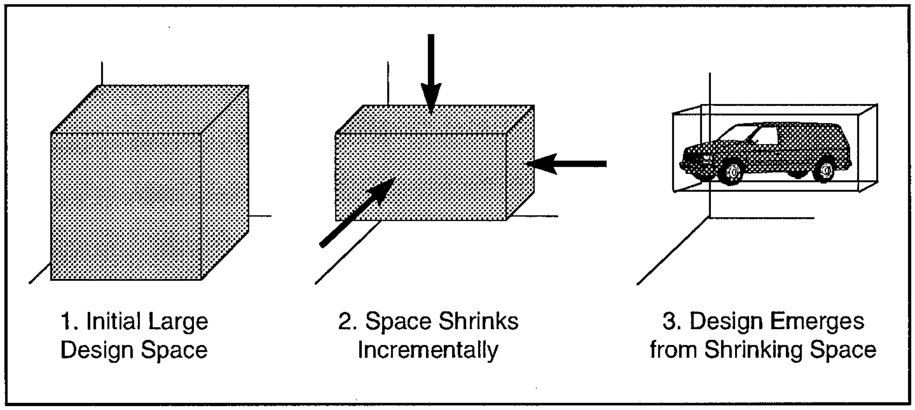

Figure 4: Shrinking Design Space to Discover a Design

Toyota has pioneered a

more promising vision that begins with a design space much larger than necessary, and then shrinks it incrementally to *discover* where the desired design is (Figure 4) [Ward et al. 1995].

Current design tools offer no quantitative support for this vision. With a market in design characteristics, low prices identify slack characteristics (dimensions of the space) that the chief engineer can collapse by buying up allocations of that characteristic. This action simultaneously reduces the amount of that characteristic available for use by designers and increases the funds in the system available to purchase other characteristics to compensate for the decrease in the given characteristic. As designers buy and sell, the relative prices of the various characteristics change, identifying a new slack dimension that can further shrink the design space.

Set-based design offers several advantages over conventional approaches.

- Traditional design evaluates one point solution after another in a hill-climbing search. For many design domains, the evaluation of a point can be expensive, time-consuming, or both: a detailed finite-element analysis might require a day of Cray computer time, and some tests require construction of a vehicle mock-up for crash testing. Furthermore, point-based design serializes design decisions, causing further delays.

- A shrinking design space can guarantee convergence properties that would be difficult to ensure otherwise among a distributed team. If everyone is shrinking his own subspace, the system as a whole must be converging.

- Flow fields are an important mechanism for emergent organization among agents [Parunak 1997]. The flow of currency in a marketplace is one important flow field; the shrinking space approach provides another. Instead of imagining that boundaries contract in a space of uniform density, hold the boundaries fixed and permit the density of the space to vary. A set of design options is then like a Dutch polder from which more and more water is pumped out. In distributed set-based design, the options discarded by one designer guide other designers. That is, the flow of design options across the boundaries of design sets helps organize the community.

2.3 RAPPID as a Situated System

RAPPID does not automate the entire design problem. Its market mechanisms function alongside conventional interactions, which we describe as SLOWH (Standard Legacy-Oriented Work Habits). The market mechanisms themselves are not entirely automated, but are implemented partly in computer algorithms and partly in human behaviors, a mixture that we describe as "hybrid carbon-silicon systems." Table 1 summarizes these distinctions.

3. Components, Characteristics, and Markets

3.1 Components, Characteristics, Constraints, and Markets

Agents (software objects with individual threads of control and self-initiative) can be used to represent any of four kinds of entities in a market-based design system. The current implementation of RAPPID

Table 1: RAPPID/SLOWH and Carbon/Silicon Distinctions

	RAPPID (Market Mechanisms)	SLOWH Mechanisms
Carbon (Human)	RAPPID-trained designers	Traditional human interactions (phone conversations, meetings, back-of-envelope sketches, prototype models, ...)
Silicon (Computer)	RAPPID server and clients	Traditional tools for CAD, FEM, Workflow, data exchange, ...

represents only Components and Markets as agents.

A *Component* is a node in the tree that represents the structure of the product. In general, a designer or design team is assigned to each component. Depending on the design approach that is used, the tree may or may not be defined in advance, but it is still important to distinguish between the component agent, which represents a specific functional slot in the design, and a specific candidate to fill that slot. For example, the transmission agent in the design of a power system might consider several different physical transmissions. An alternative approach is for each physical component to function as an agent, competing for a role in the design. While this approach may be useful for catalog-based design, the more fundamental view of the product node as the component agent supports a broader set of problems, including those in which the specific physical component has not yet been defined.

A *Characteristic* is a definable attribute or parameter of a component, such as its weight, power consumption, RPM, torque, or size. Characteristics are defined per component. The weight of (say) the motor is a different characteristic than the weight of the transmission, though both are of the same characteristic type.

A *Constraint* is a relation between two or more characteristics. Constraints typically arise either from laws of nature (e.g., power consumption equals voltage times current flow) or design decisions (e.g., the output RPM from the motor equals the input RPM to the transmission; a given RPM and torque characterize the same shaft). Initially, we expect most constraints to exist in the minds of human designers, but we are building a role for them into our architecture to permit them to be captured and automated in later versions.

A *Market* is a process that maps potential buyers and potential sellers of a good to one another and optionally to a price at which a sale can take place. The goods traded in such a market are characteristics or options for characteristics. Each distinct good requires a separate market, and markets for different goods may have different protocols. We visualize each market as existing in the form of a server on a network. In practice, many markets might exist on a single server, but this implementation detail makes no difference to the actual operation of the system.

3.2 Different Kinds of Characteristics

All characteristics are not created equal. A given characteristic may be classified on the basis of the number of components to which it applies, whether its aggregation across components is additive or not, and whether it is coupled to other characteristics.

3.2.1 Scope of Characteristics

A given characteristic may be meaningful only *internal* to a single component, as an *interface* between selected components, or over the entire *system.*

An *internal* characteristic is meaningful only within a component, and it is defined entirely by the designer or design team responsible for that component. If the component is atomic (that is, with no further decomposition and assignment to lower-level design teams), RAPPID mechanisms play no role in the management of its characteristics. However, if the component is at some higher level of the product decomposition tree, characteristics that are internal to it may be either interface or system characteristics among its sub-components.

Interface characteristics enable the functional interaction of the components within a system. A classic example is the torque and RPM between a motor and a transmission. Only components that directly interface with one another trade in interface characteristics, and the issue they seek to resolve is compatibility between cooperating components rather than distribution across competing components. Little or no vertical information movement (that is, between components and their system) is needed to determine interface characteristics. Mating components need only agree on the required interface characteristics and to achieve those characteristics within the scope of their allocated system characteristics. Not all interface characteristics need to match exactly on both sides of an interface, since sometimes one component in the interface simply requires that it receive at least (or at most) some amount of a characteristic from its partner.

A *system* characteristic must be shared among all sub-components of some component. Thus all components in a system are potentially interested in the system characteristics for that system. The overall budget for a system component is set by the parent component (the "system"), which may be represented either directly by the customer, or by the chief engineer acting as surrogate for the customer. In a complex product with several layers to the product tree, transactions concerning system characteristics have a strongly vertical flavor. Although peer components do trade back and forth in system characteristics, the constraints on these characteristics are imposed from above, and components pass them on to lower-level sub-components.

3.2.2 Aggregation of Characteristics

Characteristics that apply to more than one component may be either additive (like weight or power) or non-additive (like resonant frequency or volume). The total system value for additive characteristics is the sum of the component values. If one component consumes more of an additive system characteristic, other components must make do with less. Non-additive characteristics aggregate in more complicated ways. Interface characteristics are intrinsically non-additive, and many system characteristics are additive, but there are non-additive system characteristics that in some ways resemble interface characteristics.

Table 2 summarizes the three kinds of non-internal characteristics. The arrows in the "System (Non-Additive)" column indicate whether the particular feature for these characteristics is more like additive system characteristics or interface characteristics.

3.2.3 Coupled Characteristics

Some sets of characteristics are tightly coupled, and must be varied together in exploring the space of possible designs. Such coupled characteristics frequently arise in dealing with non-additive characteristics (both system and interface), which often come in coupled sets. One cannot get torque on a shaft without also getting RPM, so markets to deal in these coupled characteristics must provide ways of ensuring that

Table 2: Classes of Design Characteristics

	System (Additive)	System (Non-Additive)	Interface (Non-Additive)
Examples	Weight, Power Consumption	Volume, Resonance	Torque, RPM
Main info mvmt	Vertical	←	Horizontal
Major constraints	Among components (e.g., weight)	← →	Among characteristics (e.g., torque and rpm)
Problem to be solved	Allocation of scarce characteristic among competing components	←	Compatibility of characteristics between cooperating components
Quantitative constraints (min, max, range)	Total amount of characteristic across the system (sum of values for components)	← (not accessible as a sum)	The amount of characteristic between mating components
Interested components	All components in a system or subsystem (e.g., "power" is relevant only to electrical subsystems)	←	Only components that interface directly with one another.

they trade together. A buyer who advertises for torque and RPM will not be happy receiving one bid for the appropriate torque but a useless RPM from one vendor, another for the right RPM but a useless torque from another. Because these characteristics are coupled non-additively, they need to be bought and sold together.

4. The RAPPID Market Protocols

RAPPID has explored a variety of extensions to conventional market mechanisms. This section introduces the basic supply-and-demand market, which is appropriate for additive, uncoupled goods, and extends it to deal with set-based design. Non-additive or coupled goods require a different approach, which is also described.

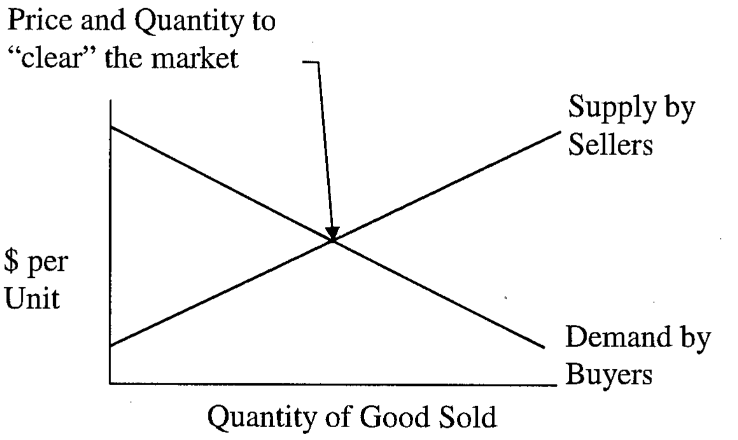

Figure 5: Schematic Supply and Demand Diagram

4.1 Traditional Supply-and-Demand Markets

A traditional market for additive goods accepts bids from both buyers and sellers, and closes at the price and quantity at which supply equals demand. The associated supply and demand diagram (SDD, shown schematically in Figure 5) shows how the amount of a good demanded by buyers and made available by sellers varies with the price of the good. The “Supply by Sellers” curve shows how more goods are available for sale at higher prices than at lower ones. The “Demand by Buyers” curve shows how buyers demand fewer goods at higher prices. These curves have slopes of opposite signs (positive for supply, negative for demand) for a wide range of commodities in competitive markets. As a result, the supply and demand curves for a commodity cross somewhere in the (Price x Quantity) space, at the price where the quantity of goods offered for sale on the market just equals the quantity demanded by buyers. Sellers willing to sell at or below this price can be paired with the buyers willing to buy at or above this price. All the goods offered at this price will be bought, no one who wishes to buy at this price will be refused, and the market is said to “clear.”

A good’s SDD is compiled from bids by potential buyers and sellers. At each price level, the diagram shows the quantity offered for sale at or below that price, and the quantity desired by buyers at or above that price. Because the goods are additive, each curve aggregates information across participants—supply across sellers, and demand across suppliers. A supplier offering 5 units at $3 will contribute 5 units to every point on the supply curve at $3 or higher, since he would naturally sell at higher prices. A buyer asking for 8 units at $7 would also contribute 8 units to demand at lower prices, since she would naturally buy those units at lower prices as well.

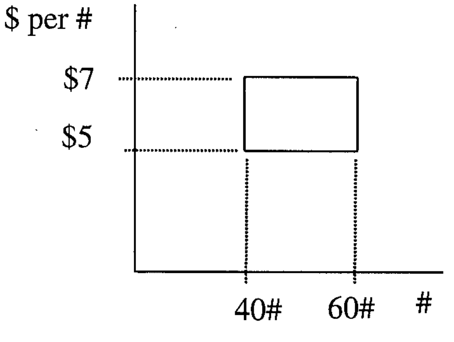

Figure 6: A Set-Based Bid for Weight

4.2 Set-Based Bids

RAPPID urges designers to think in terms of sets of designs rather than specific point solutions. Using RAPPID, a designer does not ask, "What would 50# of vehicle weight be worth to me?" but rather, "I need at least 40# and perhaps as much as 60#. What range of prices am I willing to pay?" Initially, these windows on characteristics and prices are set quite wide. By exchanging rough estimates with one another, members of the team eliminate design alternatives that they would otherwise have considered, narrowing their own windows. When detailed evaluations finally become necessary, they can be restricted to the small region of design space to which the windows have converged over time.

A designer's set-based bid for a characteristic is a rectangle of low and high values along each dimension of the Cartesian plane defined by # and $, as in Figure 6.

An extension of the SDD aggregates set-based bids for additive uncoupled characteristics. The lower-left hand corner of a buyer's bid rectangle is the buyer's most conservative bid, as is the upper-left hand corner of a seller's rectangle. The upper-right-hand corner of a buyer's rectangle represents the least-conservative outcome. The agent may end up valuing weight so highly that it will pay its maximum price for the maximum amount of weight. For a seller, the lower-right-hand corner represents a "fire sale" mentality, a willingness to sell all its weight for the lowest possible price.

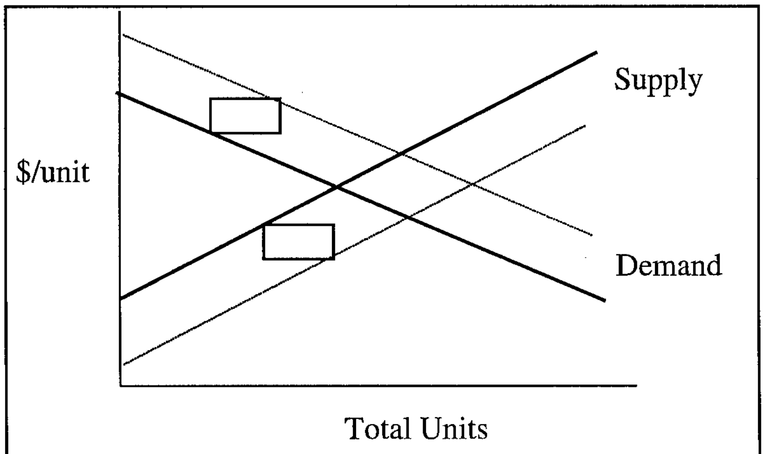

Figure 7: Set-Based SDD

Figure 7 sketches the result of combining the worst-case and best-case curves on a single SDD. The solid lines represent the worst-case bids. The dashed lines represent the best-case bids. The small rectangles suggest how these lines are generated from the "rectangle" interpretation of a set-based bid.

The four lines describe a lozenge at the center of the figure. Each of its vertices conveys useful information.

- The left vertex shows where the market would clear under the most conservative assumptions. Because it is a "worst-case" point, it is safe to let the market actually close at this point, and trade the indicated number of pounds. This action will change the constraints on some of the players, and lead to a different situation in subsequent market rounds.
- The right vertex shows the largest amount that could possible clear, given the current estimates of the participants, and helps estimate the range needed in the final design.
- The top and bottom vertices estimate maximum and minimum clearing prices, and summarize the likely prices that the characteristic might command in this design activity.

Only one point in each participant's rectangle leads to an immediate sale of the characteristic being traded. The rest of the interval estimates a participant's possible maximum supply or demand throughout the design process, which we translate into market terms by assigning to the participants initial stakes in option markets for the good. The larger the interval specified by a user, the larger the option position it is required to take.

4.3 Trading Non-Additive Characteristics

SDD’s are based on the idea of adding together the supply or demand for a characteristic at various price levels. This addition is impossible for non-additive characteristics, and so the computation of a clearing price by such a computation cannot be applied. However, the language of bids is still useful in enabling designers to communicate their preferences.

RAPPID is exploring a successive-refinement interface to facilitate this exchange. Participants offer set-based bids either to supply or to consume a given characteristic, and indicate qualitatively how their price varies with the value of the characteristic within this region. The RAPPID market server identifies regions of the <characteristic, price> plane in which sellers and buyers overlap. In general, such an overlap is a smaller region, but still a set of points. Depending on the qualitative shapes that participants indicate for their price curves, the market server can sometimes shrink the area even further. The market server communicates the bounds of this region back to the participants, who refine their designs in the direction indicated by the subregion and submit revised, still more constrained bids. The process repeats until the region is so small that conventional point-based design methods can be applied.

4.4 Trading Coupled Characteristics

Non-additive characteristics often occur in closely coupled sets, such as torque and RPM values between two components in a power transmission system. The RAPPID model for trading coupled characteristics can be compared with the way one buys a car, by specifying the model and then selecting options for the various subsystems and accessories. So we can conceive of a designer’s assigning a value both to the base system or interface and to ranges of its various characteristics. In the case of rotational power, the consumer seeks to buy a rotational interface (the base) with two characteristics (the accessories), torque and RPM. The buyer specifies a range of prices for the base. For each characteristic, the buyer also specifies an upper limit for the maximum additional price above the base that could be realized by an appropriate choice for that characteristic, and a qualitative indication of how the price varies with the value of the characteristic. The bottom of the value range for each characteristic is always zero, and so is not explicitly specified.

As design progresses and designers explore the characteristics, they add price to the base. That is, if a designer narrows the range of a characteristic in a way that guarantees $30 of incremental contribution to the customer, the upper limit of that characteristic drops by $30, and both limits of the base rise by $30. When the process finally converges, no incremental price remains in the accessories, and all of it is in the base.

5. Summary

Collaborative design of complex manufactured products is a difficult problem in multidimensional optimization. An important part of this problem is allocating product characteristics to the various design teams. Preliminary experiments with simple design problems show that the RAPPID market-based environment can support effective communication among designers. The system is being refined and further experiments will validate its applicability to real industrial problems.

6. References

[Clearwater 1996] S.H.Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Management.* Singapore: World Scientific.

[Parunak 1997] H.V.D.Parunak, “‘Go to the Ant’: Engineering Principles from Natural Agent Systems.” *Annals of Operations Research* (forthcoming).

[Ward et al. 1995] A.Ward, J.K.Liker, J.J.Cristiano, and D.K.Sobek II, “The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster.” *Sloan Management Review* (Spring), 43-61.

[Wellman 1995]M.P.Wellman, "A Computational Market Model for Distributed Configuration Design." *AI-EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 9:2 (April), 125-34.

INTERNET DOCUMENT INFORMATION FORM

A . Report Title: A Maketplace of Design Agents for Distributed Concurrent Set-Based Design

B. DATE Report Downloaded From the Internet 4/5/99

C. Report's Point of Contact: (Name, Organization, Address, Office Symbol, & Ph #): Industrial Technology Institute
P.O. Box 1485
Ann Arbor, MI 48106

D. Currently Applicable Classification Level: Unclassified

E. Distribution Statement A: Approved for Public Release

F. The foregoing information was compiled and provided by:
DTIC-OCA, Initials: VM_ **Preparation Date**: 4/5/99__

The foregoing information should exactly correspond to the Title, Report Number, and the Date on the accompanying report document. If there are mismatches, or other questions, contact the above OCA Representative for resolution.