MEDIATOR: TOWARDS A NEGOTIATION SUPPORT SYSTEM

Matthias Jarke
M. Tawfik Jelassi
Melvin F. Shakun

May 1985

Center for Research on Information Systems
Computer Applications and Information Systems Area
Graduate School of Business Administration
New York University

Working Paper Series

CRIS #93

GBA #85-36(CR)

ABSTRACT

MEDIATOR is a negotiation support system (NSS) based on evolutionary systems design (ESD) and database-centered implementation. It supports negotiations by consensus seeking through exchange of information and, where consensus is incomplete, by compromise. The negotiation problem is shown --graphically or as relational data in matrix form-- in three spaces as a mapping from control space to goal space (and through marginal utility functions) to utility space. Within each of these spaces the negotiation process is characterized by adaptive change, i.e., mappings of group target and feasible sets by which these sets are redefined in seeking a solution characterized by a single-point intersection between them.

This concept is being implemented in MEDIATOR, a data-based micro-mainframe NSS intended to support the players and a human mediator in multi-player decision situations. Each player employs private and shared database views, using his/her own micro-computer decision support system enhanced with a communications manager to interact with the mediator DSS. Sharing of views constitutes exchange of information which can lead towards consensus. The human mediator can support compromise, as needed, through use of solution concepts and/or concession-making procedures in the NSS model base. As a concrete example, we demonstrate the use of the system for group car buying decisions.

Table of Contents

1. INTRODUCTION

Negotiation support systems (NSS)--computer assisted negotiations--provide decision support in problems involving multiple decision makers, thus extending decision support systems (DSS)--e.g., see Keen and Scott Morton (1978), Sprague and Carlson (1982), Bonczek et al. (1981)--where the initial emphasis has been on single decision maker situations.

In general in NSS we are interested in multiplayer, multicriteria, ill-structured, dynamic problems. Shakun (1981a, 1981b, 1986) develops evolutionary systems design (ESD) as a methodology for problem definition and solution (design) in complex contexts involving multiplayer, multicriteria, ill-structured, dynamic problems. In particular, in this paper ESD is used as a basis for MEDIATOR, a system designed to support negotiations in a setting which we now describe in overview form and develop in detail in the sections below.

1.1. Negotiation Setting Overview

A group of N players is involved in negotiations. A human mediator supports these negotiations and he in turn is supported by the negotiation support system, MEDIATOR. The (human) mediator supports negotiations by assisting the players in a process of consensus seeking within which compromise is possible. Using MEDIATOR, the mediator assists in consensus seeking by aiding the players to build a common (group) joint problem representation of the negotiations. The negotiation problem representation is shown by MEDIATOR--graphically or as relational data in matrix form--in three spaces as mappings from control space to goal space (and through marginal utility functions) to preference (here utility) space. (In some cases involving risk a fourth space, criteria space, can be used between goal space and preference space--see Giordano et al. (1985) and Shakun (1986)). These spaces can be redefined while using MEDIATOR. For use of a goals/values referral process to redefine goal space see Shakun (1981a, 1986).

At each stage of the negotiations, the common joint problem representation shows the acknowledged degree of consensus (or conflict) among the players, i.e., at each stage players may show different individual problem representations. The evolution of problem representation can be described as a process of consensus seeking--through sharing of views which constitutes exchange of information--within which compromise is possible. The mediator can support compromise through use of axiomatic solution concepts and/or concession-making procedures in the MEDIATOR model base. Computer display of the evolving problems representation can be used to support continued consensus seeking. In each space (control, goal and preference) the negotiation process represents adaptive change, i.e., mappings of group target and feasible sets in seeking a solution--a single point intersection between them (Shakun 1985, 1986).

In the basic scenario as described above, we think of the mediator as supporting the negotiations and in turn being supported by MEDIATOR, but not himself deciding on them. However, MEDIATOR should also be useful in compulsory arbitration where the mediator decides (chooses) the solution. In some contexts, the mediator can be a group leader, e.g., the president of a company, who finally makes a decision supported by MEDIATOR. In other contexts, MEDIATOR could support the players directly without the use of a human mediator. Here we work with the basic scenario as noted above.

1.2. Database-Centered DSS Design Overview

A number of DSS design strategies have been proposed, including those that start from the decision models used, from the user interfaces requires, or from a task analysis. In organizations where decisions are based on large amounts of existing data, it seems more natural to follow a database-centered approach. This method embeds the decision models and user interfaces of a DSS in an database management environment which provides them with data, stores their execution sequences, and retains their results. A database approach to DSS was first proposed by Donovan (1976) for single-user DSS and later extended by Blanning (1984), Jarke (1981), and others to cover not only the data management but also the model management and multiuser aspects of DSS.

In the negotiation support setting discussed in this paper, the database is also used as a communication center among the mediator and the players. Besides providing the initial data underlying the problem to be solved, the DBMS also manages the evolving group joint problem representations. Furthermore, it provides a large number of tools for generating this joint problem representation and protecting it against unauthorized or erroneous access.

1.3. Paper Outline

In the following sections, we develop this negotiation support system concept in detail. In section 2 we summarize the single decision maker case as background for the group negotiation problem discussed in section 3, based on (Shakun, 1985). In section 4 we illustrate the use of MEDIATOR by an application to group car buying. The database-centered system architecture for MEDIATOR is developed in section 5. Section 6 presents concluding remarks.

2. THE CASE OF ONE DECISION MAKER

A DSS for MCDM involving one decision maker and applied to car buying is discussed in detail in Jacquet-Lagreze and Shakun (1984). Consider a set A of strategies (controls, inputs, decisions, choices, actions). In the car buying decision, A is the set of available cars representable by positive integers in R^1, car space. Let g be a function from A to R^p, the p-dimensional real vector space. which characterizes outcomes (goals, outputs, consequences, characteristics, criteria). In case of cars, the criteria include price, gas consumption, space, etc. Then $y=g(a)$ for $a \in A$ is a vector of R^p representing the outputs of a particular input choice, a; g(A) is the set of possible outputs representing technologically feasible performance. These outputs are generally constrained a priori by preliminary goal target Y_0 information. For example, these constraints could be $Y_0 = \{y \varepsilon R^p: y_i \geq b_i,\ i=1,\ldots,p\}$. The intersection of Y_0 and g(A) is called $g(A_0)$, a set of a priori admissible outputs. $A_0=\{a \varepsilon A: g(a) \varepsilon Y_0\}$ is the corresponding set of a priori admissible inputs.

In addition to the admisible sets of cars, A_0 and goals, $g(A_0)$ we have a preference structure defined on $g(A_0)$. Here we assume a utility function u(y) which is nonlinear and additive:

$$u(y) = \sum_{i=1}^{p} u_i\ (y_i) \qquad (1)$$

With the UTA utility assessment procedure (Jacquet-Lagreze and Siskos, 1982) implemented in the microcomputer program, PREFCALC (Jacquet-Lagreze, 1985), the marginal functions $u_i(y_i)$ are taken as piecewise linear and nondecreasing or nonincreasing. Based on UTA, a disaggregation-aggregation learning process involving both wholistic and analytical judgments is implemented. Working with a small sample $A_1 \subseteq A_0$, a decision support system (Jacquet-Lagreze and Shakun, 1984) can aid a decision maker in defining his utility function (1). Applying the utility function to the set of cars A_0 results in a ranking of cars according to their numerical utilities. The car with the maximum utility is the buying decision. Figure 1 shows the criteria space.

Figure 1

The technologically feasible set g(A) intersects the a priori goal target Y_0 to give the a priori admissible set $g(A_0) = g(A) \Omega\ Y_0)$ which typically in car buying has many points. In order to find a single point solution in criteria space, the intersection set $g(A_o)$ must be reduced in size. This can be done either by contracting Y_o or $g(A)$. Since for cars the latter set is fixed at a particular time, we contract Y_o by using the user's utility function. By maximizing utility, the target Y_o contracts to evolved goal target Y which has a single point intersection (solution) with $g(A_o)$ at point B_1 whose preimage in A_o is the car buying decision (Ignore dotted curves, B_2, B_C, Y^2, and Y^C in figure 1 for the moment).

3. GROUP DECISION MAKING: NEGOTIATIONS

Assume each decision maker (player) in a group called coalition C has worked individually with the single-user DSS procedure outlined in section 2. If the same car does not have the highest utility for all players there is a conflict. Refering to Figure 1, with two players (e.g. husband and wife), if B_1 is player 1's output (highest utility) solution and B_2 is player 2's, there is a conflict. Note geometrically that Y^C, the coalition (group) goal target -- the intersection of the goal targets Y^1 and Y^2 for players 1 and 2, respectively, i.e. $Y^C = Y^1 \Omega Y^2$ -- has an empty intersection with $g(A^C{}_o)$, the group admissible output set. In Figure 1, for simplicity $g(A^C{}_o) = \Omega g(A^j{}_o)$ for players j = 1, 2 is simply shown as $g(A_o)$. If group goal target Y^C expands, e.g. by expansion through negotiations of goal targets Y^1 and Y^2, there could be a solution at output point B_C, the intersection between expanded Y^C and $g(A^C{}_o)$.

It is clear from our discussion of Figure 1 that the search process for a solution involves contracting or expanding sets. By expansion (contraction) we mean that some new (old) points are added (dropped) to (from) a set; this expansion (contraction) does not preclude dropping (adding) some other points from (to) the set. Thus expansion/contraction involves a mapping from an original (current) set to a new set. For a group C, two sets are subject to expansion/contraction mapping. They are: (1) $g(A^C)=g(A)$, the group technologically feasible output set or more precisely the admissible set $g(A^C{}_0)=g(A)\Omega Y^C{}_0$ where $Y^C{}_0=\Omega Y^j{}_0$, and (2) the group goal target $Y^C=\Omega Y^j$. In searching for a solution, i.e., searching for a single point intersection between $g(A^C{}_0)$ and Y^C we note the following:

1. For the group goal target Y^C, higher utility aspirations (or goal demands) by players contract the target; lower utility aspirations (expressed in concession making) expand the target. Goal target expansion/contraction involves negotiations.

2. For the group admissible technologically feasible set $g(A^C{}_0)$, axioms can contract the feasible set and new technology can expand it. For example, with nondecreasing (or nonincreasing) marginal utility functions, the Pareto optimality axiom for utilities (Owen, 1982; Harsanyi, 1977; Luce and Raiffa, 1957) constrains (contracts) the feasible goal set to the upper right boundary in Figure 1 when searching for solutions. New technology cars on the market can expand the feasible set. In other words, feasible set contraction can employ solution concepts involving specification of axioms imposing agreed-upon properties on the solution; expansion can involve withdrawal of of axioms previously specified or creation of new technological inputs.

The above search focusing on goal space is paralleled in car space and utility space because of the mapping from car space to goal space to utility space (via the marginal utility functions). Figure 2 shows utility space for two players corresponding to the goal space of Figure 1.

FIGURE 2

Consider the group utility target $U^C=\Omega U^j$ where U^j is player j's utility target. In arriving at a solution at point $P_C = (u_1(B_C),$

$u_2(B_C)$), the group utility target -- initially U^C (initial) based on individual player use of the single-user DSS -- has expanded to U^C (final) intersecting the feasible set at P_C. (Ignore other items on Figure 2 for the moment). The progress of negotiations, here concession-making in goal and utility spaces and corresponding concessions in car space can be shown by a DSS either graphically as in Figures 1 and 2 or as relational data in matrix form as in Table I.

TABLE I

In Table I, car $a \varepsilon A^C{}_o = \Omega A^j{}_o$, the group joint set of a priori admissible cars, is specified by name. The goals are: Y_1 =C120 is the gasoline consumption, liters/100Km, at 120 Km/hr; y_2= space is in square meters; y_3=price is in French francs; y_4=maximum speed is in km/hr. Utilities u_1 and u_2 are the utilities of players 1 and 2, respectively. For exchanging information, the DSS could display the larger set $a \varepsilon \bigcup A^j{}_o$ which includes cars a priori admissible to at least one player. In this case, a car inadmissible for player j would be listed as "inadmissible" in the utility column u_j, but it conceivably could become admissible in the course of negotiations.

Thus, Table I shows a set of 10 cars and their corresponding goal and utility values for two players. The utility values u_1 for player 1 are taken from Jacquet-Lagreze and Shakun (1984) based on use of the single user DSS. For illustration, the utility values u_2 for player 2 are listed in reverse order of those for player 1. In row 1 of Table I, we see the goal point B_1=(10.48, 7.96, 46700, 176) of Figure 1 and utility point P_1=(.752, .383) of Figure 2 corresponding to player 1's first car choice, Opel. Similarly in row 10 of table I we see B_2=(12.26, 7.81, 68593, 182), P_2=(.383, .752) corresponding to player 2's first choice, BMW. Thus, to begin with, player 1's feasible target is defined by row 1; similarly row 10 for player 2. Concession making involves players adding additional rows to their respective targets, thereby expanding them. Given the symmetry of the situation, the solution is likely to be either Visa with P_C=(.616, .576) or Golf with P_C=(.576, .616) or a random choice between them.

In addition to the above displays, the NSS can show graphically the marginal utility functions. For output goal y_i, $u_{ij}(y_i)$ gives the marginal utility function of player j. If for a particular i the DSS shows both $u_{i1}(y_i)$ and $u_{i2}(y_i)$ on the same graphical axes, then the two players can compare, exchange information (perhaps leading towards consensus) and negotiate on their marginal utility functions. The marginal utilities u_{ij} can also be included in the relational data of Table I by inserting columns u_{i1} and u_{i2} for i=1,2,3,4, i.e., 8 columns of the u_{ij} inserted, say, between the y_4 and u_1 columns. The DSS could display the projection of the relational data of Table I onto goal y_i, u_{i1} and u_{i2} to enable the players to compare their marginal utility values for a particular goal y_i.

If players change their marginal utility functions so that they approach one another, the feasible set in utility space approaches a positive-sloping 45^o line whose highest utility point is the solution, P_{C*} (Figure 2) thus achieving consensus. Of course, U^C (initial) is readily adjusted to U^C (adjusted) to give a single point intersection at P_{C*}. In other words, in utility space, figure 2, there is a function F: P_C --> P_{C*}, P_1 --> P_{1*}, P_2 --> P_{2*} mapping the original feasible set to points along the dotted straight line with solution at point P_{C*}. U^C (adjusted) is also shown following the mapping: U^C (initial) --> U^C (adjusted).

The arrival at a common coalition utility function (through exchange of information and negotiation until players' marginal utility functions are identical) means in goal space, Figure 1, that individual players' goal targets Y^1 and Y^2 have become the same. In other words, although not drawn on Figure 1, now $Y^1 = Y^2 = Y^C$, the coalition goal target which intersects $g(A^C{}_o)$ at a solution point B_{C*} whose preiamge in car space is the car buying decision.

In addition to exchanging information and negotiating to expand targets, players can consider the use of axioms to contract the feasible region, e.g., (1) to a single solution point in utility space--in Figure 2, Nash axioms (Owen, 1982; Harsanyi, 1977; Luca and Raiffa, 1957) might give solution point P_C which is accomodated by the mapping: U^C (initial) --> U^C (final), or (2) to a constrained set of points (e.g. the Pareto optimal set might be $\{P_1, P_C, P_2\}$ in Figure 2). The latter could be followed by compromise (concessions) to select a single point from this set, e.g. P_C, or perhaps consensus leading to P_{C*} might be realized.

4. USING MEDIATOR: APPLICATION TO GROUP CAR BUYING DECISIONS

As noted in the negotiation setting overview, a human mediator supports group negotiations and he in turn is supported by the negotiation support system, MEDIATOR. The (human) mediator supports negotiations by assisting the players in a process of consensus seeking within which compromise is possible. Using MEDIATOR, the mediator assists in consensus seeking by aiding the players to build a group joint problem representation of the negotiations--in effect, joint mappings from control space to goal space (and through marginal utility functions) to utility space.

Assume each decision maker (player) in a group has worked individually with the single-user multicriteria DSS as discussed in section 2. Using PREFCALC he has established his initial individual mappings from control space to goal space (and through the marginal utility functions) to utility space. For this illustration of car buying we assume a negotiation setting between two players (e.g., husband and wife) wherein the players respond positively to the mediator's suggestion that players build a joint problem representation with the help of MEDIATOR.

For the group representation, MEDIATOR uses a common set of dimensions-- the union of the individual player dimensions--to define group (joint) control, goal and utility spaces. The evolving problem representation is shown--graphically or as relational data in matrix form--in the three group spaces, as discussed in section 3.

TABLE II

Table II shows the initial group mappings from control (car) to goal to utility spaces (ignore first and second evolved utilities for the moment). Suppose that player 2's initial individual problem representation had only three goal dimensions, say y_1, y_2, and y_3, whereas player 1's had all four goals. The common set of goal dimensions--the union--has all four goals with player 2 placing zero weight on y_4. Note that in this example there is no conflict in group control and goal space, i.e., players have the same individual problem representation in these spaces. They only differ in their representations in group utility space as shown under "initial utilities" in Table II. A look at the initial individual marginal utility functions, Figure 3, reveals the underlying preference conflict. We consider several scenarios based on play by student/faculty players.

FIGURE 3

4.1. Scenario 1

Players look at the initial utilities in Table II and perhaps, at the mediator's suggestion, at the utility functions in Figure 3. From Table II, MEDIATOR displays the car rank orders and utilities shown in Table III. The mediator asks whether players would now like to consider compromise or to seek consenses further by exchanging information. In scenario 1, we assume that either immediately or after viewing and discussing the marginal utility functions (figure 3) but not changing them, players are interested in compromise. The mediator can support compromise through use of axiomatic solution concepts (Nash, Kalai-Smorodinsky, etc.) and/or concession making procedures (Rao-Shakun, etc.) in the MEDIATOR model base --see (Shakun, 1985) for a detailed discussion.

TABLE III

As an example, the mediator can suggest concession making following conditional car target expansion. Under this procedure each player successively expands the list (target) of cars which he would be willing to accept. At stage 1, player 1's car target would be his first choice, Opel, and player 2's his first choice, M230. The intersection of these two car targets is empty. Using Table III, if players continue to expand their individual car targets by stages until a nonempty intersection is achieved, then concession making will continue to the third stage with Opel as the intersection and compromise solution.

As another attempt at compromise, the mediator can ask MEDIATOR to compute the maxmin solution concept. First MEDIATOR normalizes each player's utilities between 0 (for his last car choice) and 1 (for his first car choice). Using Table II, for each car the normalized utility for each player and minimium utility comparing normalized utilities between players are computed and shown in Table IV.

TABLE IV

The car which maximizes the minimum utility giving a maximin utility of .77 is Opel which is the maxmin solution. Thus both concession making following conditional car target expansion and the maxmin solution concept give Opel.

4.2. Scenario 2

After looking at the initial utilities in Table II and the marginal utility functions, Figure 3, players decide to discuss their marginal utility functions and modify them to those shown in Figure 4. This leads to an evolved group problem representation. Thus, in Table II the overall utilities evolve from the initial utilities to the first evolved utilities. Here consensus on a car decision--Opel--has been achieved since Opel gives each player his highest utility.

FIGURE 4

4.3. Scenario 3

In evaluating the results of scenario 2 player 2 realizes that although Opel has the highest computed utility, .66, he is not at all familiar with this car and so doesn't want to buy it. He prefers M230 or Volvo which have only utility of .65. Player 1 says that M230 and Volvo are low down on his preference list. He suggests P505 as a compromise being his second choice (with a utility of .72 and close to Opel at .74) and giving player 2 a utility of .63 which is almost high as .65 associated by player 2 with M230 and Volvo. The scenario now divides into two sub-scenarios.

In sub-scenario A, player 2 feels that player 1's proposed compromise of P505 is reasonable and accepts it. In sub-scenario B, player 2 says that, although player 1's compromise suggestion of P505 is reasonable in the light of the first evolved utilities, he now realizes something is bothering him about the problem representation. It does not include the nationality (country of manufacture) of the cars. He would like to assign a preference weight to German cars. Thus, player 2 has introduced a fifth goal dimension, y_5, car nationality, to the group problem representation--imagine a column for goal y_5 after goal y_4 in Table II where German cars (Opel, Golf, M230, BMW) are assigned a nominal value of, say, 2 and all other cars a value of 1.

Using PREFCALC, player 2 modifies his marginal utility functions to those shown by the dotted lines in Figure 5. He places a weight of .1 on car nationality, y_5, and modifies the relative weights on the other criteria so that the sum of the weights equals 1. Note with PREFCALC's utility normalization the criterion weight equals the marginal utility at the most preferred goal value considered. Player 1 places zero weight on car nationality, y_5 so that his marginal utilities in Figure 5 are in effect the same as in Figure 4.

FIGURE 5

The marginal utility functions in Figure 5 give overall utilities shown under "second evolved utilities" in Table II. For player 1, there is no change--his second evolved utilities are the same to his first evolved utilities. For player 2, the second evolved utilities show the German Opel with utility .70, the German M230 with .70, the Swedish Volvo with .60 and the French P505 with .57. Player 2 still rules out the Opel as an unfamiliar car. He now prefers the M230 (utility .70) over the Volvo (.60) whereas before introducing car nationality they were tied at .65 each. While before the P505, at utility .63 was close to player 2's first choices of M230 and Volvo each with utility .65, now the utility gap between P505 (utility .57) and the first choice M230 (utility .70) is large so that player 1's compromise suggestion of P505 represents a large utility drop for player 2 from his first choice. The mediator asks MEDIATOR to compute the maxmin solution. Using Table II, for each car MEDIATOR computes the second evolved nomalized utility for each player and, comparing these, the minimim utility--see Table V.

TABLE V

The maxmin solution is Opel. With player 2 ruling it out, player 1 argues that P505, as the maximin solution over the remaining cars, is fair. Besides he would have to drop to low utility levels of .45 and .44 (Table II) if he were to consider M230 or Volvo, respectively. Player 2 is convinced and accepts P505 as the solution.

5. SYSTEM ARCHITECTURE FOR MEDIATOR

In this section, we describe software requirements and a system architecture for MEDIATOR. Basically, MEDIATOR integrates a collection of software components used by the players and the human mediator through the use of a shared database. The need for analyzing such components in a DSS, in addition to the description of operational research models, arises from two sources. Any DSS must offer a user-friendly interface and efficient data access. Otherwise, it will not be used by computer-naive decision makers. More specifically, however, a multi-person DSS like MEDIATOR must also facilitate and structure the communication among the players and with the human mediator.

One approach to implementing such a communication facility is a direct message exchange subsystem based on electronic mail (Bui and Jarke, 1984). Another approach --the so-called decision room-- leaves the responsibility for player communication outside the system: the players are assembled in a single room and can communicate without computer aids (Huber, 1982).

In contrast to these two methods, our approach is database-centered. The database-centered approach was introduced for single-user DSS by Donovan (1976), and extended to hierarchically organized distributed DSS by Jarke (1981, 1982). This paper extends the approach further to Negotiation Support Systems. MEDIATOR achieves communication mostly through the sharing of data stored in a common database. This database would be typically located on a mainframe or on a separate file server accessible by all players and by the mediator. It contains base data underlying the decision-making process as well as intermediate results of the negotiation process -- the sequence of group joint problem representations.

The rules of communication (also called communication protocols (Tanenbaum, 1981)) are implemented through granting different access rights to players and mediator. In the following subsections, we first motivate this approach by a requirements analysis, and then provide a more detailed technical description. As a running example, we shall use once more the two-player car buying application.

5.1. Systems Requirements for MEDIATOR

MEDIATOR is designed to provide user-friendly interfaces, efficient data and model access, and structured communication facilities to both the players and the mediator. Systems requirements for MEDIATOR can be grouped into two categories.

The first class of requirements is derived from the method itself. Since negotiation is viewed as an evolutionary process of information exchange leading to consensus or compromise, the system has to provide efficient support for interactive use by decision makers and mediators with limited computer skills. Moreover, the system has to offer at least two kinds of representations for information display. A relational database system must support the matrix representation needed in the detailed display of criteria and utilities vs. alternatives (see Table II). Additionally, the systems can present data graphically (e.g., piece-wise linear marginal utility functions, see Figure 3). Finally, the system must be able to analyze and present the consequences of changes in control space, goal space, and utility space.

The second class of systems requirements results from some implicit assumptions in the proposed method. MEDIATOR has to satisfy these assumptions prior to the actual negotiation procedure. The main assumption is that of the idea of building a <u>group joint problem representation</u> is accepted by players. The remainder of this subsection investigates the consequences of this assumption for MEDIATOR's design in detail. The assumption has three facets: a jointly acceptable database of underlying facts, jointly acceptable definitions of alternatives, and mutually understood definitions of criteria and preferences.

<u>Jointly acceptable database</u>. The method assumes that the players agree on a common underlying set of facts about the domain of decision. The example of arms control negotiations shows that such an agreement may be very difficult to reach. The players may not even agree on a common scope of alternatives for a particular negotiation (e.g., strategic vs. Euro-strategic vs. space weapons). Moreover, disagreement on the underlying facts is almost certain. Therefore, MEDIATOR allows the players to agree that each will use their own data separately, i.e., an agreement exists that the players cannot agree on a common set of data. This version of the assumption may be the only way to get negotiations started if there is deep distrust among the players -- witness again arms control negotiations.

Jointly acceptable alternative definitions. This assumption requires that the players agree on a common definition of the dimensions (not necessarily feasible regions) of the control space. We distinguish syntactic and semantic disagreements that have to be resolved in pre-negotiations.

Syntactic disagreements involve a different understanding of terms. For example, two players may give the same alternative a different name (synonyms) or two different alternatives the same name (homonyms). It is very important and usually not too difficult to resolve such misunderstandings.

Semantic disagreements involve a different partitioning of the control space. Different partitioning may mean varying degree of detail, or it may mean completely different dimensions. Disagreement often results from differences in knowledge, or from basically inconsistent views of the problem. The former is easier to resolve than the latter.

For example, in a car-buying decision one player may distinguish cars by their engine type, another one by their make, yet another one by their make, model, and version. The obvious solution is to define alternatives by combining all suggested partitionings. However, this may lead to an intolerably large number of alternatives. It is the task of the mediator to assist the players in defining a mutually understandable and acceptable, yet manageable set of alternatives. MEDIATOR can help in this task using certain concepts of database theory (see Section 5.4).

Mutually understood criteria and preference definitions. While the previous assumptions concerned the control space, this one involves possible misunderstandings in the goal space definition. Of course, the method does not require players to use the same criteria. However, it is important that the mediator and his support system, MEDIATOR, understand the meaning of criteria, in order to make useful suggestions. Again, there may be syntactic or semantic problems. The former involve the naming problems of synonyms and homonyms, whereas the latter could mean different units of measure, or different ways to compute criteria from the available data.

As an example, one player may compute a criterion "space" in square meters, whereas another one uses "space" or its synonym "roominess" but measures it in cubic feet. On the semantic side, both players could define "space" as the size of the inner sitting room; then, it would be desirable to merge the two criteria. Alternatively, one of the players may use "space" for the outer size of the car. This could lead to the apparent paradox that one player tries to minimize "space" while the other is maximizing it. Clearly, it is appropriate here to rename and separate the criteria.

Each player defines preferences on criteria, e.g., a utility preference measure. However, preference measures used by players need not be the same.

As a consequence of these assumptions, MEDIATOR supports a <u>two-phase negotiation process</u>. In the first phase, called <u>view integration</u>, the human mediator is supported in achieving a joint problem representation in the three steps of: database selection, alternative definition, and criteria and preference definition. Upon successful completion of this phase, the second phase, called <u>negotiation</u>, proceeds as described in sections 3 and 4.

5.2. Software Capabilities and Components

An architecture for the MEDIATOR DSS should offer some software capabilities to support the systems requirements described in section 5.1. We shall first review the major components of single-user DSS and then propose a specific architecture for MEDIATOR.

Each player and the mediator employ a single-user personal DSS which has the traditional three components of model management, data management, and dialog management (Sprague and Carlson, 1982). For MEDIATOR, this single-user DSS is a data-based version of PREFCALC (Jacquet-Lagreze, 1985) for the players and an enhanced version for the mediator.

The dialog manager is responsible for effective interaction between the DSS and its users, namely each player and the human mediator. It provides menu management, screen composition, and graphics as well as relational representation facilities (Jarke et al., 1984).

The model manager consists of executable modules together with modelling language facilities and execution management. In particular, the negotiation models in the mediator DSS allow mappings of user changes (or adaptations) in all three spaces (control, goal, and utility space).

The data manager accesses and maintains the user's private as well as the jointly acceptable mainframe databases. It contains a standard DBMS with enhanced data dictionary and view management facilities (Jarke et al., 1984; Jelassi, 1985; Jelassi et al., 1985]. The "data dictionary" stores metadata such as alternative definitions, criteria definitions, function definitions, and units of measure. A "generalized view processor" helps the user define their personal customized view to the underlying database. In particular, alternatives and criterion values can be derived automatically from the stored database records and their attributes.

For n players, there are n+1 DSS of this nature. In addition, group decision (Bui and Jarke, 1984) or negotiation support systems require a communications manager to integrate the single-user DSS (Figure 6). In MEDIATOR, this is accomplished in the following manner (Figure 7).

Each player and the mediator retain their private databases typically stored on a personal computer. The jointly acceptable data are stored in a common database located on a mainframe or larger minicomputer and accessible by all the personal computers. The model/method base may contain different tools for each player but they share the PREFCALC method. Conceptually, this method could be stored in a common model base associated with the common database. From an implementation viewpoint, it is more efficient to have copies on each microcomputer, in order to avoid communication delays.

FIGURE 6

FIGURE 7

After establishing their individual preferences using single-user PREFCALC, players transfer their definitions of alternatives and criteria, and their matrix and utility function representation to the common database. Each player occupies a private section of that database which can be only accessed by himself and by the mediator. The mediator will then start the process of integrating these personal problem problem representations into the group joint problem representation.

Once this is accomplished, the joint problem representation is stored in the publicly accessible area of the common database. From then on, the "official" negotiation will only work with the joint representation. The players are free to continue using their local representation and other decision support tools for personal deliberations.

From a computer science point of view, MEDIATOR's design poses the following research questions. How do we provide:

1. efficient database access for each player?

2. database and model base facilities to support the mediator?

3. communication between players and mediator?

4. user interfaces for players/mediator?

The following two subsections address our solutions to these problems, first for the individual player DSS, then for the mediator DSS and its communication with the player DSS.

5.3. DSS of the Individual Players

The DSS of the individual players are based on a stand-alone version of PREFCALC (Jacquet-Lagreze, 1985). PREFCALC is a single-user DSS implemented on a personal computer; its underlying algorithms have already been described in Section 2. The method assumes its input to be stored in a relational format where the records (rows of the matrix) correspond to alternatives and the columns to criteria. The table entries are criterion values.

In earlier work (Jarke et al., 1984; Jelassi, 1985; Jelassi et al., 1985), we enhanced the system with user-friendly capabilities to (a) access external mainframe databases, (b) define alternatives from sets of records rather than from single records, and (c) use composite criteria computed from stored attributes by user-defined or selected functions. The following description is based on (Jarke et al., 1984).

Figure 8 shows how the PREFCALC input is generated from the database. The method starts from a set of ALTERNATIVES, each characterized by a number of properties or attributes. For example, in a car-buying example the alternatives are types of cars and the base relation would contain relevant attributes such as "maximum speed", "fuel consumption at speed 120 km/h", etc. However, the stored relation may differentiate the type of car in many more classes than needed for the decision. Thus, a decision alternative may correspond to a set of several database records.

Some (but not necessarily all) of the attributes of the database records may be important to a particular user's decision problem. The attributes in this subset are used as CRITERIA. However, the user may also wish to derive more complex criteria from the stored attributes, or ask for a presentation in different units of measure (e.g., horse powers instead of kilowatts). Therefore, some computation may be necessary to derive criteria from the stored data.

Both tasks (alternative and criteria definitions) are accomplished by the afore-mentioned "generalized view processor" in conjunction with a menu interface generator. The resulting user view is called the DECISION MATRIX. In order to create a decision matrix, the user has to define --through a sequence of menus-- how decision alternatives and decision criteria are derived from the underlying database.

<u>Alternatives</u> are defined in two steps. In the <u>data staging</u> step, the user selects a subset or CATEGORY of alternatives to be considered. For example, in a car buying application, the user may be interested only in trucks but not in other types of cars. Data staging extracts data from one or more mainframe databases and constructs from them a single selected subrelation (the CATEGORY) on which all further processing will be performed, using the microcomputer DSS database. Users may either choose from a menu of category names defined in the CATEGORY DEFINITIONS section of the data dictionary, or define their own category via a distributed database query.

In the <u>grouping</u> step, the user chooses a grouping of database records within the CURRENT CATEGORY relation such that each group constitutes an ALTERNATIVE. Groups are defined by common values of certain attributes (the ALT-NAME). For example, some player may be interested in distinguishing cars only by their make and model, but not by details such as number of doors, engine power, etc.

Criteria are derived from attributes of the database records. In the simplest case, an attribute value can directly serve as a criterion value (e.g., maximum speed). Frequently, however, the criterion value may be a function of one or several attribute values. For example, a criterion "consumption" may be defined as the average of the stored database values of fuel consumption in the city and on highways.

Moreover, whenever alternatives correspond to groups of records rather than to single records, criterion values must be based on aggregate functions over these records (e.g., average, minimum, maximum, forecast for next year). The data dictionary contains a library of such CRITERIA DEFINITIONS from which the user can choose the CURRENT CRITERIA, using a menu of CRIT-NAMES. (Of course, a more sophisticated user can also add functions to the library.)

Finally, the combination of alternative definitions (grouping) and criteria definitions (computations) allows the derivation of criterion values for alternatives (CRIT-VALUE) from the database. All of the above operations can be performed within an extended relational database framework discussed in detail in (Jarke et al., 1984; Jelassi 1985).

Figure 9 gives an example of decision matrix construction for player 2 in the example of section 4. This player wants to use the car mostly in the city and is therefore interested in a spacious but not too expensive car with little consumption in the city. This player ignores other known car characteristics (e.g., speed, number of doors, etc.); moreover, the database consulted by player 2 contains only information about the DIN consumption. Player 2 is not interested very much in the differences among different versions of a car but rules out sports versions (e.g., the Golf GTI in the example table). Note, that the criterion, space, must be computed from the stored values of car length and width.

FIGURE 8

FIGURE 9

After the construction of the input matrix, the CATEGORY relation will be needed only if the multi-criteria decision method proposes the inclusion of a new criterion in order to resolve apparent inconsistencies in user preferences. Otherwise, the method proceeds to construct utility functions through aggregation and disaggregation of preferences, as described in Section 2 (see also Jacquet-Lagreze and Shakun (1984)).

5.4. DSS of the Mediator

In the previous subsection, it was demonstrated that a minor extension of the relational model of databases (Codd, 1970; Ullman, 1982) is sufficient to support the single-player data preparation process for the multiple criteria DSS, PREFCALC. In this subsection, this result will be extended. Relational operations, enhanced by redefinitions of terms, can also efficiently support the <u>view integration</u> phase of mediation. The discussion will follow the same sequence (database selection, alternative definition, criteria and preference definition) as before. Subsequently, we review the mediation support tools used in the negotiation phase.

The first task in establishing a group joint problem representation is the choice of underlying databases upon which the definition and evaluation of alternatives can be based. In general, the mediator will have to start with the union of all such databases as far as they are made available to him. In this case, it is crucial to establish the logical relationships between different sources of information about the same entity. For example, one catalog may refer to cars by make, model, version, etc., another one by product number. A translation table must be used to permit a join among the different entity identifiers. This task may be complicated if no 1-to-1 mapping exists; in this case, the least common superset must be constructed to permit mappings. In most intraorganizational negotiations, however, the database selection step will be simple because players access the same organizational (mainframe) databases to begin with.

The database selection step establishes a logical view of the mainframe databases as a large "universal relation" (Ullman, 1982). From this, the group joint CATEGORY relation can be easily defined. As mentioned in Section 3, either the union (if there are few feasible alternatives) or the intersection (if there are many group-feasible alternatives) of the individual CATEGORY relations can be chosen as the joint representation. In a relational query language, this means that the individual CURRENT CATEGORY definitions are simply conjunctions and disjunctions of restriction predicates.

Next, the individual ALTERNATIVES definitions must be integrated. Since the group CURRENT CATEGORY relation contains all the attributes of the individual CURRENT CATEGORY relations, grouping will simply use all grouping attributes of the individual ALTERNATIVES definitions simultaneously. Unfortunately, this solution may result in very long alternative names and a large number of alternatives to be considered. For example, if one player distinguishes cars by their maximal speed, another one by their make, and a third one by make, model, and version, a particular group alternative could be named: "180-190 km/h, Mercedes(user2), Mercedes(user3), M190, E".

The concept of a "functional dependency" as developed by database theory (see, e.g., (Ullman, 1982)), can be exploited to simplify this naming problem. A database attribute is called functionally dependent on a set of other database attributes if for each combination of values of these attributes, the dependent attribute can assume at most one value. Obviously, each attribute is functionally dependent on itself. Therefore, the first simplification is to unify the two occurences of "Mercedes" (provided both players mean the same thing -- this, the mediator DSS can test by looking at the databases and groupings used by both players).

Assume that the database schema in the data dictionary also states that maximal speed is functionally dependent on make, model, and version; i.e., for each version of a car, there is only one maximal speed. In this case, MEDIATOR can automatically simplify the group joint alternative grouping to make, model, and version. The simplified example alternative name then becomes just "Mercedes M190 E". For the sake of player 1, maximal speed will be retained as a criterion (but not as an alternative name) in the decision matrix.

If such automatic simplification proves insufficient, the human mediator will make other suggestions. One option was already mentioned: reducing the set of alternatives by presenting only the intersection-feasible ones. (In a many-player situation, the requirement of mutual feasibility can be relaxed to, e.g., "acceptable to at least 50%", etc.). As another option, consider the case of different degree of specialization among the players. If the large number of alternatives is created by varying degree of detail (i.e., one alternative definition is a subset of the other), the mediator may suggest postponing the decision about detailed alternatives until after a preselection of "good" higher-level alternatives.

Next, the group joint <u>criteria</u> definitions (columns of the group decision matrix) must be established. This step starts formally by executing a relational join operation over all the player's decision matrices understood as derived relations. The join columns are the alternative names as established in the previous step. This will result in a preliminary version of the group decision matrix in which the alternative names are common to all players but all criteria are disjunct. This means: players are assumed to assign weights of 0 to all columns but those steming from their own decision matrix. If there are n players each with m criteria, there will be nm criteria in the preliminary group decision matrix.

The mediator will now try to collapse criteria that appear more than once and to unify similar criteria. In section 5.1, we have already illustrated the pitfalls. The function definitions stored in the CRITERIA DEFINITION section of the players' data dictionaries are the major MEDIATOR tool to assist in criteria integration. If the function definitions of two criteria are equal, proportional (possibly different units of measure), or reciprocal there is a good chance that two criteria mean the same thing (respectively one is the negation of the other), even if they have different names.

By contrast, if criteria have the same name but differ significantly in their function definition, they may mean different things. An indicator of such semantic disagreements may be the marginal utility functions of the players. Therefore, MEDIATOR offers overlay of marginal utility curves for any pair (or small group) of players. Usually, one would expect that utility curves of players for the same criterion differ in weight and steepness but they will rarely cross (one monotonically increasing, the other one decreasing). If they do cross, this may mean severe value disagreements, or simply misunderstanding of terms.

Human mediator intervention to resolve such questions remains necessary even in the presence of an NSS. Looking at the function definitions alone, it may be very difficult or even impossible to prove or disprove equivalence of functions (even though the form of the function definitions in our SQL extension are very standardized). Therefore, the integration step will only be supported but not completely automated.

Once the alternatives and criteria have been integrated as far as appropriate, MEDIATOR constructs the group joint problem representation using the player's preference information.

As an example, consider the view integration process preceeding the negotiations described in section 4. In contrast to player 2 who wanted a city car, player 1 wants to use the car mostly for business trips. He is therefore initially interested in highway consumption, high speed, a limited price, and much space. His database has more detailed information on consumption at various speeds but is otherwise identical to that of player 2. In the first step of view integration, both players agree to use the intersection of the two sets of acceptable cars, ruling out sports versions which were not acceptable to player 2. Both players name one of their criteria "consumption" but a review of the criteria definitions by MEDIATOR reveals that one means the DIN consumption, the other one the highway consumption. Since both measures are highly correlated, and player 1's business trips will account for most of the kilometers anyway, the players agree in the criteria integration phase to work on the basis of highway consumption, and to call this criterion C120. The criterion, space, has identical definitions in both decision matrices and will simply be merged. Based on this information, both players reconsider their utility evaluations and come up with the initial group joint problem representation shown in Table II.

An interesting design question for negotiation support systems in general arises after the view integration. In which form should the result be fed back to the players? MEDIATOR's answer was chosen for reasons of simplicity: the same view of the joint problem representation is offered to all players.

Alternatively, one could try to adapt the joint representation to each player's language, i.e., trying to attempt to translate the global view back to the individual view as far as possible. Database theory has shown the impossibility to do this automatically; however, practical ways around this problem have also been devised (Furtado and Casanova, 1985). However, these methods are very difficult to implement. Moreover, it is not clear from an application standpoint whether this solution is even desirable -- consider, e.g., a second channel of communication among the players which might result in considerable confusion unless a common language is enforced.

The remaining MEDIATOR tools support the human mediator in the actual negotiation phase as described in Section 3. The implementation of the comprehensive example presented in section 4 is based on the group joint problem representation as just developed. Here, we just summarize the major software tools grouped by the problem space in which they apply. We do not consider behavioral tools such as Delphi, NGT, etc.; for an overview, see (Bui and Jarke, 1984), (DeSanctis and Gallupe, 1984), or (Huber, 1984).

MEDIATOR allows the human mediator to perform what-if analyses of possible suggestions he might make. Before, e.g., suggesting that players should lower their utility threshold, the mediator must make certain that this will generate additional alternatives for discussion. Otherwise, the players will feel that they made a concession for nothing and the climate of the negotiation may deteriorate.

In the control space, the relational query language offers the option of including or excluding sets of alternatives from consideration by certain attribute or criterion values. The mediator can apply such queries either directly at the level of the DECISION MATRIX (usually to make it smaller) or at the level of the group CURRENT CATEGORY relation (usually to increase the set of feasible alternatives that appear in the DECISION MATRIX). The same technique may be applied either to the reference set of alternatives presented to the users, or to the CATEGORY as a whole.

Changes in the goal space involve a redefinition of the set of CURRENT CRITERIA, either by adding or by deleting CRIT-NAMEs from the DECISION MATRIX. Since the mediator DSS has full access to the group CURRENT CATEGORY relation, such changes can usually be effected without re-computing the whole DECISION MATRIX or re-accessing the mainframe databases. Suppressing a criterion may even be done by the dialog manager, without changing the internal representation at all.

Before a change in the goal space is made, the mediator will frequently be interested in the importance of criteria. Would dropping a criterion change the ratings? Is the ranking by a particular criterion inconsistent with the overall utility ranking of alternatives? To answer such questions, certain display techniques for relational data are employed, most prominently alternative ranking by some criterion.

The idea of ranking alternatives is also used to answer what-if questions in the utility space. Since player utilities are simply additional attributes of the DECISION MATRIX relation, they can be easily used as sorting criteria. Alternatives can be presented ordered by a particular player's utility either down to a certain rank ("present the five best alternatives for player 1") or down to a certain minimal utility ("present all alternatives above normalized player 1 utility .50"). This facility is already offered by PREFCALC for a single player. See section 4 for an example of the display of two players' utility rank orders in concession making. For more than two players, it may also make sense to display the aggregated utilities of certain coalitions.

The second representational tool at the utility level involves the overlayed marginal utility curves. What-if questions allow the mediator to vary the weights and numbers of linear pieces of each player tentatively. As stated before, all of these what-if studies are intended to prevent having the player's agree to useless concessions and redefinitions that would unnecessarily delay the decision process and destroy the trust of players in the mediator's (and in the DSS's) abilities. An underlying assumption made in this context is that players are actually interested in a fast decision while preserving their interest -- an assumption that is usually justified in intraoriganizational negotiations but may not hold in other cases.

In summary, the tools described here are mostly database and display tools, related to the algorithms presented in Section 3. Other mathematical or behavioral tools may also be needed but their discussion would go beyond the scope of this paper; see (Shakun, 1985) for a detailed presentation of axiomatic and concession-making procedures.

Once the human mediator has formed an opinion from the what-if analyses, he broadcasts messages to the players either directly or by notifying them of proposed changes tentatively made on the group problem representation. Broadcasting messages keeps the mediation process as unbiased and open as possible. (Our underlying hypothesis is that players may discontinue the use of MEDIATOR if it means loss of information to them.) Examples of such changes are: changes in the joint set of alternatives; introduction of new criteria or changes in weights; areas where concessions of players may reduce differences in opinion; and changes in utility values.

6. CONCLUDING REMARKS

The system design for MEDIATOR supports building a group joint problem representation (view integration). Negotiation involves the evolution of this problem representation--consensus seeking -- within which compromise is possible. At any stage of problem representation, the mediator can support compromise through use of axiomatic solution concepts and/or concession-making procedures in the MEDIATOR model base.

With systems like MEDIATOR we are moving towards decision support sytems for multiplayer, multicriteria, ill-structured, dynamic problems, thus implementing in a decision support context the methodology of evolutionary systems design (ESD)--policy making under complexity.

References

1. Bonczek, R.H., Holsapple C.W., and Whinston, A.B., 1981. Foundations of Decision Support Systems, Academic Press, New York.

2. Bui, X.T., and Jarke, M., 1984. A DSS for cooperative multiple criteria group decision making, Proceedings 5th International Conference on Information Systems, Tucson, Az, 101-113.

3. Codd, E.F., 1970. A relational model of data for large shared data banks, Communications of the ACM 13:6, 377-387.

4. DeSanctis, G., Gallupe, B., 1985. Group decision support systems: a new frontier, Data Base 16:1, 3-10.

5. Donovan, J.J., 1976. Database system approach to management decision support, ACM Transactions on Database Systems 1:4, 344-369.

6. Furtado, A., and Casanova, M., 1985. View updates, in Query Processing in Database Systems, W.Kim, D.Reiner, D.Batory, Eds., Springer-Verlag, New York.

7. Giordano, J.L., Jacquet-Largeze, E. and Shakun, M.F. 1985. Decision Support System for Design and Negotiation of New Products, Graduate School of Business Administration, New York University, New York, NY.

8. Harsanyi, J.C. 1977. Rational Behavior and Bargaining Equilibrium in Games and Social Situations, Cambridge University Press, Cambridge.

9. Huber, G.P., 1982. Group decision support systems as aids in the use of structured group management techniques, DSS-82 Transactions, 96-108.

10. Huber, G.P., 1984. Issues in the design of group decision support systems, MIS Quarterly 7:3, 195-204.

11. Jacquet-Lagreze, E. 1985. PREFCALC, Version 2.0. EURO-DECISION, BP 57,78530 Buc, France.

12. Jacquet-Lagreze, E., and Shakun, M.F., 1984. Decision support systems for semi-structured buying decisions, European Journal of Operational Research 16:1, 48-58.

13. Jacquet-Lagreze, E., and Siskos, J., 1982. Assessing a set of additive utility functions for multiple criteria decision making: the UTA method, European Journal of Operational Research 10:2, 151-164.

14. Jarke, M., 1981. Ueberwachung und Steuerung von Container-Transportsystemen, Gabler, Wiesbaden, West Germany.

15. Jarke, M., 1982. Designing decision support systems: a container management example, Policy Analysis and Information Systems 6:4, 351-372.

16. Jarke, M., Jelassi, M.T., and Stohr, E.A., 1984. A data-driven user interface generator for a generalized multiple criteria decision support system, in Proceedings IEEE Workshop on Languages for Automation, New Orleans, 127-133.

17. Jelassi, M.T., 1985. An Extended Relational Database for Generalized Multiple Criteria Decision Support Systems, Ph.D. Dissertation, Department of Computer Applications and Information Systems, New York University.

18. Jelassi, M.T., Jarke, M., and Stohr, E.A., 1985. Designing a generalized multiple criteria decision support system, Journal of MIS 1:4, 24-43.

19. Keen, P.G.W., and Scott-Morton, M.S., 1978. Decision Support Systems: An Organizational Prespective, Addison-Wesley, Reading, MA.

20. Luce, R.D. and H. Raiffa. 1957. Games and Decisions, John Wiley & Sons, New York.

21. Owen, G., 1982. Game Theory, 2nd ed., Academic Press, New York.

22. Shakun, M.F., 1981a. Formalizing conflict resolution in policy making, International Journal of General Systems 7:3.

23. Shakun, M.F., 1981b. Policy making and meaning as design of purposeful systems, International Journal of General Systems 7:4.

24. Shakun, M.F., 1985. Decision support systems for negotiations, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Tucson, Az, November 1985. Also in Shakun (1986).

25. Shakun, M.F., 1986. Evolutionary Systems Design: Policy Making Under Complexity, Holden-Day, San Francisco.

26. Sprague, R.M., and Carlson, E.D., 1982. Building Effective Decision Support Systems, Prentice Hall, Englewood Cliffs, N.J.

27. Tanenbaum, A., 1981. Network protocols, ACM Computing Surveys 13:4, 453-489.

28. Ullman, J.D., 1982. Principles of Database Systems, 2nd edition, Computer Science Press, Rockville, MD.

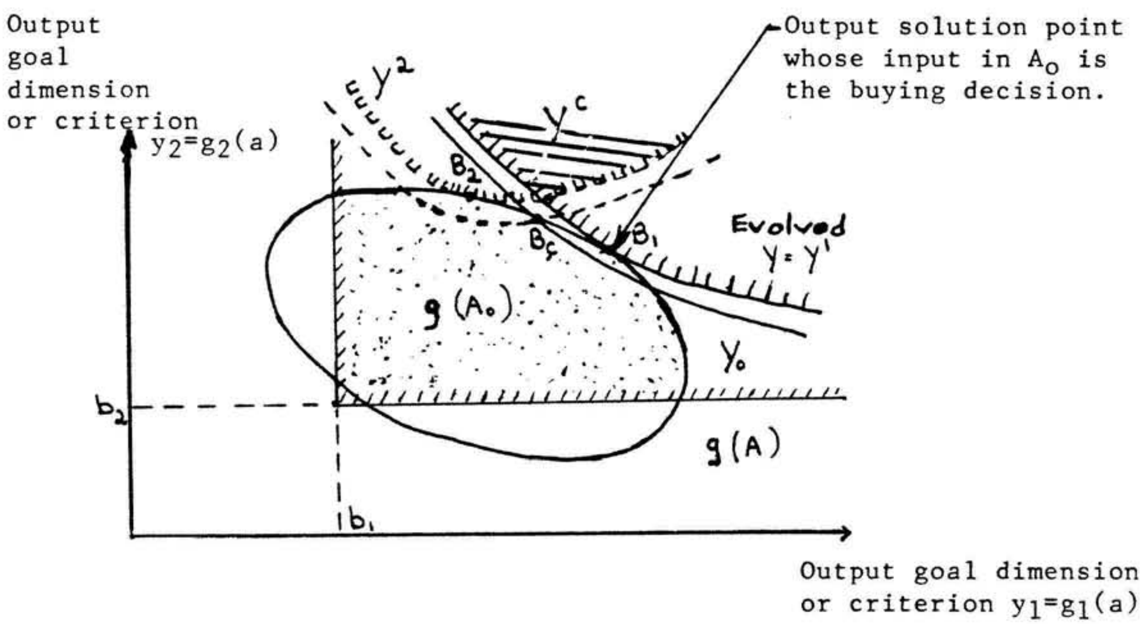

Figure 1. Output, Goal or Criteria Space

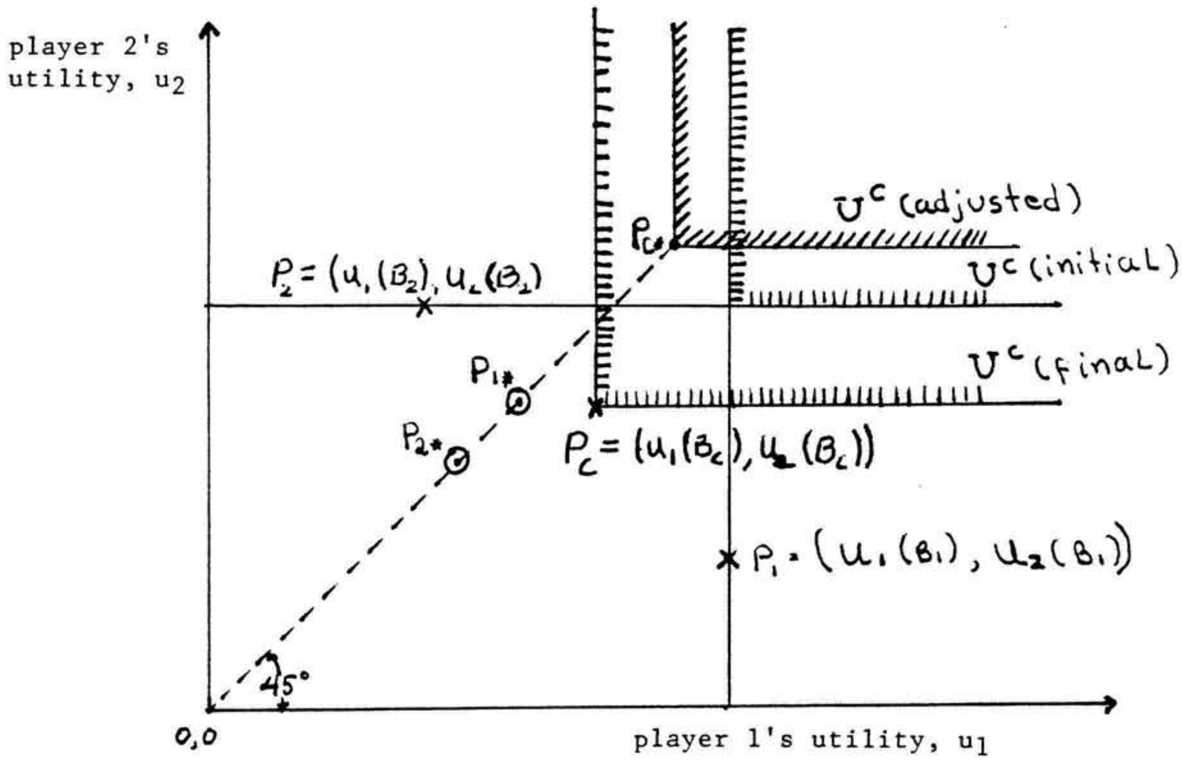

Figure 2. Utility Space For Two Players

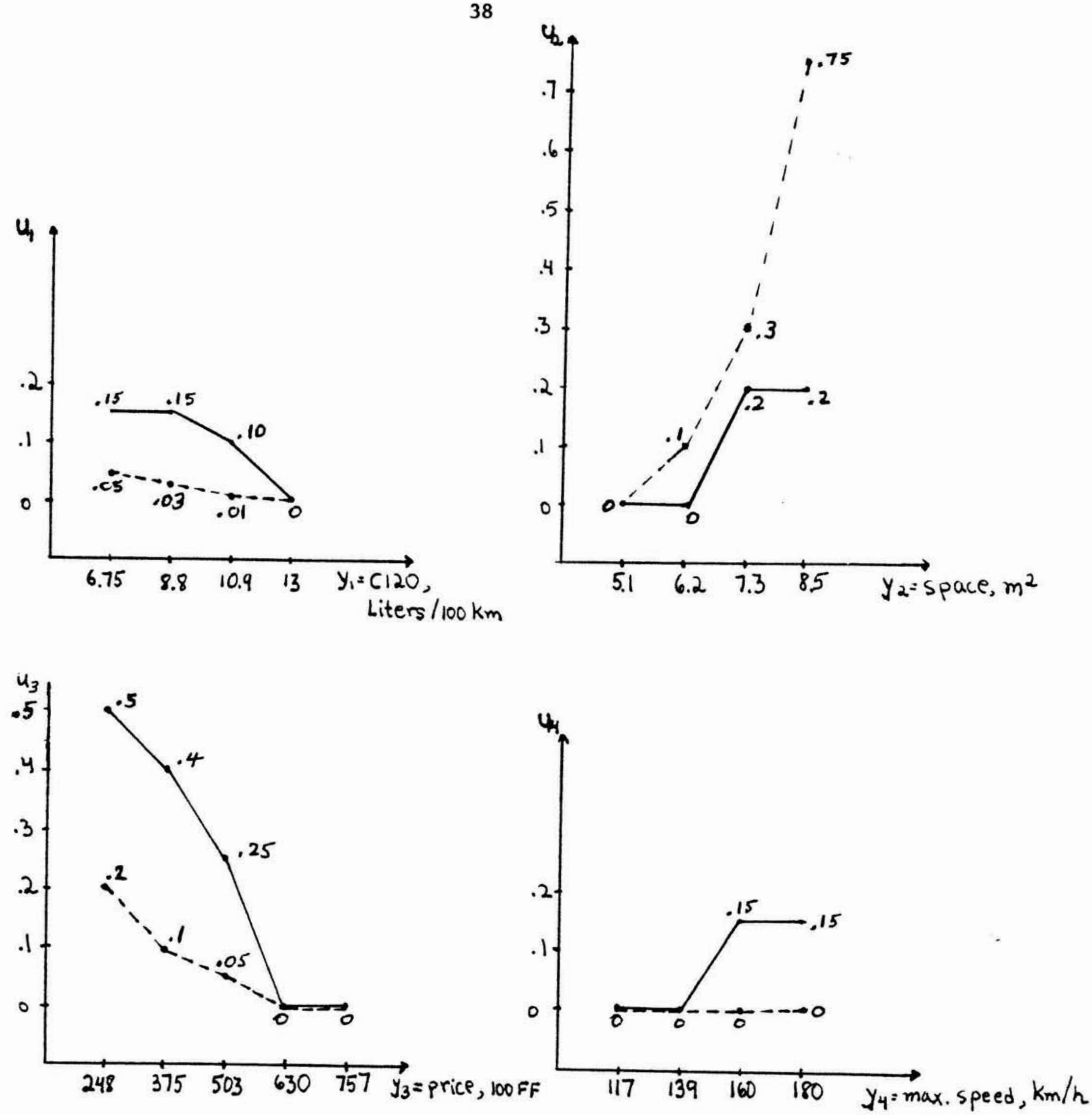

Fig. 3 Initial Marginal Utility Functions for Player 1 (Solid) and for Player 2 (Dotted).

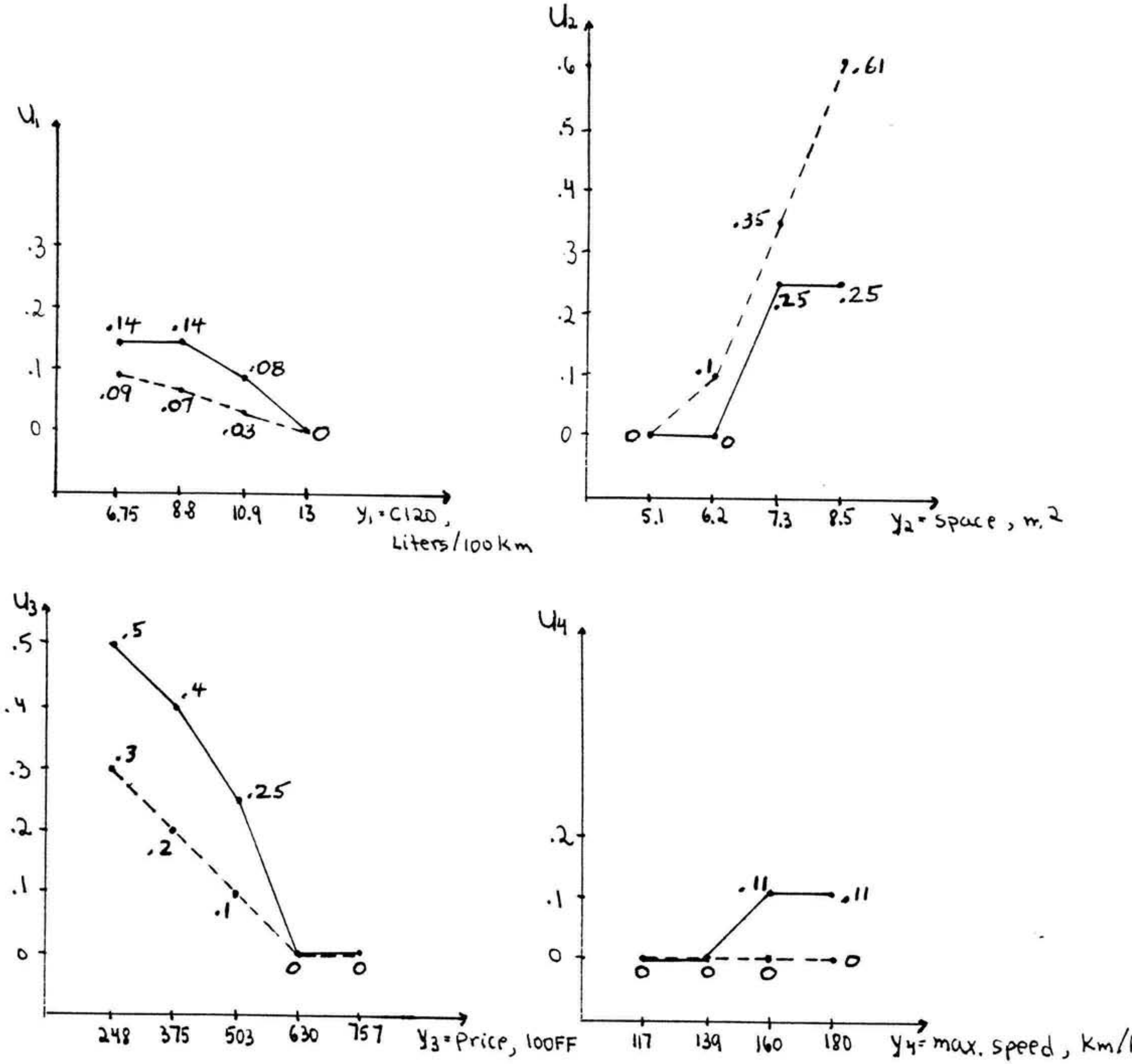

Figure 4. First Evolved Marginal Utility Functions for Player 1 (Solid) and for Player 2 (Dotted).

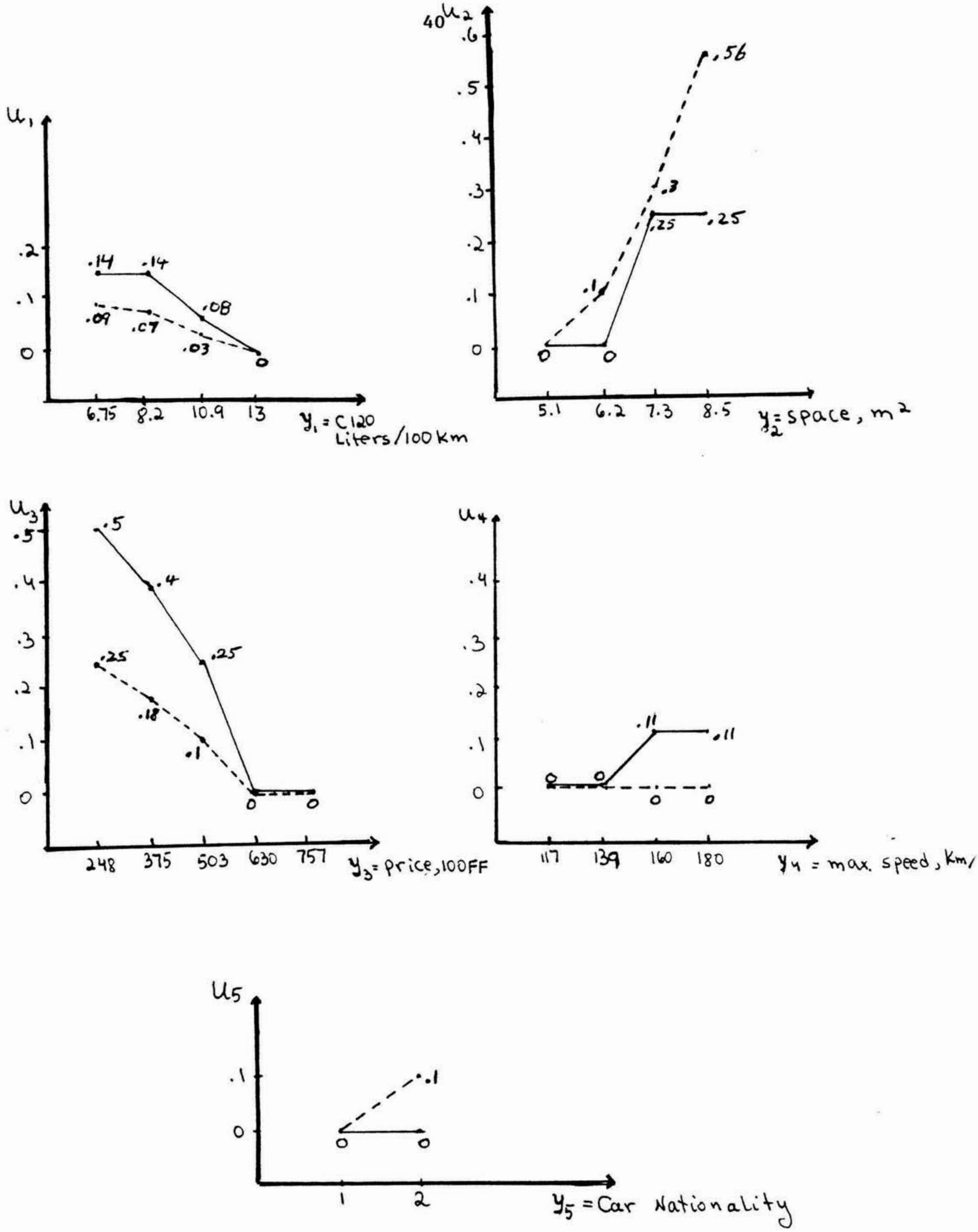

Figure 5. Second Evolved Marginal Utility Functions for Player 1 (Solid) and for Player 2 (Dotted).

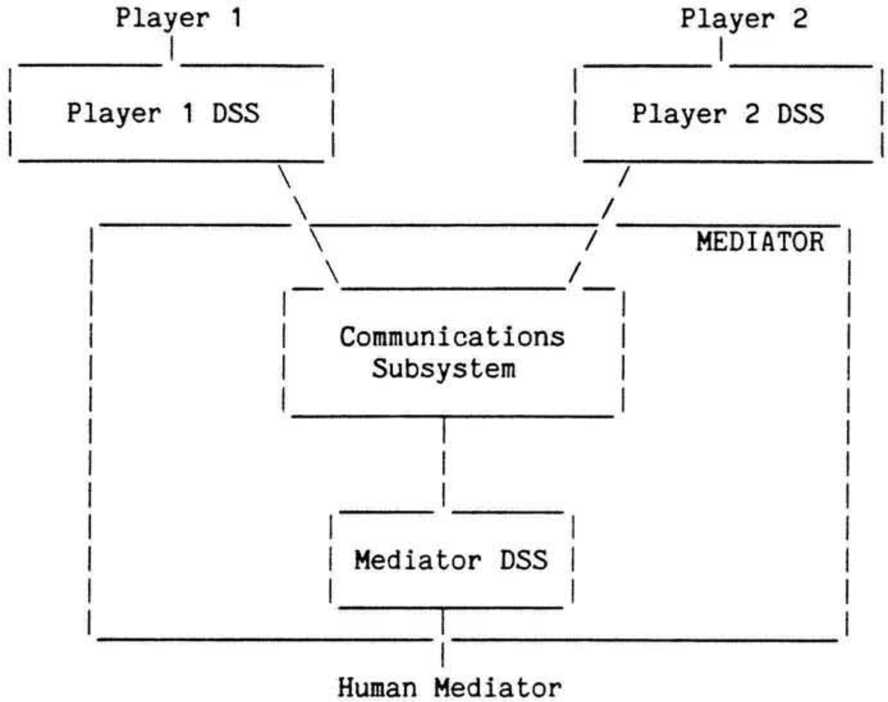

FIGURE 6: General NSS Structure -- Players' and Mediator's DSS

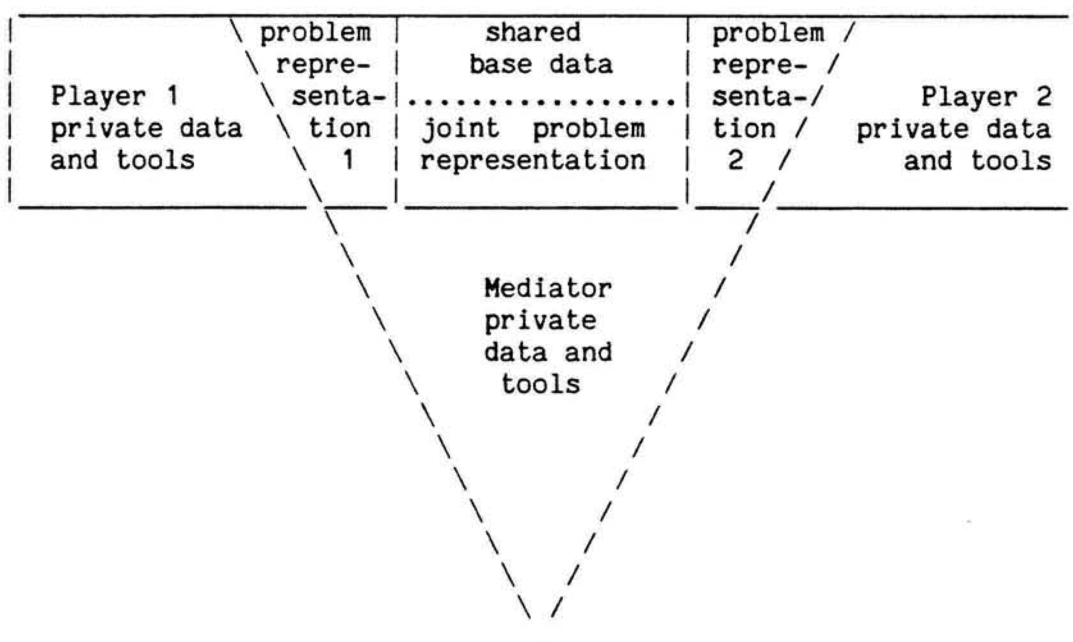

FIGURE 7: MEDIATOR Design -- Communication through Data Sharing

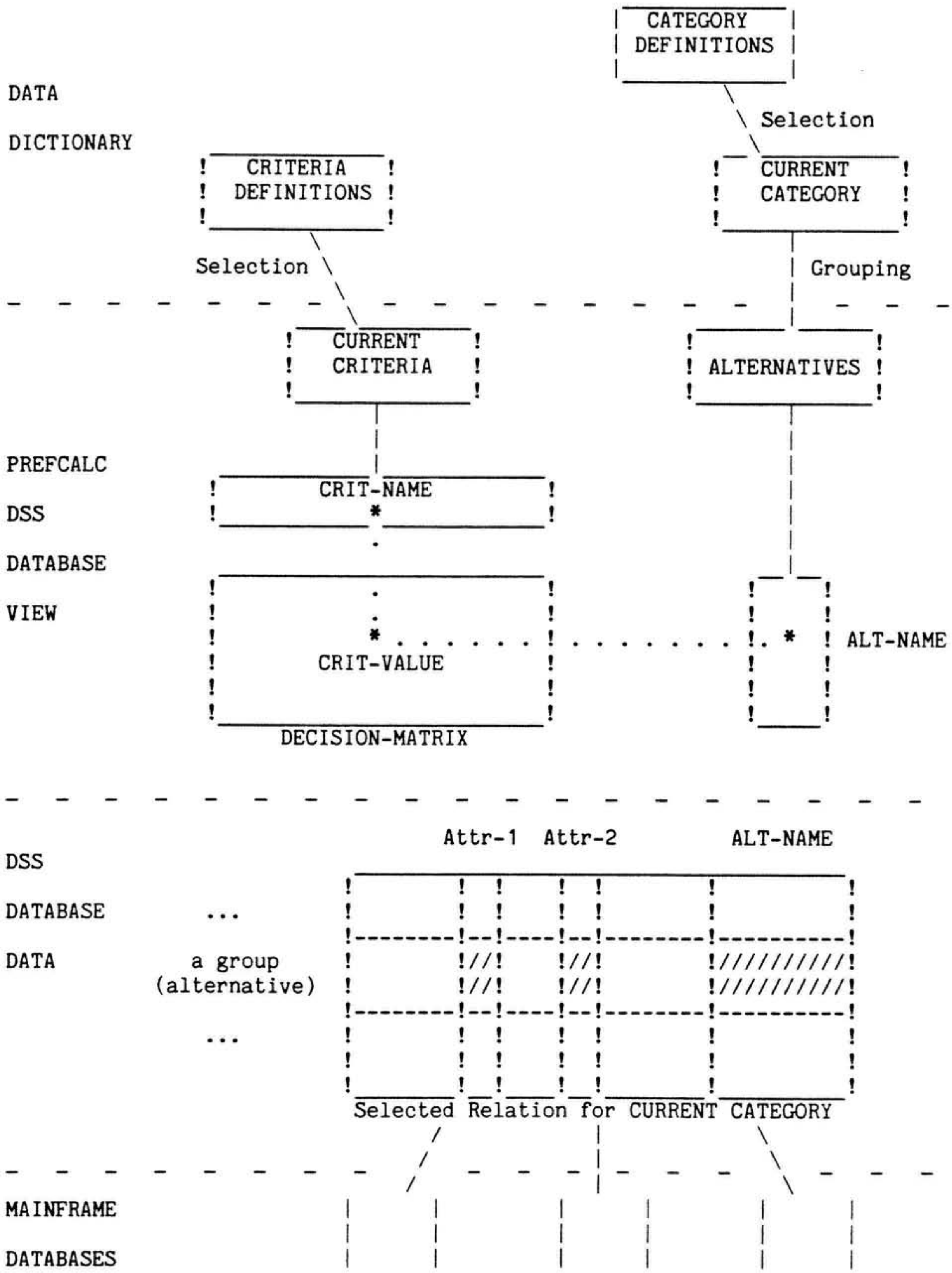

FIGURE 8: Data Extraction in MEDIATOR (simplified from (Jarke et al., 1984))

1) Player 2 decides to look at subcompacts only. A category "subcompact" has been previously defined for the "cars" mainframe database as follows:

```
DEFINE CATEGORY subcompact FOR cars
       WHERE length < 480;
```

The player can invoke this category simply by menu selection:

CURRENT CATEGORY: subcompact

make	model	version	DIN-Cons	Length	Width	Price	Speed	...
Opel	Record	2000	11.4	462	173	46700	176	
Peugeot	505	GR	9.4	458	172	49500	173	
Peugeot	104	ZS	8.7	337	152	35200	161	
Citroen	Dyane		6.8	387	150	24800	117	
Citroen	Visa	Super E	9.2	369	154	32100	142	
VW	Golf	GLS	9.7	398	158	39150	148	
VW	Golf	GTI	11.2	398	158	45000	181 *	
Mercedes	230	E	13.6	472	179	75700	180	
Citroen	CX	PALLAS	14.1	465	177	64700	178	
Citroen	CX	GTI	15.3	465	177	69700	178 *	
Volvo	244	GLE	16.9	479	171	51000	140 +	
Volvo	244	TURBO	12.5	479	171	59000	150 +	
BMW	520		12.5	462	170	68593	182	
...								

2) Player 2 does not like "GTI" sports versions and therefore adds the following category definition to the CATEGORIES model base:

```
DEFINE CATEGORY no-sport FOR subcompact
       WHERE version <> "GTI";
```

the application of which eliminates the rows marked '*'.

3) Player 2 is not interested in "length" and "width" of the car but only in the "space" it provides. The CRITERIA DEFINITIONS model base contains the definition of a virtual attribute "space":

```
DEFINE CRITERION space FOR cars
       AS avg(length*width)
```

which is inherited by all subrelations of cars and can again be applied by simple menu selection. avg(x) is a "vertical" function that computes the average value of x over all rows in a particular alternative (as defined in step 4, below).

FIGURE 9: GENERATING A PROBLEM REPRESENTATION FOR PLAYER 2

4) Player 2 is not interested in the differences between the different versions but wants to define alternatives only via make and model (as shown in the GROUP-BY below). Therefore, the two rows marked '+' must be combined into one. In order to do this, player 2 defines the DECISION-MATRIX as follows (again, MEDIATOR uses menu selection to make this process more user-friendly):

```
DEFINE VIEW decision-matrix (make, model, consumption, space, price)
AS SELECT   make, model, avg(DIN-cons), space, avg(price)
   FROM     no-sports
   GROUP-BY make, model
```

which results in the following decision matrix for Player 2:

make	model	Consumption	Space	Price
Opel	Record	11.4	7.96	46700
Peugeot	505	9.4	7.88	49500
Peugeot	104	8.7	5.11	35200
Citroen	Dyane	6.8	5.81	24800
Citroen	Visa	9.2	5.65	32100
VW	Golf	9.7	6.15	39150
Mercedes	230	13.6	8.47	75700
Citroen	CX	14.1	8.06	64700
Volvo	244	14.7	8.38	55000
BMW	520	12.5	7.81	68593

FIGURE 9: GENERATING A PROBLEM REPRESENTATION FOR PLAYER 2 (cont.)

TABLE I

GROUP RELATIONAL REPRESENTATION (REVERSE UTILITIES)
CORRESPONDING TO FIGURES 1 AND 2

control make	model	goal y_1 C120	goal y_2 Space	goal y_3 Price	goal y_4 Speed	initial utilities u_1	initial utilities u_2
Opel	Record	10.48	7.96	46700	176	.75	.38
Peugeot	505	10.01	7.88	49500	173	.73	.40
Peugeot	104	8.42	5.11	35200	161	.72	.44
Citroen	Dyane	6.75	5.81	24800	117	.65	.46
Citroen	Visa	7.30	5.65	32100	142	.62	.58
VW	Golf	9.61	6.15	39150	148	.58	.62
Mercedes	230	10.40	8.47	75700	180	.46	.65
Citroen	CX	11.05	8.06	64700	178	.44	.72
Volvo	244	12.95	8.38	55000	145	.40	.73
BMW	520	12.26	7.81	68593	182	.38	.75

TABLE II

GROUP PROBLEM REPRESENTATION

control make	control model	goal y_1 C120	goal y_2 Space	goal y_3 Price	goal y_4 Speed	initial utilities u_1	initial utilities u_2	First evolved utilities u_1	First evolved utilities u_2	Second evolved utilities u_1	Second evolved utilities u_2
Opel	Record	10.48	7.96	46700	176	.75	.62	.74	.66	.74	.70
Peugeot	505	10.01	7.88	49500	173	.73	.58	.72	.63	.72	.57
Peugeot	104	8.42	5.11	35200	161	.72	.15	.67	.29	.67	.27
Citroen	Dyane	6.75	5.81	24800	117	.65	.31	.64	.45	.64	.40
Citroen	Visa	7.30	5.65	32100	142	.62	.24	.60	.38	.60	.34
VW	Golf	9.61	6.15	39150	148	.58	.21	.55	.33	.55	.42
Mercedes	230	10.40	8.47	75700	180	.46	.76	.45	.65	.45	.70
Citroen	CX	11.05	8.06	64700	178	.44	.59	.43	.54	.43	.49
Volvo	244	12.95	8.38	55000	145	.40	.75	.44	.65	.44	.60
BMW	520	12.26	7.81	68593	182	.38	.49	.39	.47	.39	.52

TABLE III

INITIAL CAR RANK ORDERS AND UTILITIES

PLAYER 1			PLAYER 2		
Opel	Record	.75	Mercedes	230	.76
Peugeot	505	.73	Volvo	244	.75
Peugeot	104	.72	Opel	Record	.62
Citroen	Dyane	.65	Citroen	CX	.59
Citroen	Visa	.62	Peugoet	505	.58
VW	Golf	.58	BMW	520	.49
Mercedes	230	.46	Citroen	Dyane	.31
Citroen	CX	.44	Citroen	Visa	.24
Volvo	244	.40	VW	Golf	.21
BMW	520	.38	Peugeot	104	.15

TABLE IV

INITIAL NORMALIZED UTILITIES

		Player 1	Player 2	Minimum Utility
Opel	Record	1	.77	.77
Peugeot	505	.95	.71	.71
Peugeot	104	.92	0	0
Citroen	Dyane	.73	.26	.26
Citroen	Visa	.65	.15	.15
VW	Golf	.54	.10	.10
Mercedes	230	.22	1	.22
Citroen	CX	.16	.72	.16
Volvo	244	.05	.98	.05
BMW	520	0	.56	0

TABLE V

SECOND EVOLVED NORMALIZED UTILITIES

		Player 1	Player 2	Minimum Utilites
Opel	Record	1	1	1
Peugeot	505	.94	.70	.70
Peugeot	104	.80	0	0
Citroen	Dyane	.71	.30	.30
Citroen	Visa	.60	.16	.16
VW	Golf	.46	.35	.35
Mercedes	230	.17	1	.17
Citroen	CX	.11	.51	.11
Volvo	244	.14	.77	.14
BMW	520	0	.58	0