Fig. 1

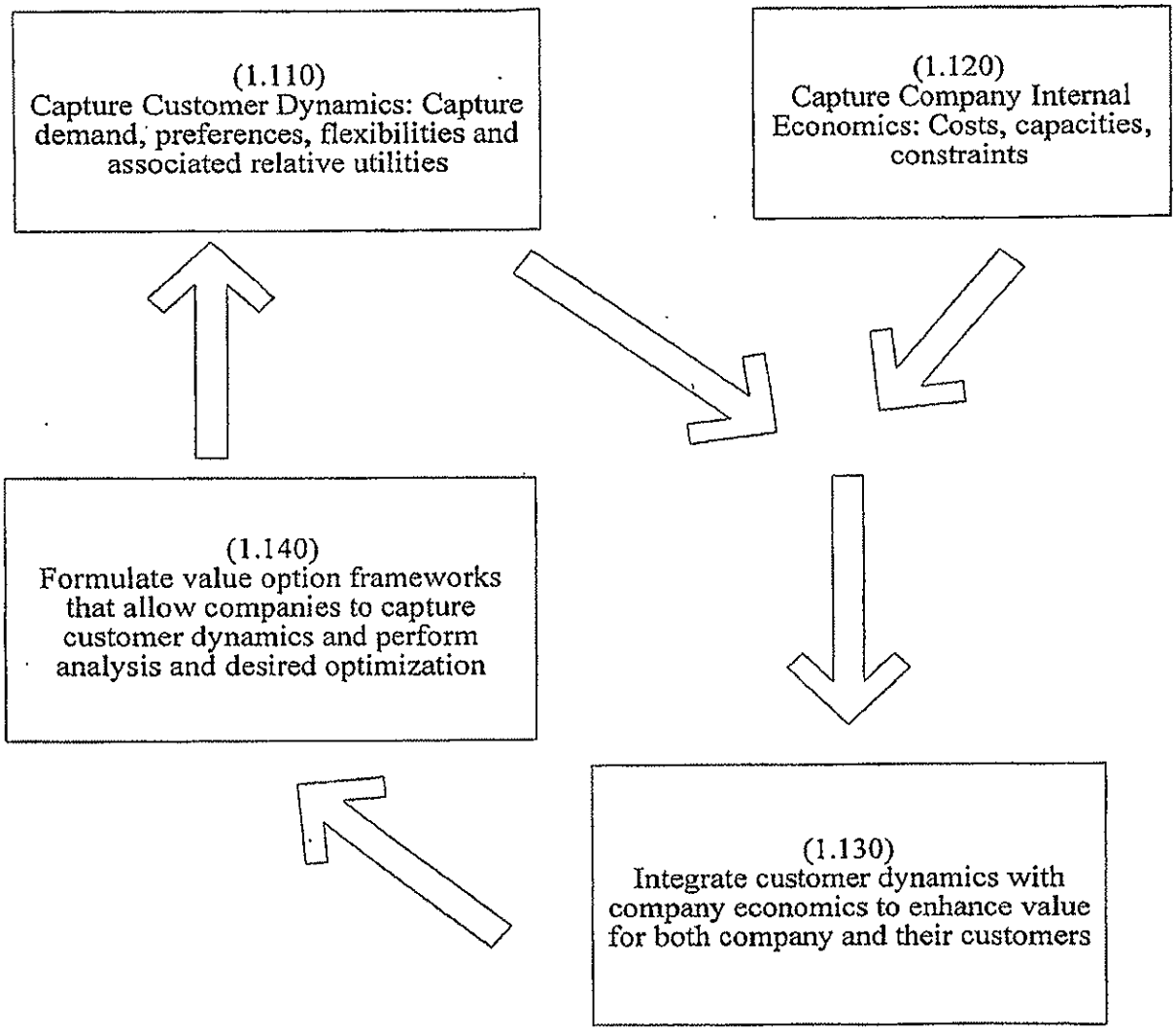

(1.110)
Capture Customer Dynamics: Capture demand, preferences, flexibilities and associated relative utilities
(1.120)
Capture Company Internal Economics: Costs, capacities, constraints
(1.140)
Formulate value option frameworks that allow companies to capture customer dynamics and perform analysis and desired optimization
(1.130)
Integrate customer dynamics with company economics to enhance value for both company and their customers

Fig. 2

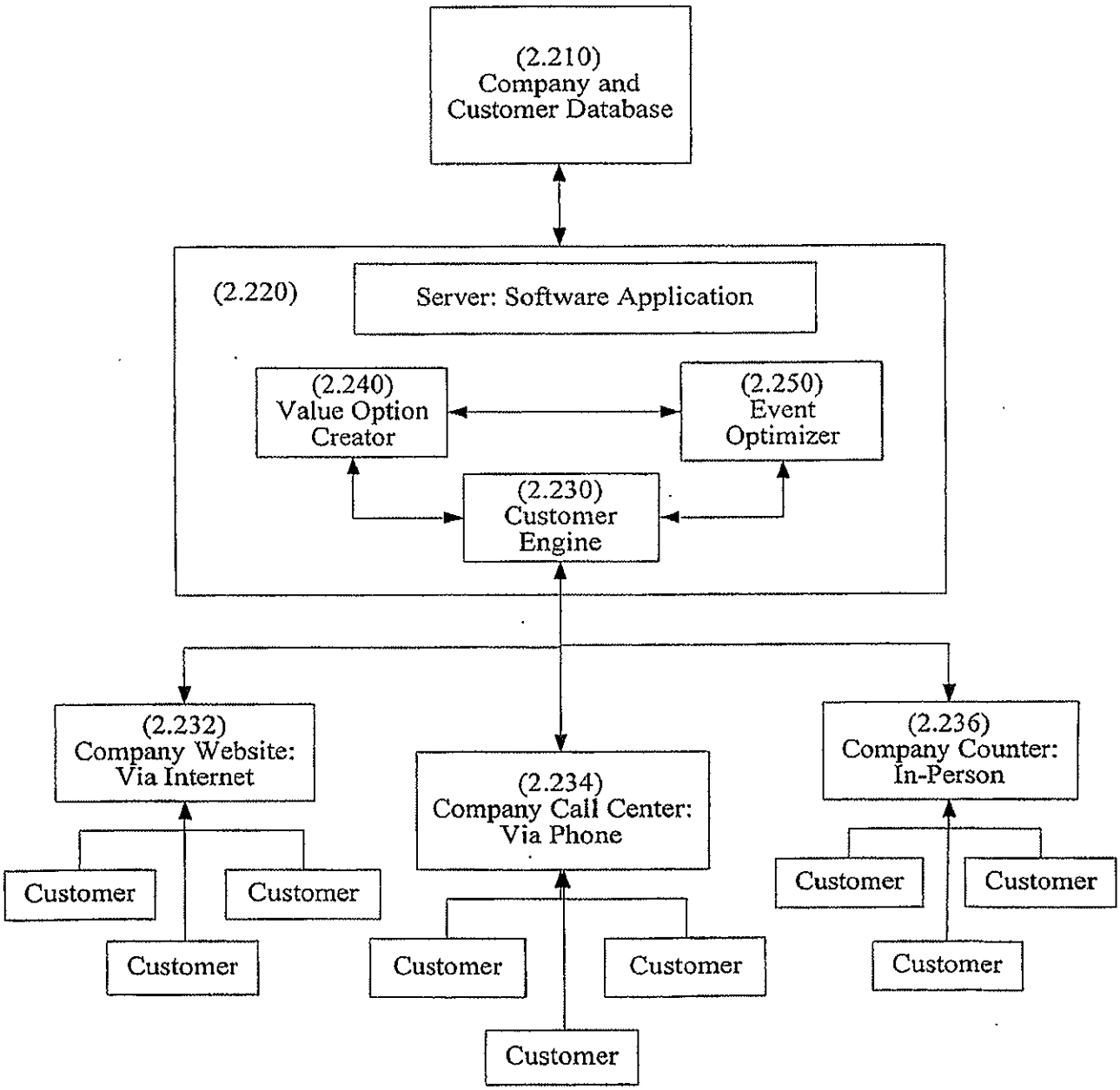

(2.210)
Company and
Customer Database
(2.220)
Server: Software Application
(2.240)
Value Option
Creator
(2.250)
Event
Optimizer
(2.230)
Customer
Engine
(2.232)
Company Website:
Via Internet
(2.234)
Company Call Center:
Via Phone
(2.236)
Company Counter:
In-Person
Customer
Customer
Customer
Customer
Customer
Customer
Customer
Customer
Customer

Fig. 3

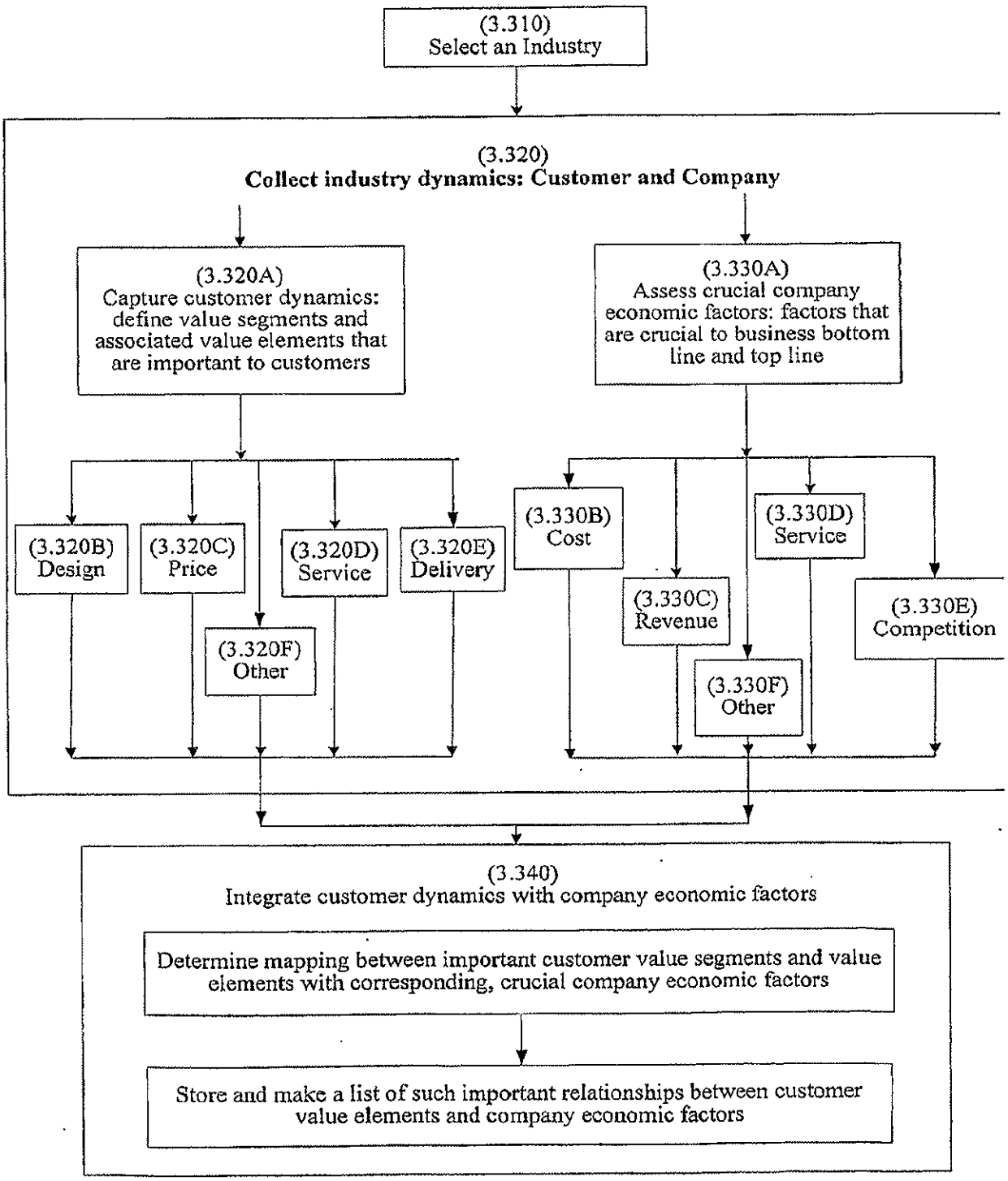

(3.310)
Select an Industry
(3.320)
Collect industry dynamics: Customer and Company
(3.320A)
Capture customer dynamics: define value segments and associated value elements that are important to customers
(3.330A)
Assess crucial company economic factors: factors that are crucial to business bottom line and top line
(3.320B) Design
(3.320C) Price
(3.320D) Service
(3.320E) Delivery
(3.320F) Other
(3.330B) Cost
(3.330C) Revenue
(3.330D) Service
(3.330E) Competition
(3.330F) Other
(3.340)
Integrate customer dynamics with company economic factors
Determine mapping between important customer value segments and value elements with corresponding, crucial company economic factors
Store and make a list of such important relationships between customer value elements and company economic factors

Fig. 4

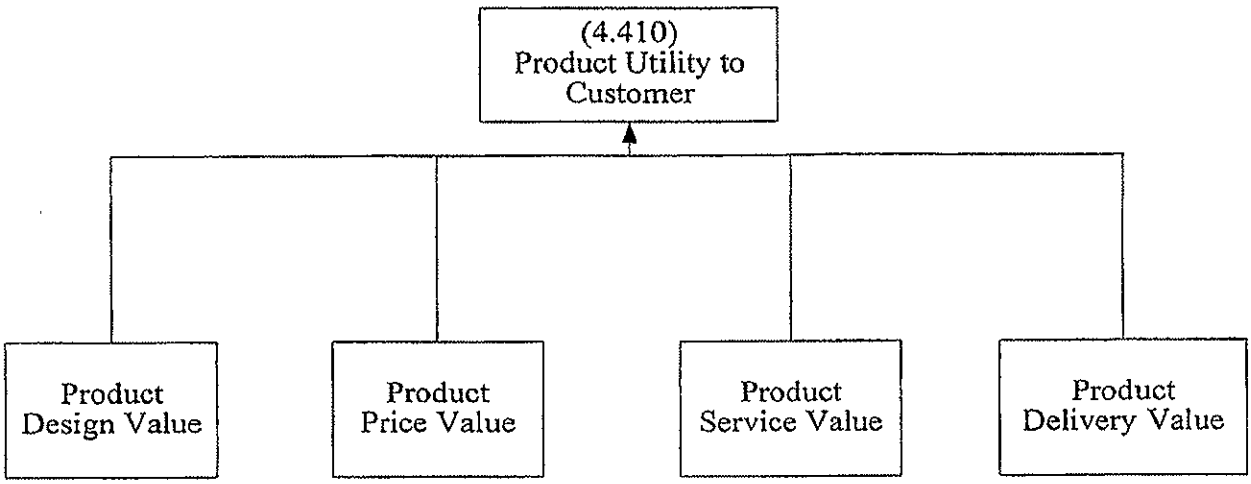

(4.410)
Product Utility to Customer
Product Design Value
Product Price Value
Product Service Value
Product Delivery Value

Fig. 5

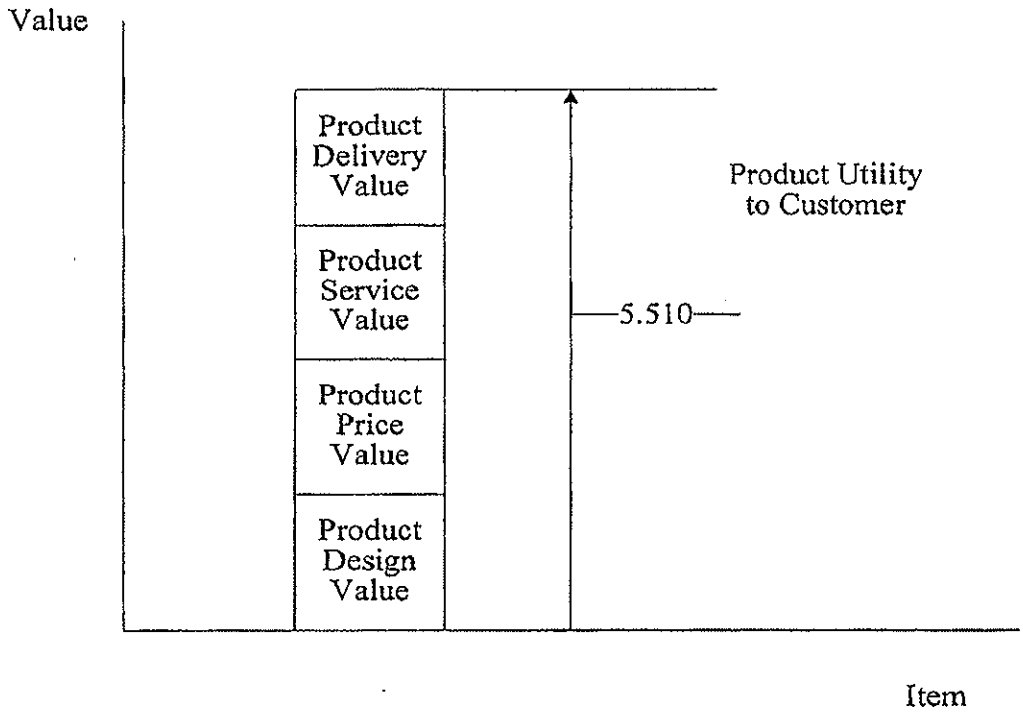

Value
Product Delivery Value
Product Service Value
Product Price Value
Product Design Value
Product Utility to Customer
5.510
Item

Fig. 6

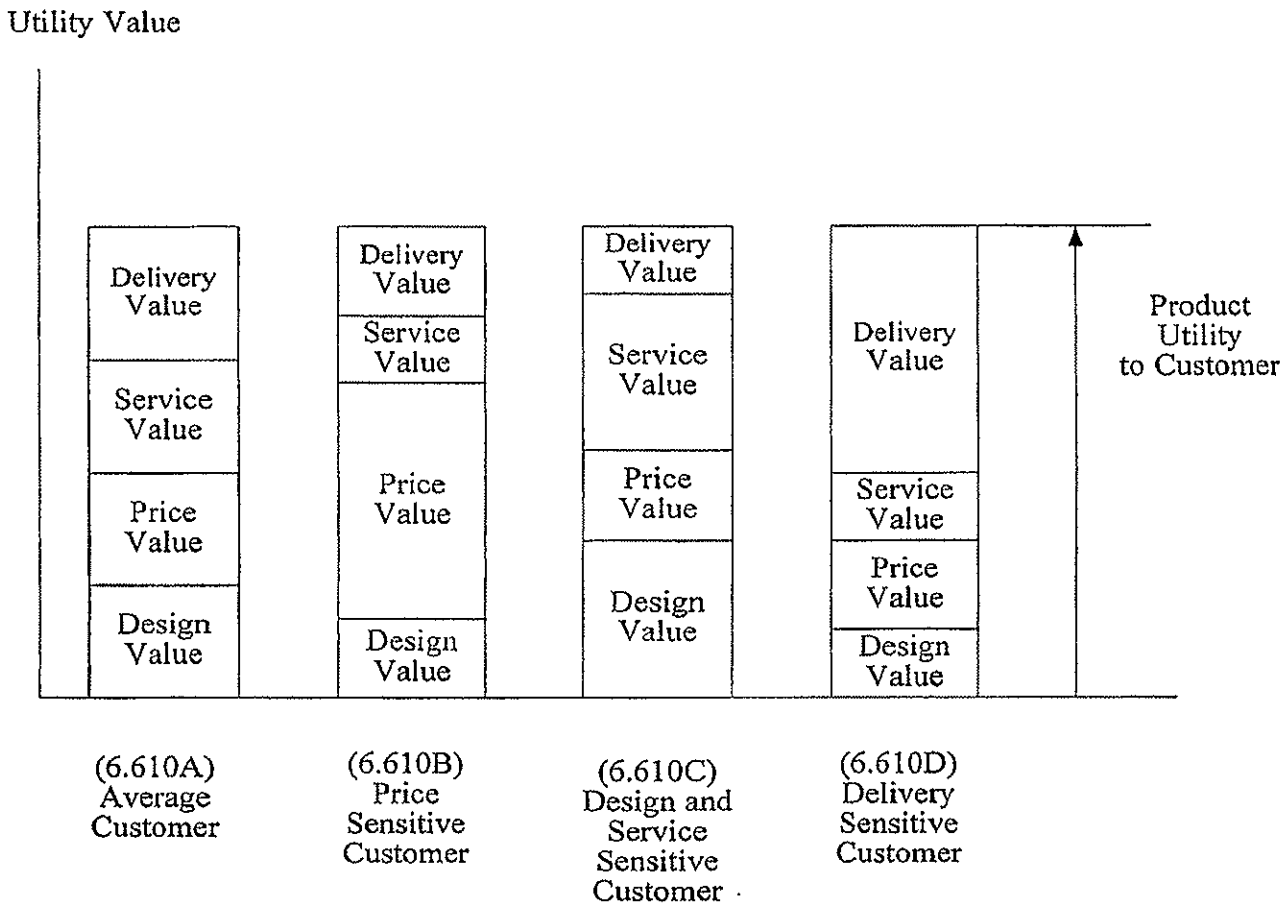
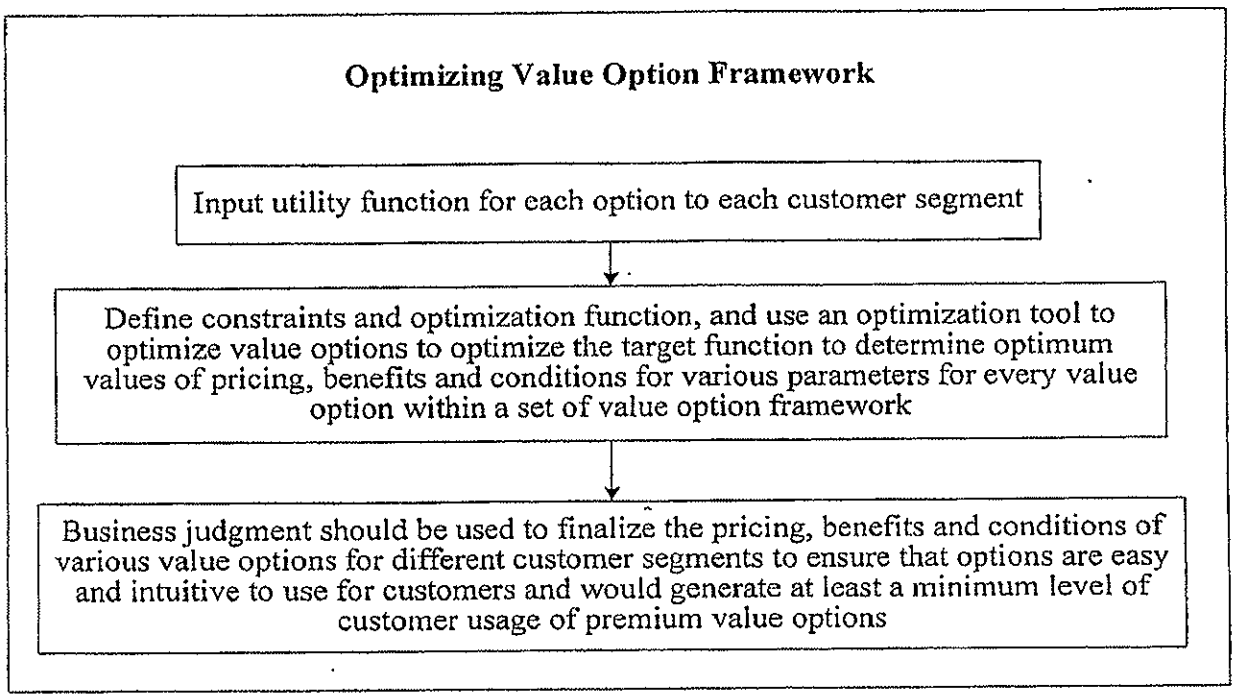

Utility Value
Delivery Value
Service Value
Price Value
Design Value
Delivery Value
Service Value
Price Value
Design Value
Delivery Value
Service Value
Price Value
Design Value
Delivery Value
Service Value
Price Value
Design Value
Product Utility to Customer
(6.610A) Average Customer
(6.610B) Price Sensitive Customer
(6.610C) Design and Service Sensitive Customer
(6.610D) Delivery Sensitive Customer

Fig. 7



The system is now ready to be deployed to generate real world interactions with the customers

Fig. 8

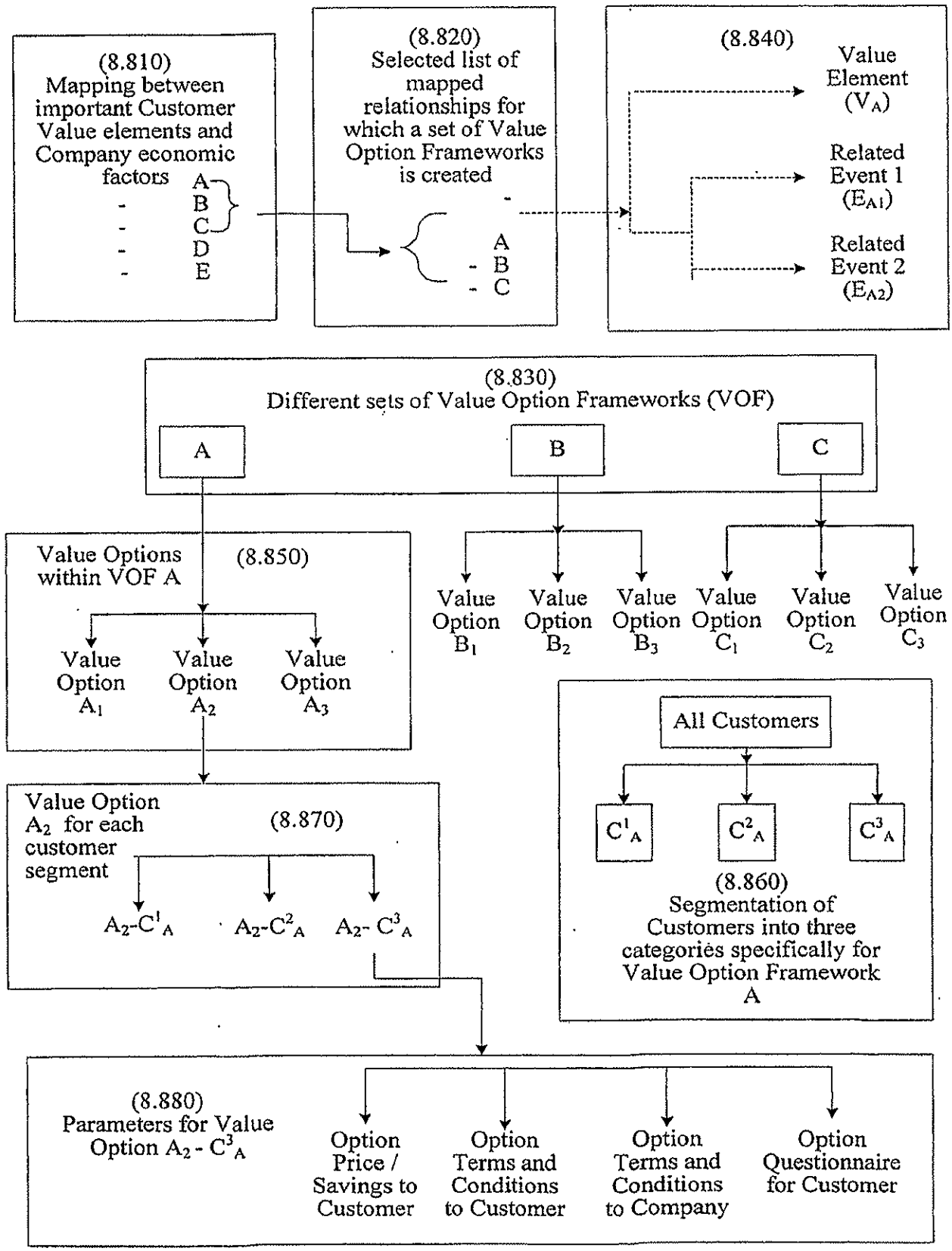

(8.810)
Mapping between important Customer Value elements and Company economic factors
A
B
C
D
E
(8.820)
Selected list of mapped relationships for which a set of Value Option Frameworks is created
A
B
C
(8.840)
Value Element (V_A)
Related Event 1 (E_{A1})
Related Event 2 (E_{A2})
(8.830)
Different sets of Value Option Frameworks (VOF)
A
B
C
Value Options within VOF A
(8.850)
Value Option A_1
Value Option A_2
Value Option A_3
Value Option B_1
Value Option B_2
Value Option B_3
Value Option C_1
Value Option C_2
Value Option C_3
All Customers
C^1_A
C^2_A
C^3_A
(8.860)
Segmentation of Customers into three categories specifically for Value Option Framework A
Value Option A_2 for each customer segment
(8.870)
A_2-C^1_A
A_2-C^2_A
A_2-C^3_A
(8.880)
Parameters for Value Option A_2 - C^3_A
Option Price / Savings to Customer
Option Terms and Conditions to Customer
Option Terms and Conditions to Company
Option Questionnaire for Customer

Fig. 9

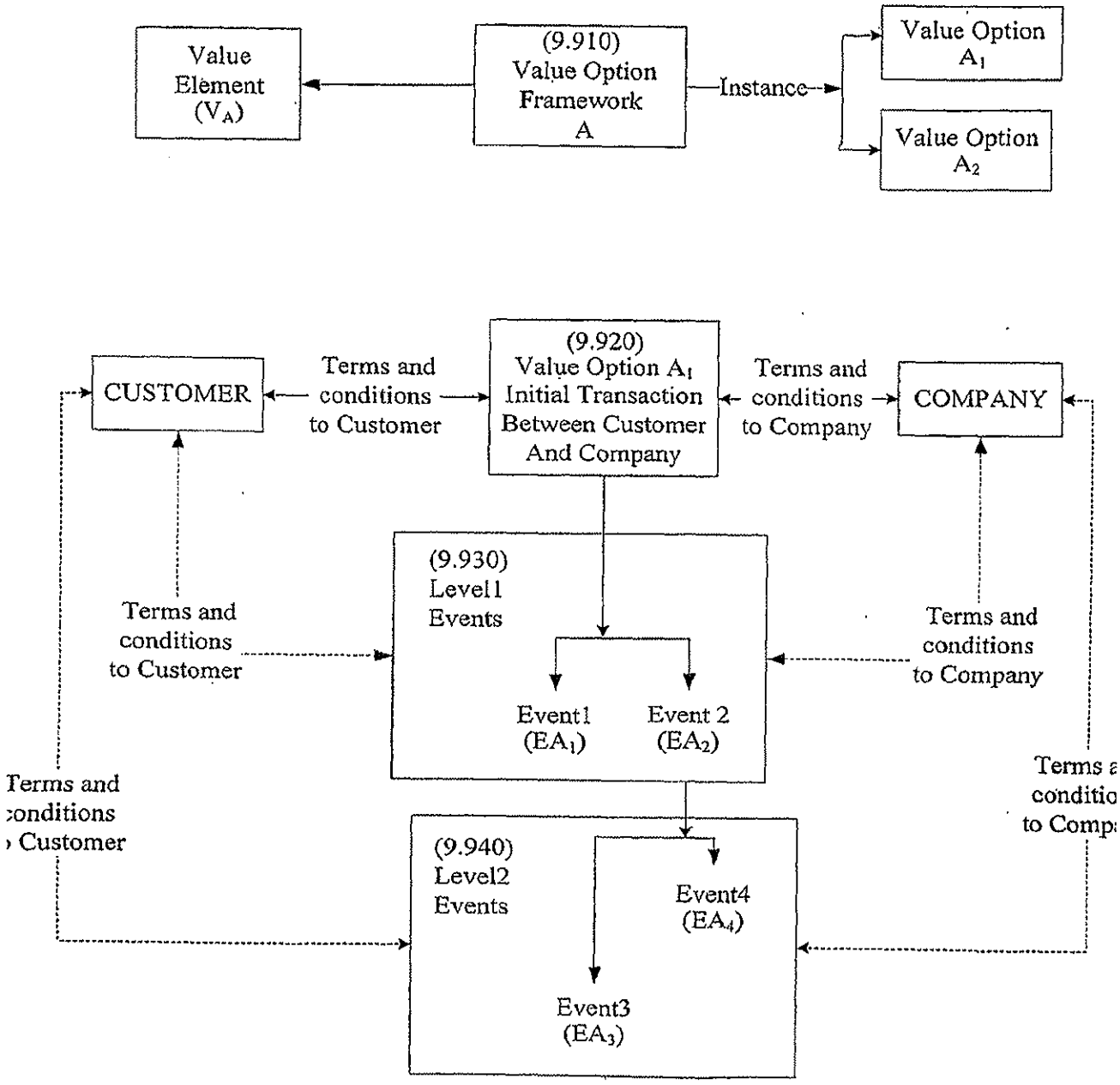

Value Element (V_A)
(9.910) Value Option Framework A
Instance
Value Option A_1
Value Option A_2
CUSTOMER
Terms and conditions to Customer
(9.920) Value Option A_1 Initial Transaction Between Customer And Company
Terms and conditions to Company
COMPANY
(9.930) Level1 Events
Terms and conditions to Customer
Terms and conditions to Company
Event1 (EA_1)
Event 2 (EA_2)
Terms and conditions to Customer
Terms a conditio to Comp
(9.940) Level2 Events
Event4 (EA_4)
Event3 (EA_3)

Fig. 10

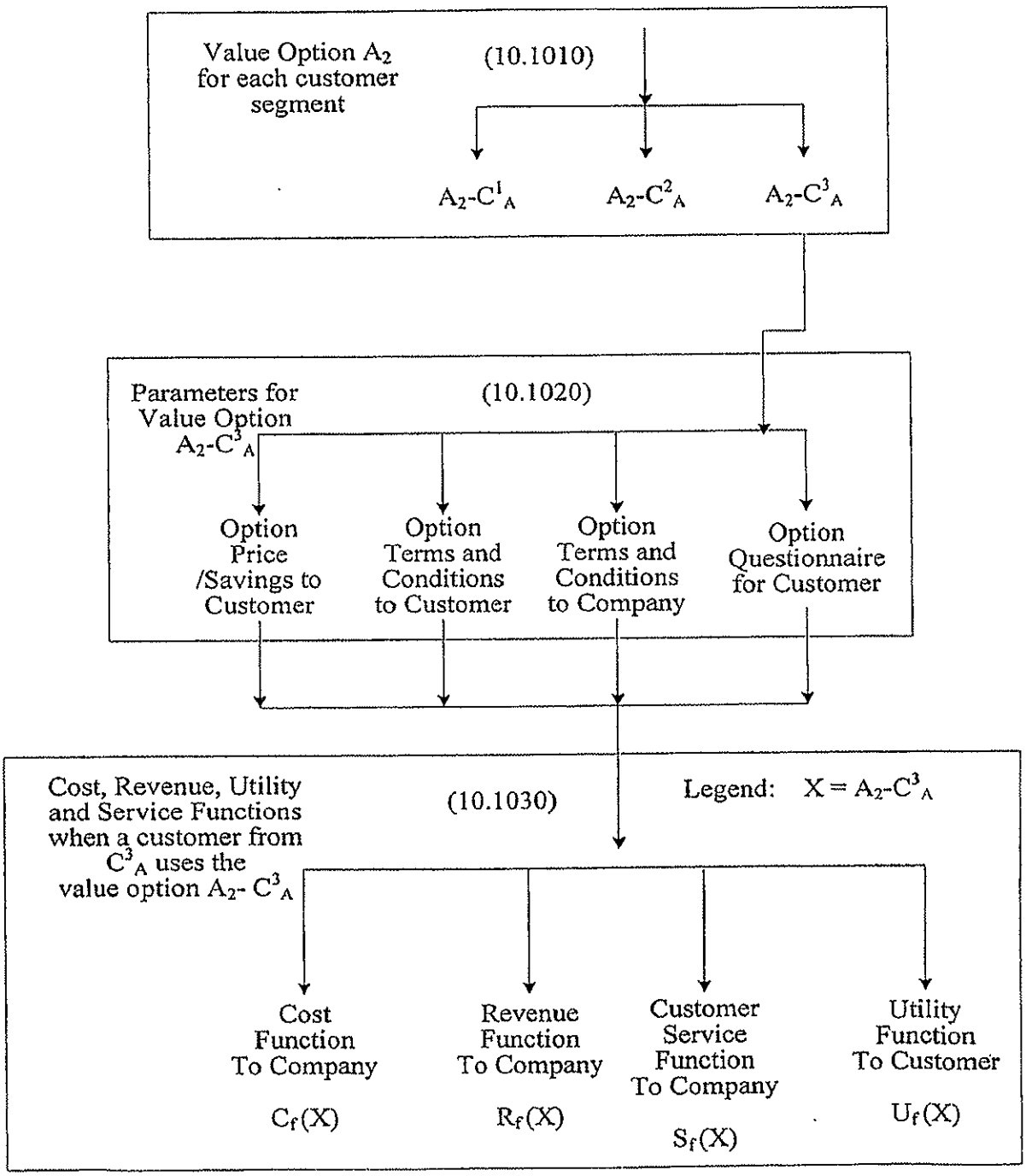

Value Option A_2 for each customer segment
(10.1010)
A_2-C^1_A
A_2-C^2_A
A_2-C^3_A
Parameters for Value Option A_2-C^3_A
(10.1020)
Option Price /Savings to Customer
Option Terms and Conditions to Customer
Option Terms and Conditions to Company
Option Questionnaire for Customer
Cost, Revenue, Utility and Service Functions when a customer from C^3_A uses the value option A_2- C^3_A
(10.1030)
Legend: $X = A_2$-C^3_A
Cost Function To Company
$C_f(X)$
Revenue Function To Company
$R_f(X)$
Customer Service Function To Company
$S_f(X)$
Utility Function To Customer
$U_f(X)$

Fig. 11

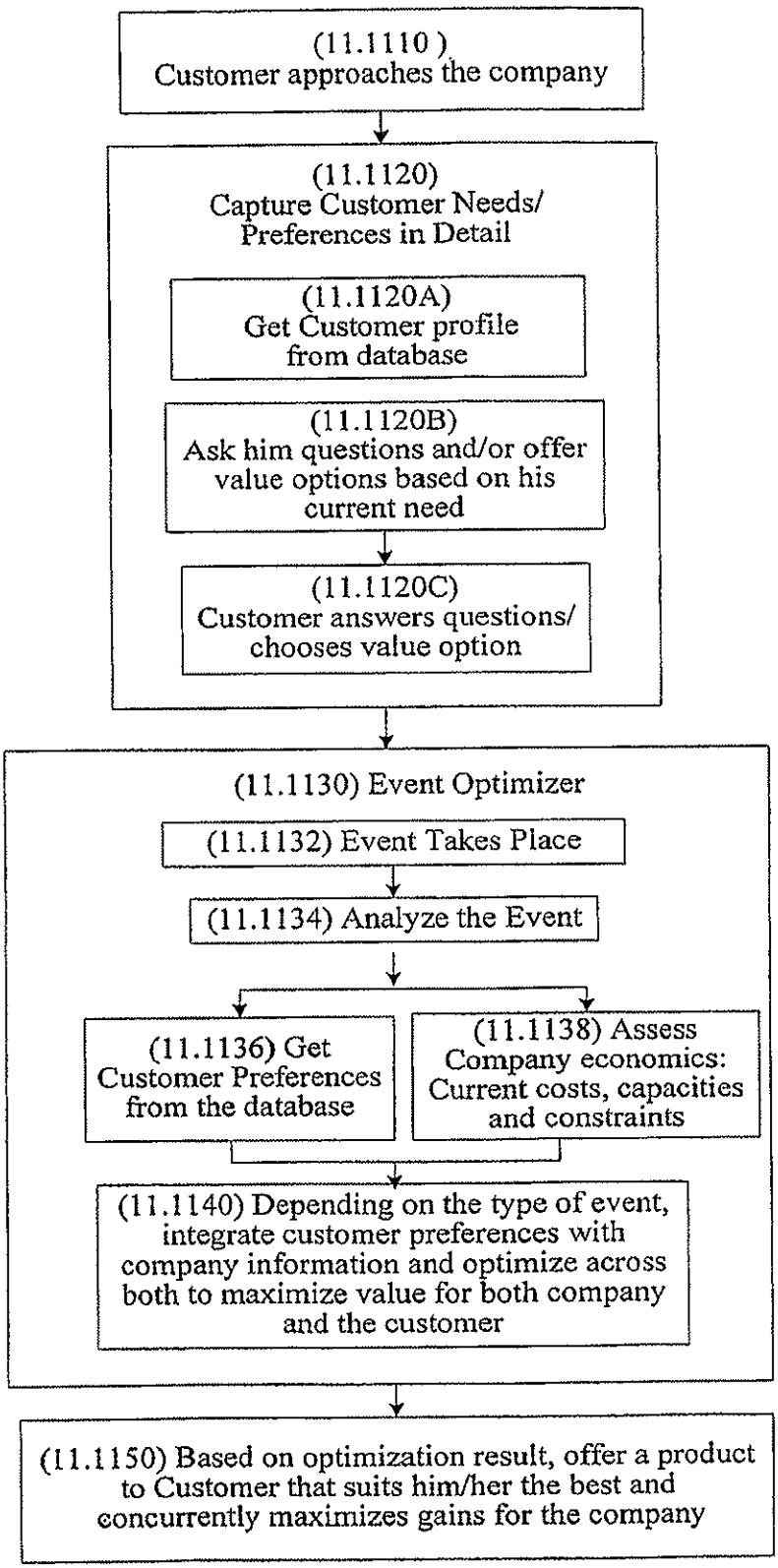

(11.1110)
Customer approaches the company
(11.1120)
Capture Customer Needs/
Preferences in Detail
(11.1120A)
Get Customer profile
from database
(11.1120B)
Ask him questions and/or offer
value options based on his
current need
(11.1120C)
Customer answers questions/
chooses value option
(11.1130) Event Optimizer
(11.1132) Event Takes Place
(11.1134) Analyze the Event
(11.1136) Get
Customer Preferences
from the database
(11.1138) Assess
Company economics:
Current costs, capacities
and constraints
(11.1140) Depending on the type of event,
integrate customer preferences with
company information and optimize across
both to maximize value for both company
and the customer
(11.1150) Based on optimization result, offer a product
to Customer that suits him/her the best and
concurrently maximizes gains for the company

Fig. 12

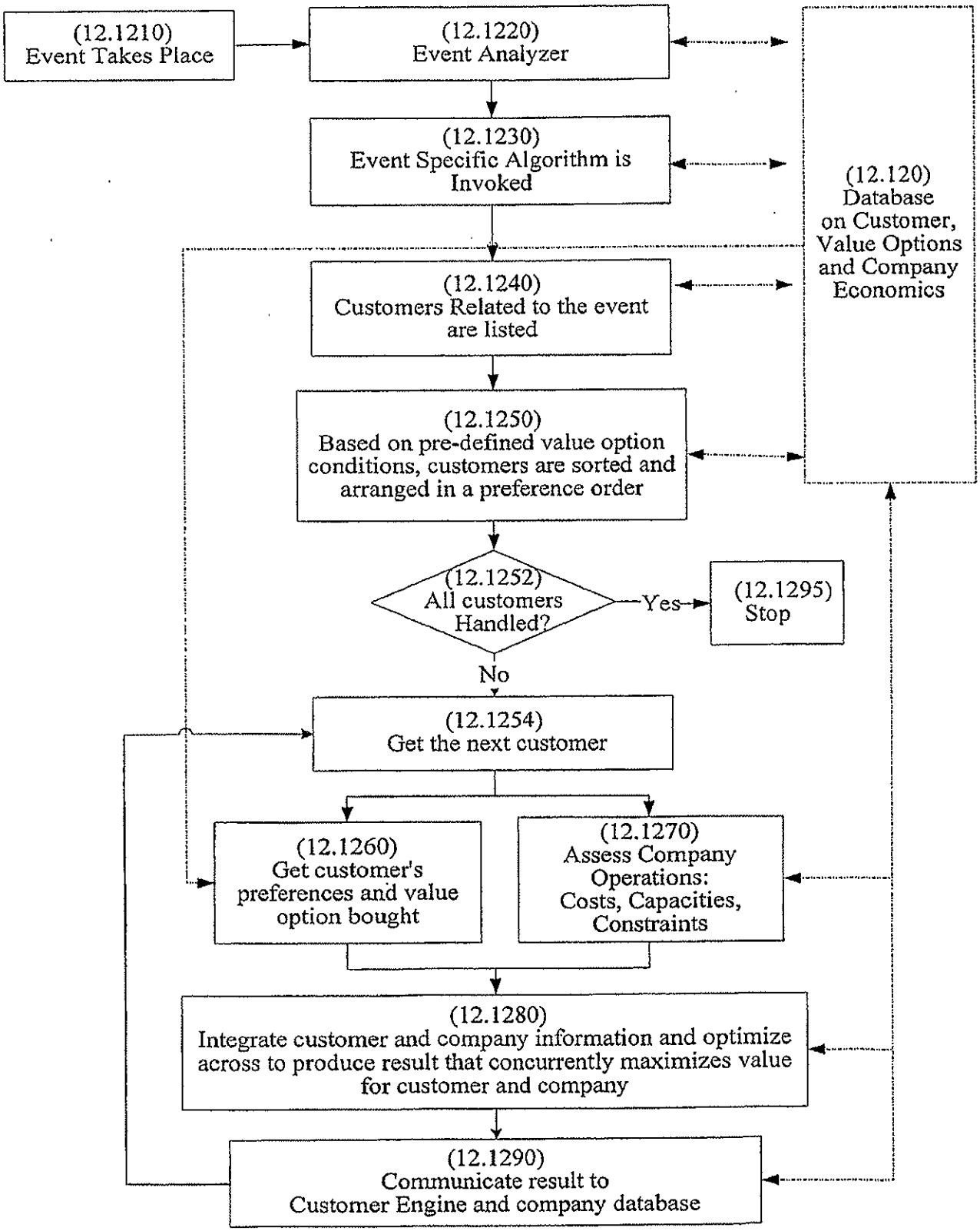

(12.1210) Event Takes Place
(12.1220) Event Analyzer
(12.1230) Event Specific Algorithm is Invoked
(12.120) Database on Customer, Value Options and Company Economics
(12.1240) Customers Related to the event are listed
(12.1250) Based on pre-defined value option conditions, customers are sorted and arranged in a preference order
(12.1252) All customers Handled?
Yes
(12.1295) Stop
No
(12.1254) Get the next customer
(12.1260) Get customer's preferences and value option bought
(12.1270) Assess Company Operations: Costs, Capacities, Constraints
(12.1280) Integrate customer and company information and optimize across to produce result that concurrently maximizes value for customer and company
(12.1290) Communicate result to Customer Engine and company database

Fig. 13

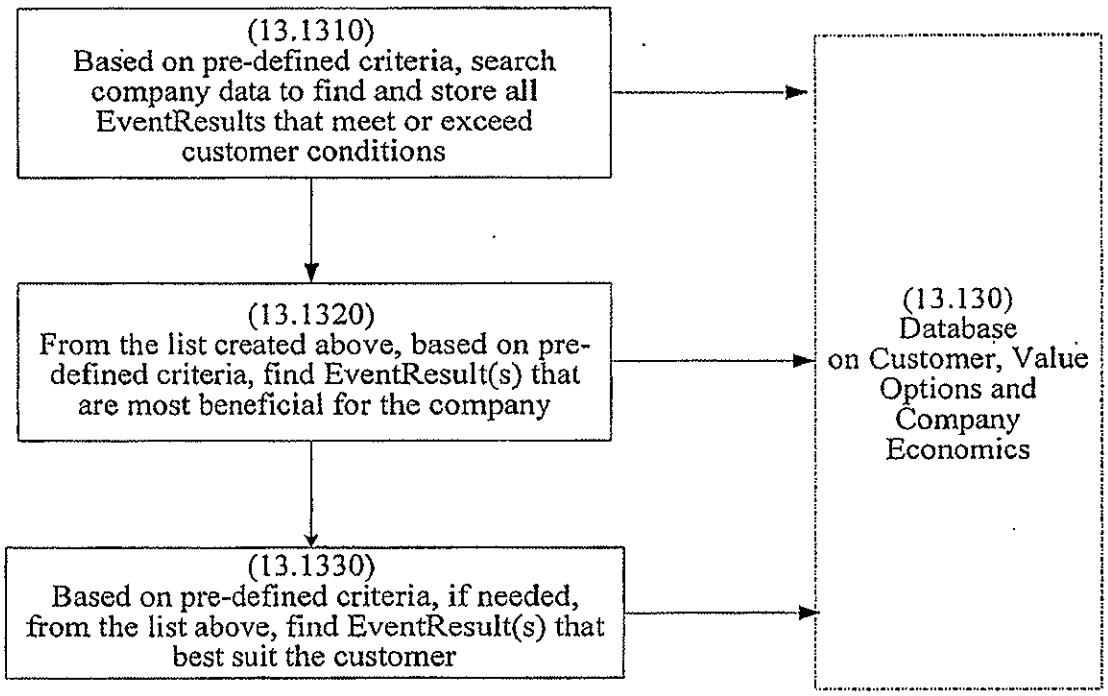

(13.1310)
Based on pre-defined criteria, search company data to find and store all EventResults that meet or exceed customer conditions
(13.1320)
From the list created above, based on pre-defined criteria, find EventResult(s) that are most beneficial for the company
(13.1330)
Based on pre-defined criteria, if needed, from the list above, find EventResult(s) that best suit the customer
(13.130)
Database on Customer, Value Options and Company Economics

Fig. 13A

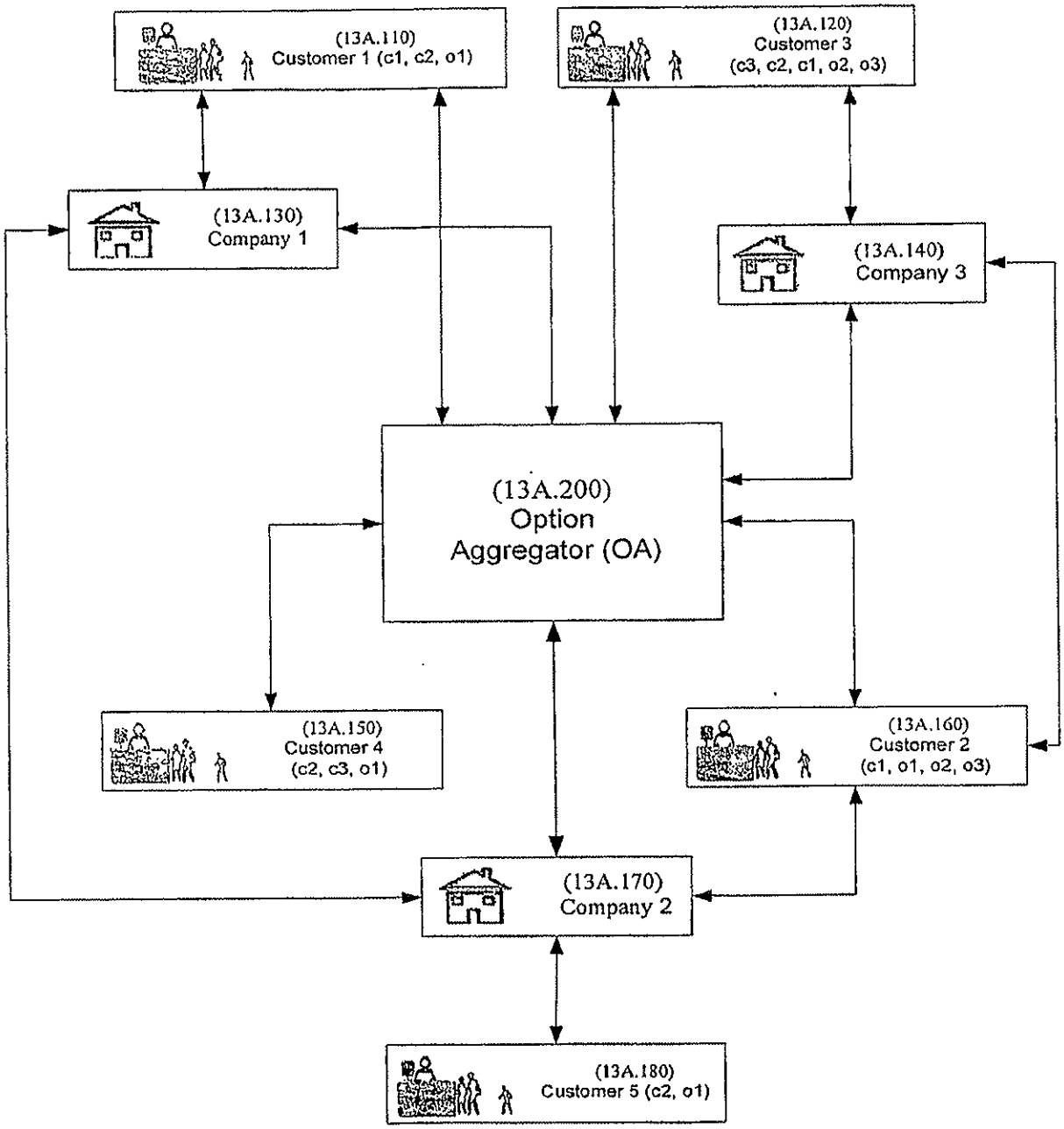

(13A.110)
Customer 1 (c1, c2, o1)
(13A.120)
Customer 3
(c3, c2, c1, o2, o3)
(13A.130)
Company 1
(13A.140)
Company 3
(13A.200)
Option
Aggregator (OA)
(13A.150)
Customer 4
(c2, c3, o1)
(13A.160)
Customer 2
(c1, o1, o2, o3)
(13A.170)
Company 2
(13A.180)
Customer 5 (c2, o1)

Fig. 13B

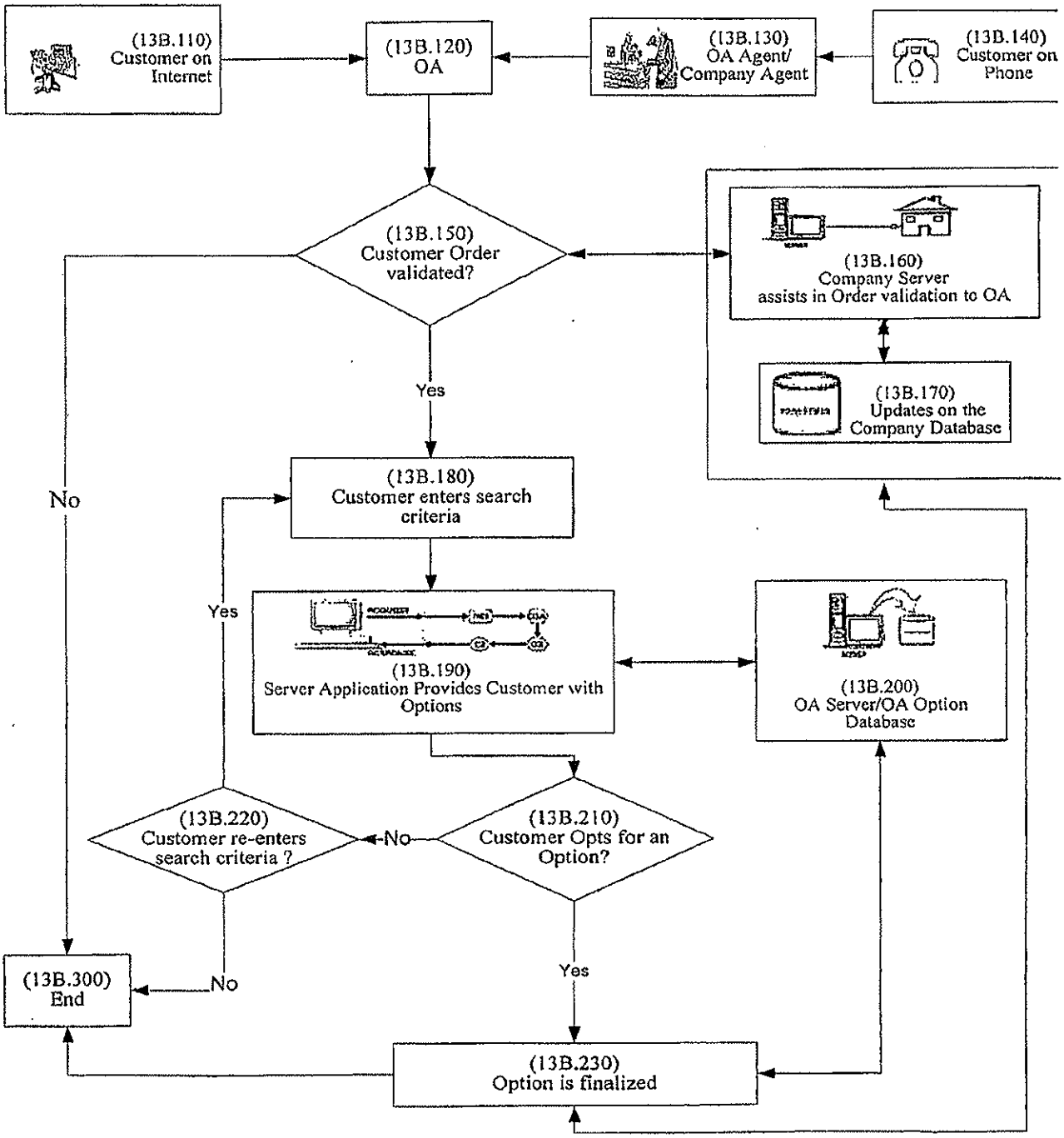

(13B.110) Customer on Internet
(13B.120) OA
(13B.130) OA Agent/ Company Agent
(13B.140) Customer on Phone
(13B.150) Customer Order validated?
(13B.160) Company Server assists in Order validation to OA
(13B.170) Updates on the Company Database
Yes
No
(13B.180) Customer enters search criteria
Yes
(13B.190) Server Application Provides Customer with Options
(13B.200) OA Server/OA Option Database
(13B.220) Customer re-enters search criteria ?
No
(13B.210) Customer Opts for an Option?
(13B.300) End
No
Yes
(13B.230) Option is finalized

Fig. 13C

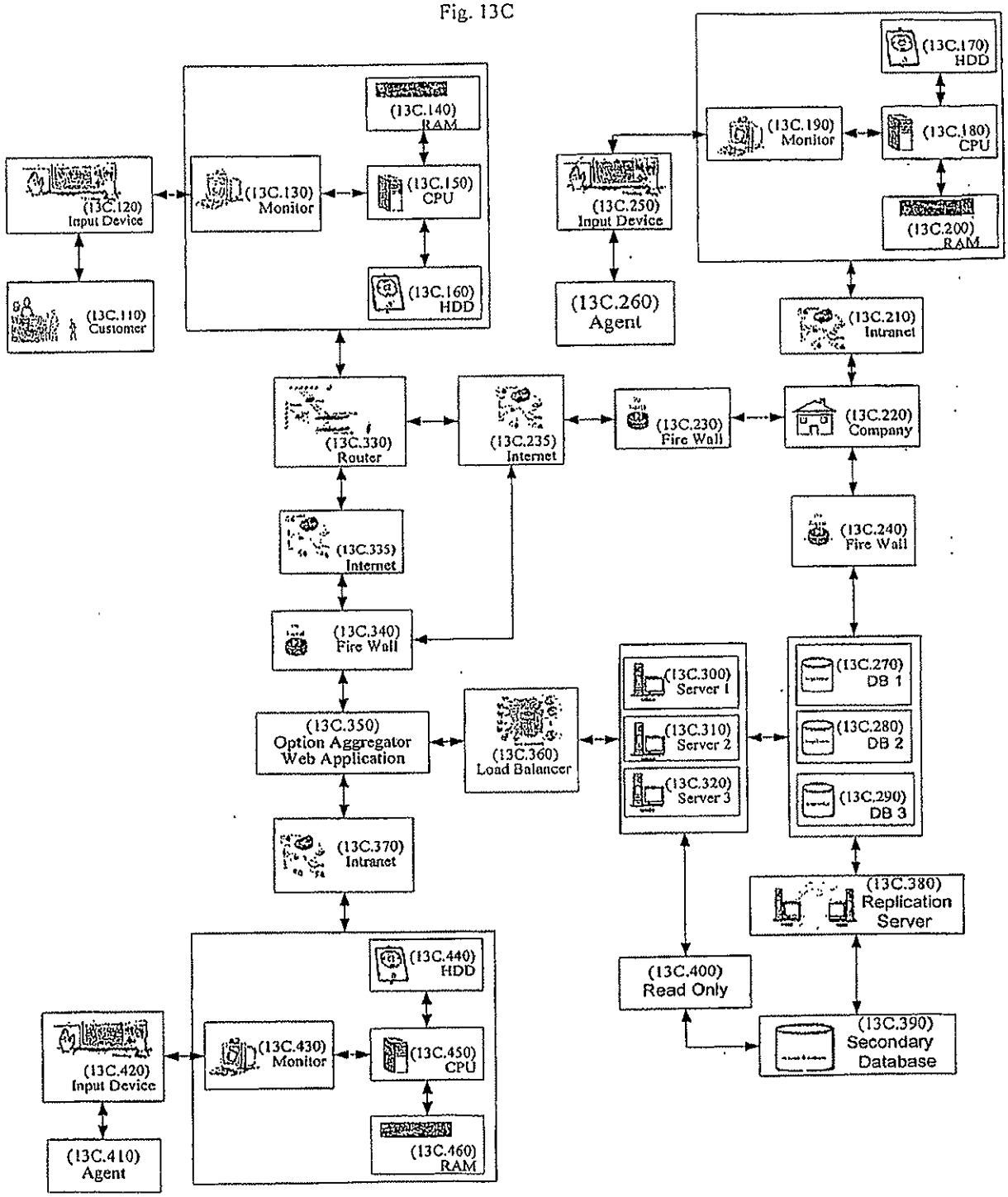

(13C.110) Customer
(13C.120) Input Device
(13C.130) Monitor
(13C.140) RAM
(13C.150) CPU
(13C.160) HDD
(13C.170) HDD
(13C.180) CPU
(13C.190) Monitor
(13C.200) RAM
(13C.210) Intranet
(13C.220) Company
(13C.230) Fire Wall
(13C.235) Internet
(13C.240) Fire Wall
(13C.250) Input Device
(13C.260) Agent
(13C.270) DB 1
(13C.280) DB 2
(13C.290) DB 3
(13C.300) Server 1
(13C.310) Server 2
(13C.320) Server 3
(13C.330) Router
(13C.335) Internet
(13C.340) Fire Wall
(13C.350) Option Aggregator Web Application
(13C.360) Load Balancer
(13C.370) Intranet
(13C.380) Replication Server
(13C.390) Secondary Database
(13C.400) Read Only
(13C.410) Agent
(13C.420) Input Device
(13C.430) Monitor
(13C.440) HDD
(13C.450) CPU
(13C.460) RAM

Fig. 13D

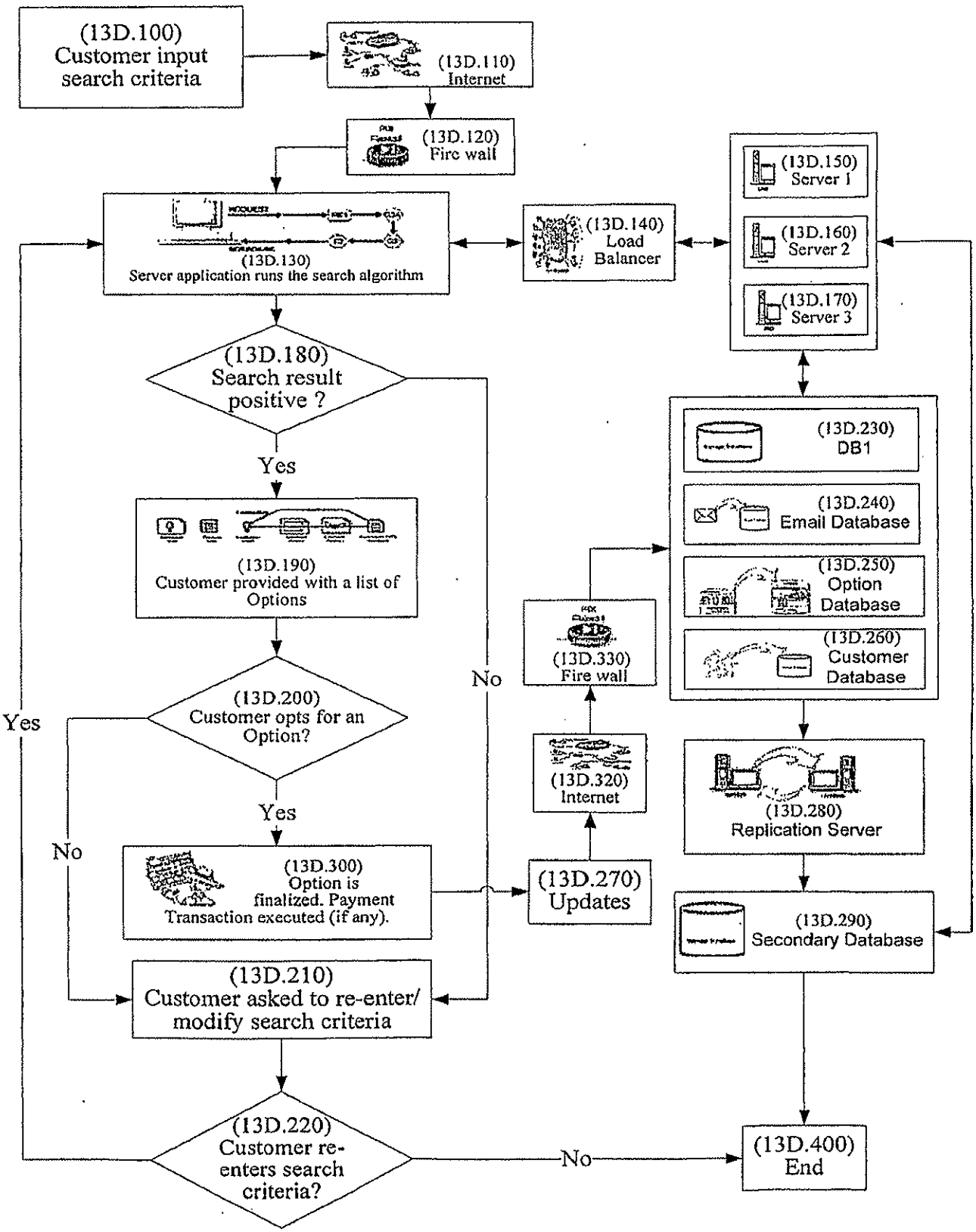

(13D.100) Customer input search criteria
(13D.110) Internet
(13D.120) Fire wall
(13D.130) Server application runs the search algorithm
(13D.140) Load Balancer
(13D.150) Server 1
(13D.160) Server 2
(13D.170) Server 3
(13D.180) Search result positive ?
Yes
No
(13D.190) Customer provided with a list of Options
(13D.200) Customer opts for an Option?
Yes
No
(13D.210) Customer asked to re-enter/ modify search criteria
(13D.220) Customer re-enters search criteria?
Yes
No
(13D.230) DB1
(13D.240) Email Database
(13D.250) Option Database
(13D.260) Customer Database
(13D.280) Replication Server
(13D.290) Secondary Database
(13D.300) Option is finalized. Payment Transaction executed (if any).
(13D.270) Updates
(13D.320) Internet
(13D.330) Fire wall
(13D.400) End

Fig. 13E

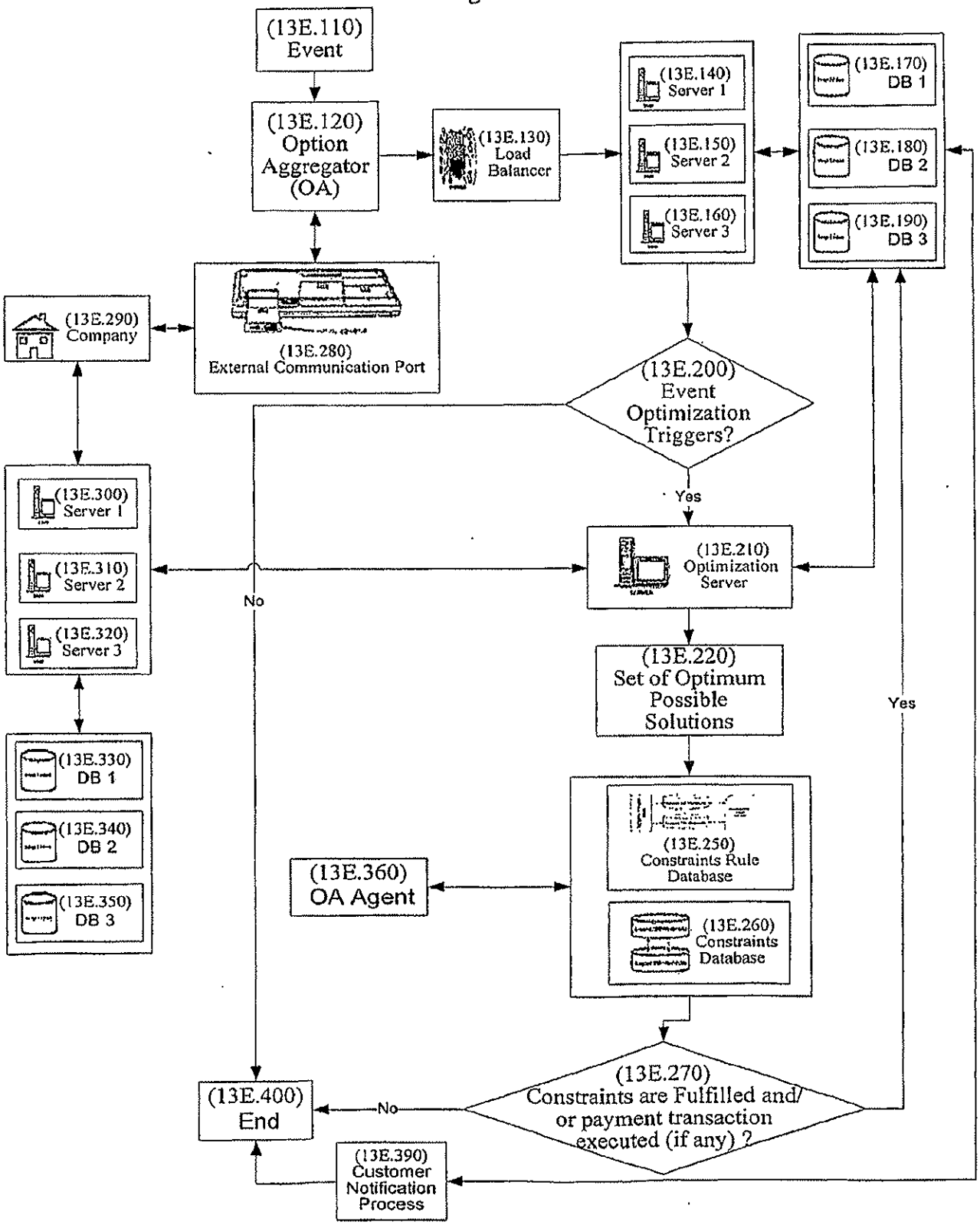

(13E.110) Event
(13E.120) Option Aggregator (OA)
(13E.130) Load Balancer
(13E.140) Server 1
(13E.150) Server 2
(13E.160) Server 3
(13E.170) DB 1
(13E.180) DB 2
(13E.190) DB 3
(13E.280) External Communication Port
(13E.290) Company
(13E.200) Event Optimization Triggers?
Yes
No
(13E.300) Server 1
(13E.310) Server 2
(13E.320) Server 3
(13E.210) Optimization Server
(13E.220) Set of Optimum Possible Solutions
Yes
(13E.330) DB 1
(13E.340) DB 2
(13E.350) DB 3
(13E.250) Constraints Rule Database
(13E.360) OA Agent
(13E.260) Constraints Database
(13E.270) Constraints are Fulfilled and/ or payment transaction executed (if any) ?
No
(13E.400) End
(13E.390) Customer Notification Process

Fig. 14

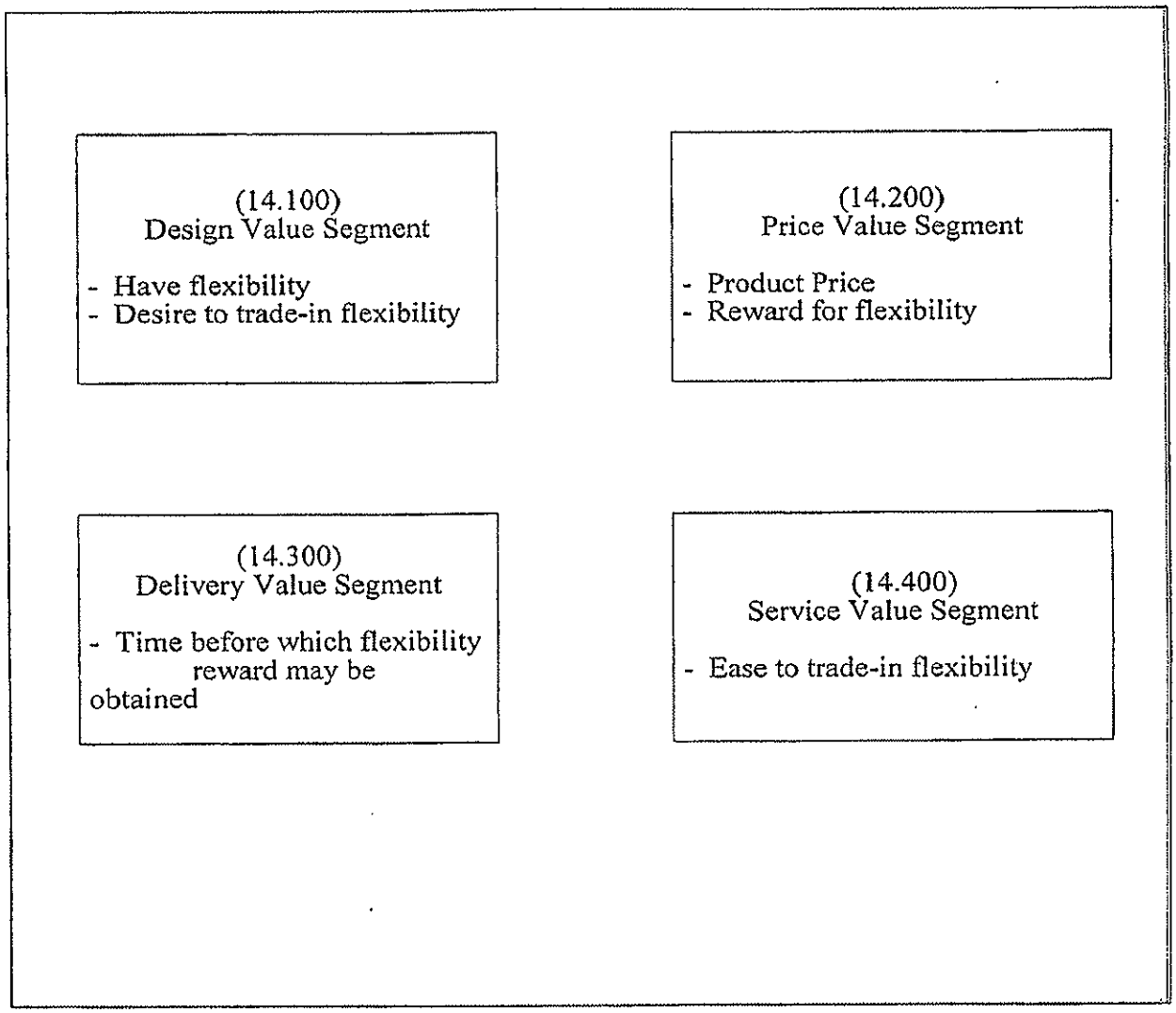
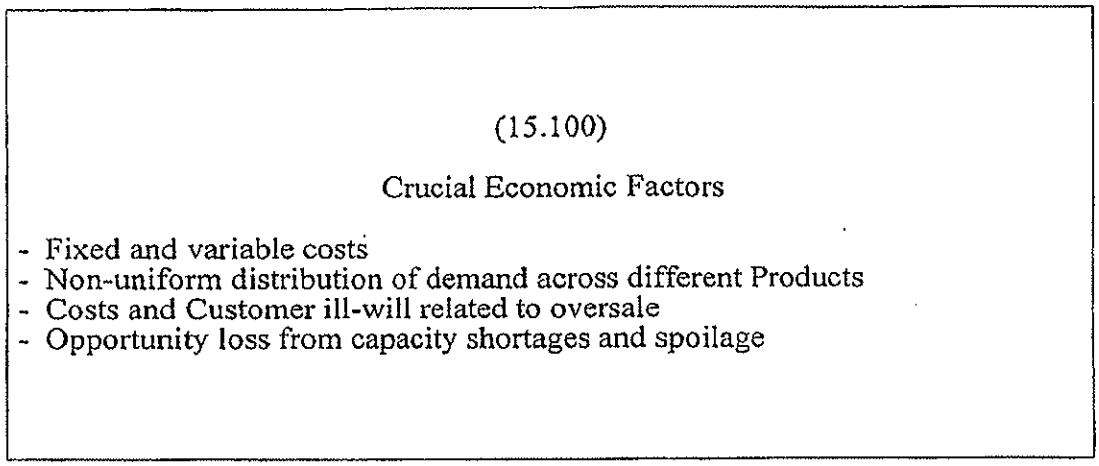

(14.100)
Design Value Segment
- Have flexibility
- Desire to trade-in flexibility
(14.200)
Price Value Segment
- Product Price
- Reward for flexibility
(14.300)
Delivery Value Segment
- Time before which flexibility reward may be obtained
(14.400)
Service Value Segment
- Ease to trade-in flexibility

Fig. 15


(15.100)
Crucial Economic Factors
- Fixed and variable costs
- Non-uniform distribution of demand across different Products
- Costs and Customer ill-will related to oversale
- Opportunity loss from capacity shortages and spoilage

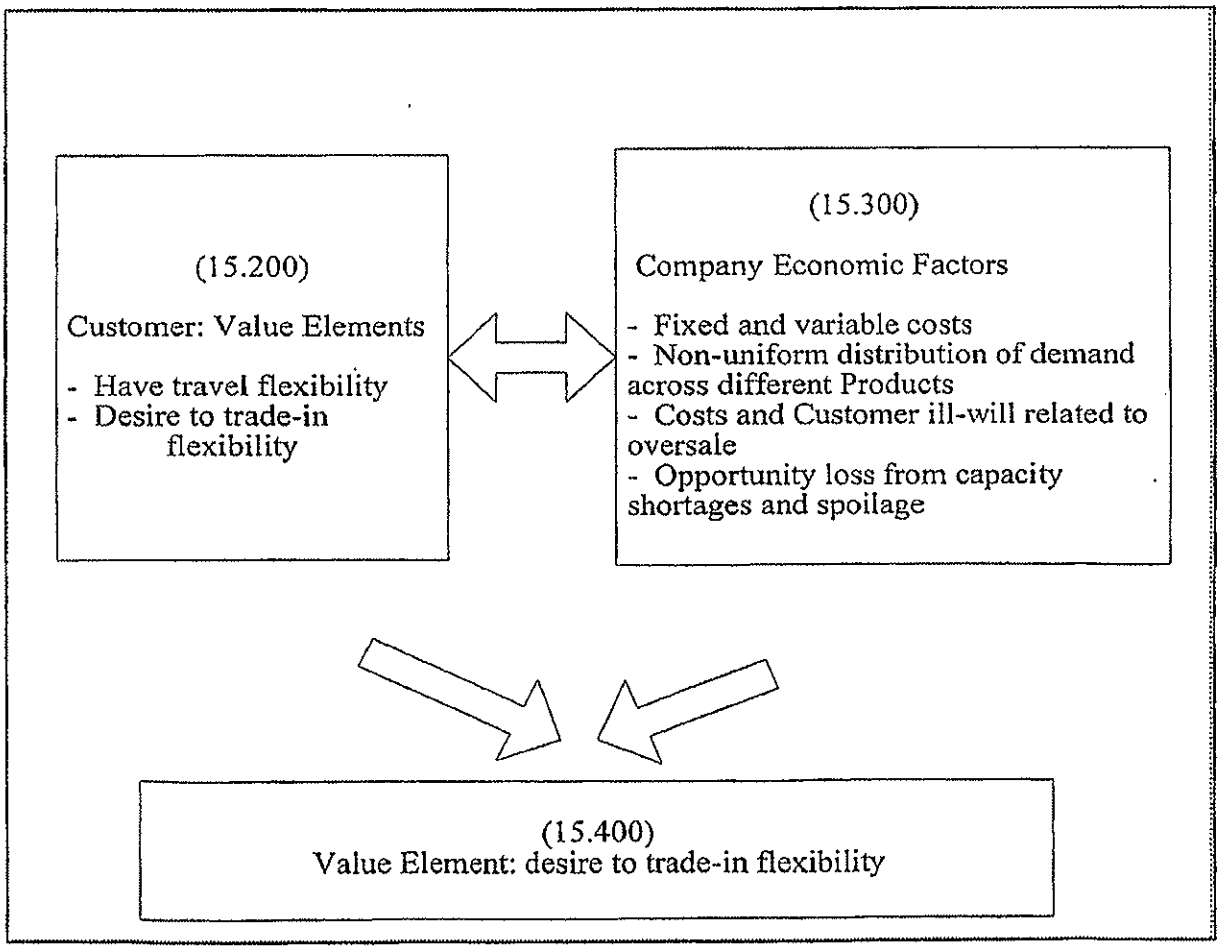

(15.200)
Customer: Value Elements
- Have travel flexibility
- Desire to trade-in flexibility
(15.300)
Company Economic Factors
- Fixed and variable costs
- Non-uniform distribution of demand across different Products
- Costs and Customer ill-will related to oversale
- Opportunity loss from capacity shortages and spoilage
(15.400)
Value Element: desire to trade-in flexibility

Fig. 16

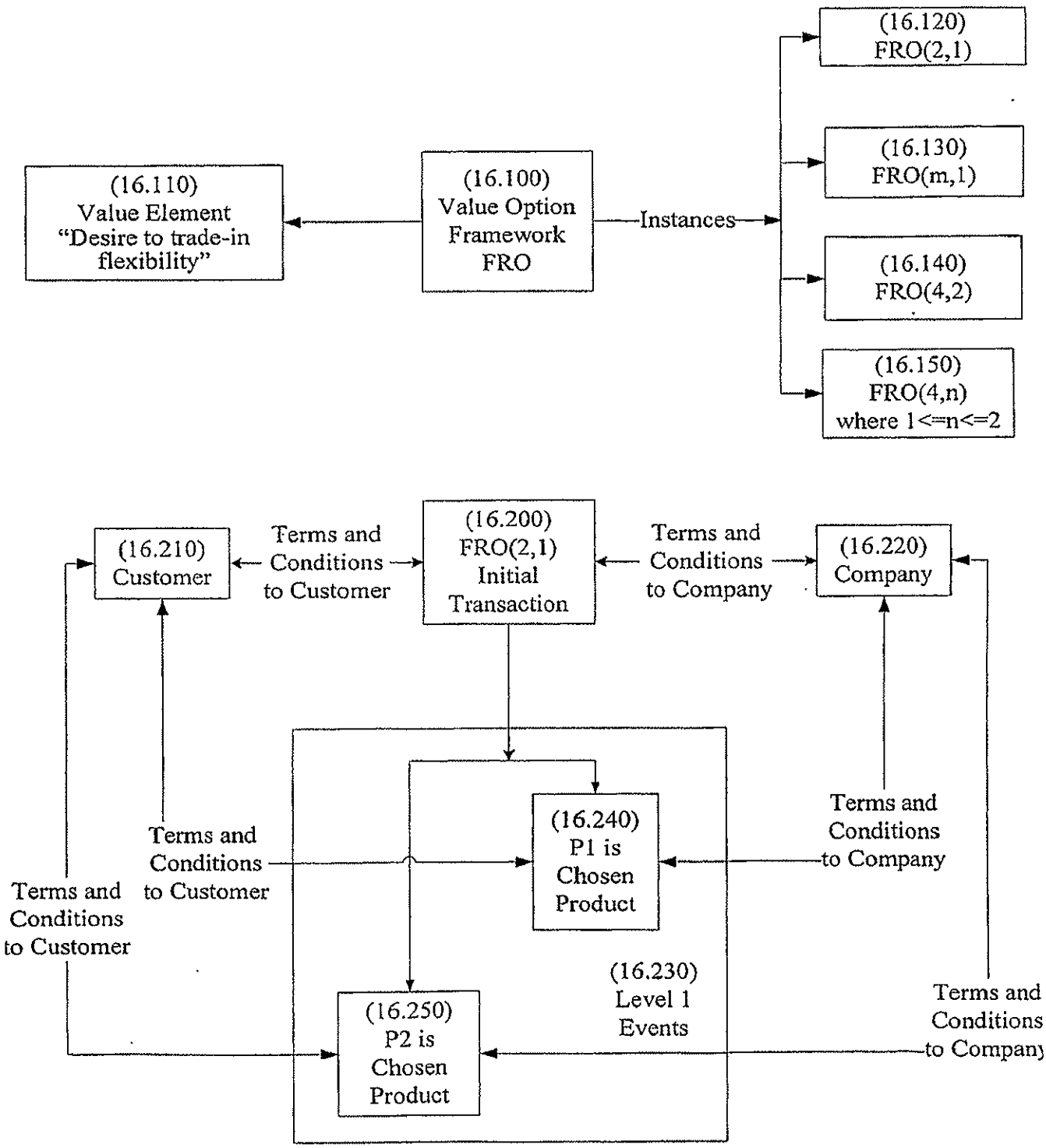

(16.120)
FRO(2,1)
(16.130)
FRO(m,1)
(16.140)
FRO(4,2)
(16.150)
FRO(4,n)
where 1<=n<=2
(16.110)
Value Element
"Desire to trade-in
flexibility"
(16.100)
Value Option
Framework
FRO
Instances
(16.210)
Customer
Terms and
Conditions
to Customer
(16.200)
FRO(2,1)
Initial
Transaction
Terms and
Conditions
to Company
(16.220)
Company
(16.240)
P1 is
Chosen
Product
Terms and
Conditions
to Customer
Terms and
Conditions
to Company
Terms and
Conditions
to Customer
(16.230)
Level 1
Events
(16.250)
P2 is
Chosen
Product
Terms and
Conditions
to Company

Fig. 17

(17.100) Product P1

XYZ Company	
Product P1	Utilization I 23rd May

(17.200) Product P2

XYZ Company	
Product P2	Utilization I 24th May

(17.300)

	(1)	(2)
(1)	Product Purchase Date Time	14th April 5:24 pm
(2)	APO Purchase Date Time	14th April 5:27 pm
(3)	Product Price Paid	$500
(4)	Initial FRO Discount	$15

(17.400)

Default Product P1

			(1)	(2)	(3)	(4)	(5)
(1)	Notify Deadline (DTU of P1)			30	7	3	1
(2)	FRO Exercise discount as a function of Chosen	P1		$5	$10	$15	$20
(3)	Product	P2		$10	$25	$35	$50

Fig. 18

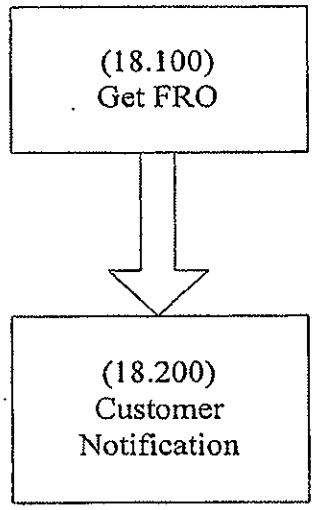


Fig. 19

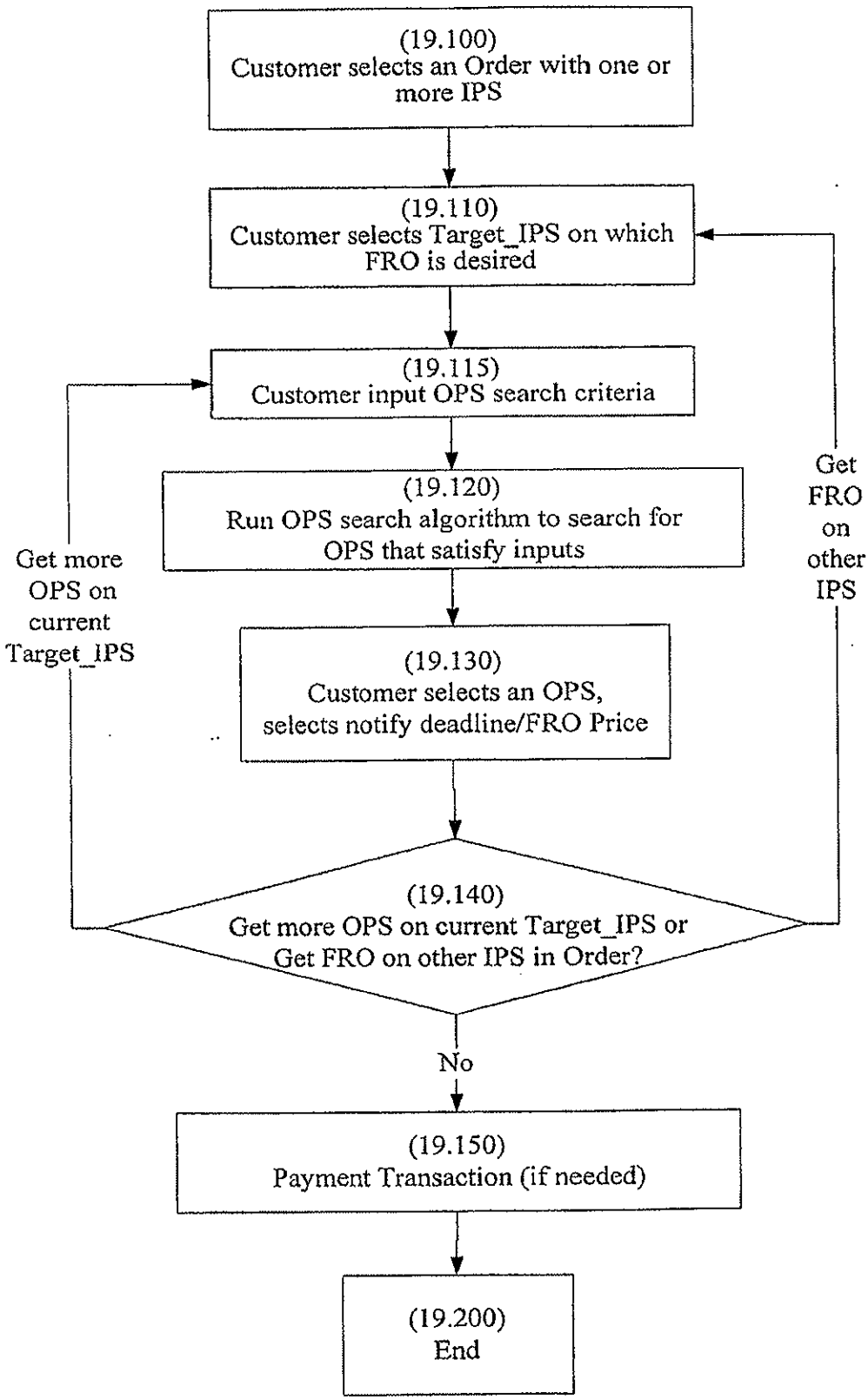

(19.100)
Customer selects an Order with one or more IPS
(19.110)
Customer selects Target_IPS on which FRO is desired
(19.115)
Customer input OPS search criteria
(19.120)
Run OPS search algorithm to search for OPS that satisfy inputs
(19.130)
Customer selects an OPS, selects notify deadline/FRO Price
(19.140)
Get more OPS on current Target_IPS or Get FRO on other IPS in Order?
Get more OPS on current Target_IPS
Get FRO on other IPS
No
(19.150)
Payment Transaction (if needed)
(19.200)
End

Fig. 20

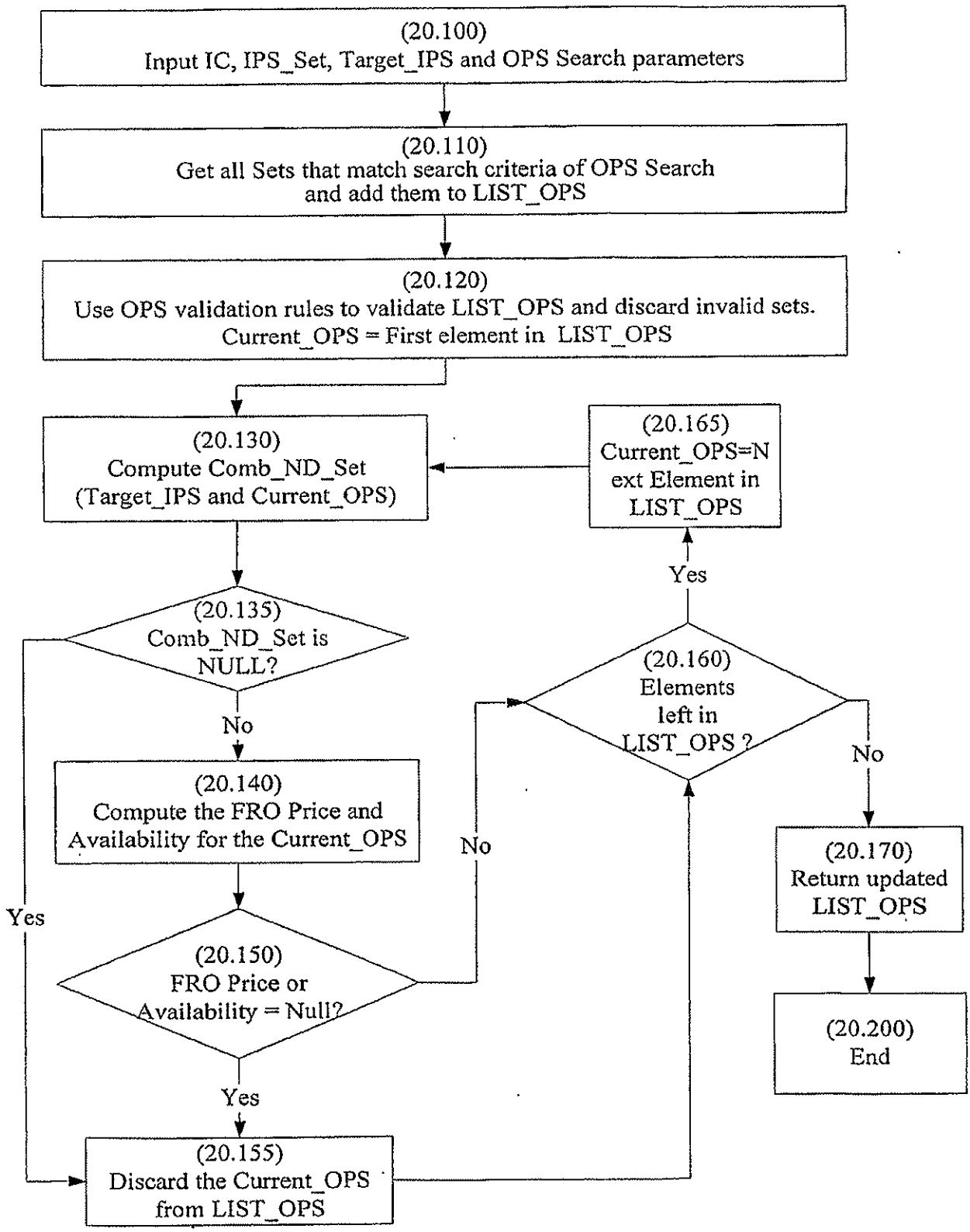

(20.100)
Input IC, IPS_Set, Target_IPS and OPS Search parameters
(20.110)
Get all Sets that match search criteria of OPS Search and add them to LIST_OPS
(20.120)
Use OPS validation rules to validate LIST_OPS and discard invalid sets.
Current_OPS = First element in LIST_OPS
(20.130)
Compute Comb_ND_Set (Target_IPS and Current_OPS)
(20.165)
Current_OPS=Next Element in LIST_OPS
(20.135)
Comb_ND_Set is NULL?
Yes
No
(20.140)
Compute the FRO Price and Availability for the Current_OPS
(20.160)
Elements left in LIST_OPS ?
Yes
No
No
(20.150)
FRO Price or Availability = Null?
Yes
(20.170)
Return updated LIST_OPS
(20.200)
End
(20.155)
Discard the Current_OPS from LIST_OPS

Fig. 21

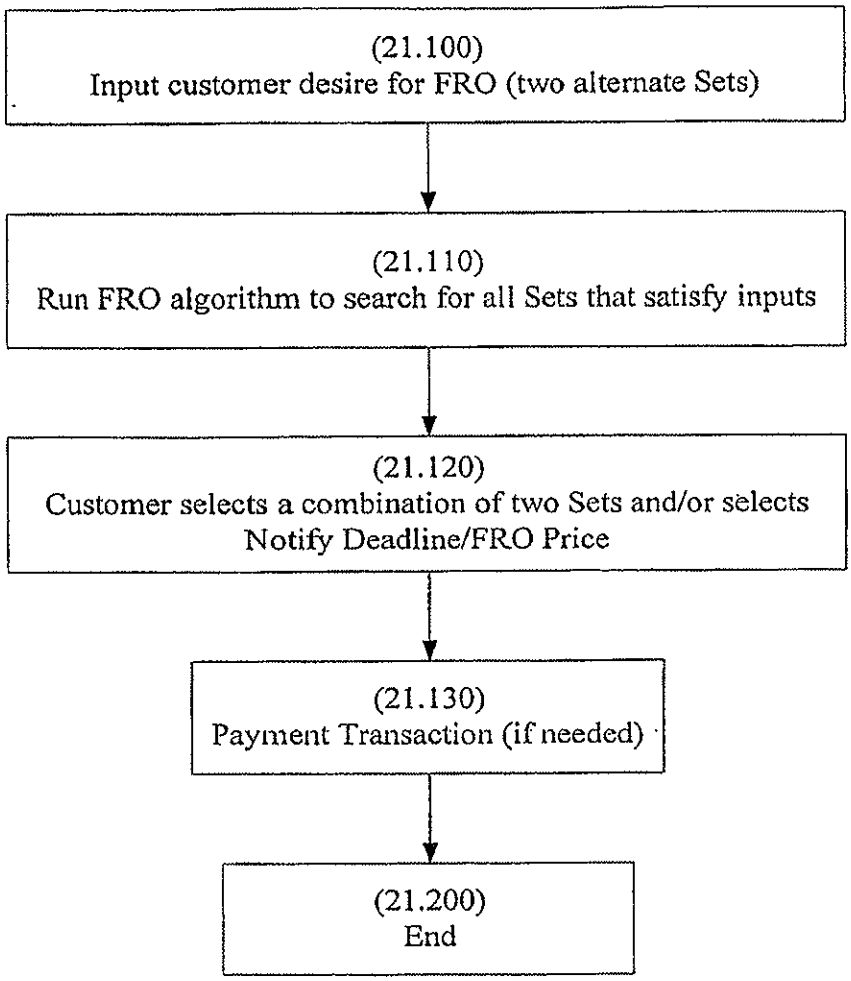

(21.100)
Input customer desire for FRO (two alternate Sets)
(21.110)
Run FRO algorithm to search for all Sets that satisfy inputs
(21.120)
Customer selects a combination of two Sets and/or selects
Notify Deadline/FRO Price
(21.130)
Payment Transaction (if needed)
(21.200)
End

Fig. 22

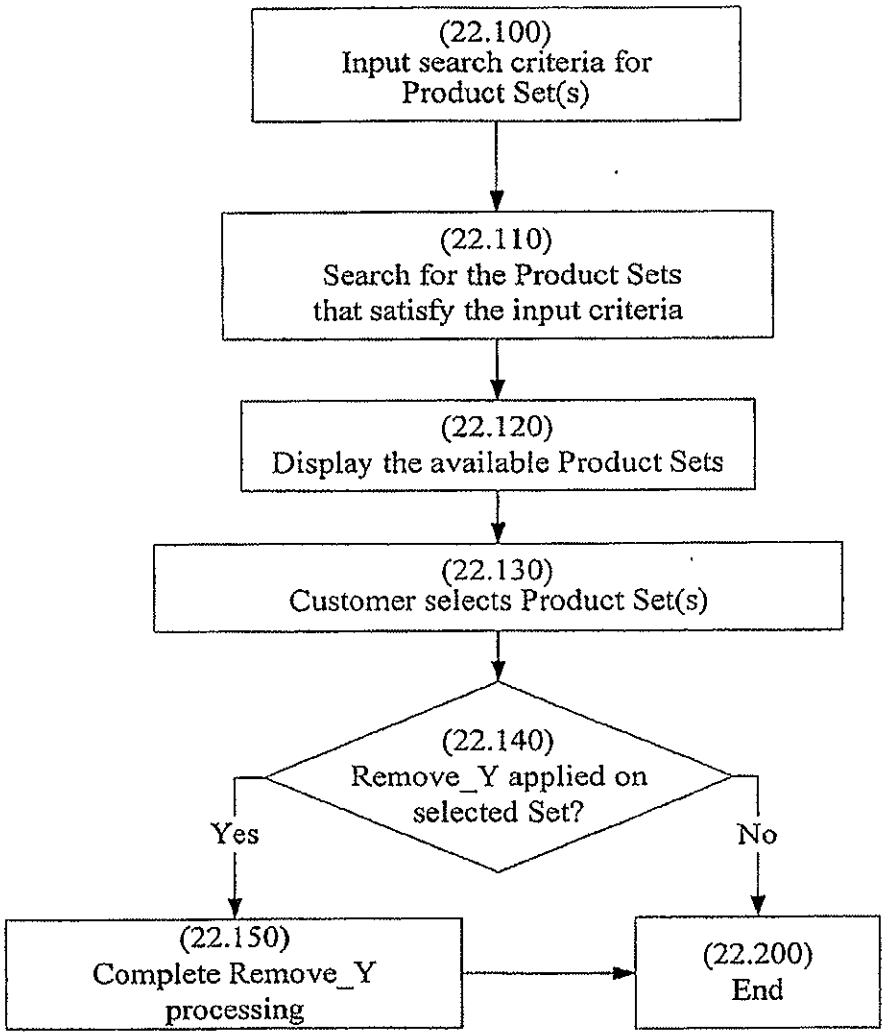

(22.100)
Input search criteria for Product Set(s)
(22.110)
Search for the Product Sets that satisfy the input criteria
(22.120)
Display the available Product Sets
(22.130)
Customer selects Product Set(s)
(22.140)
Remove_Y applied on selected Set?
Yes
No
(22.150)
Complete Remove_Y processing
(22.200)
End

Fig. 23

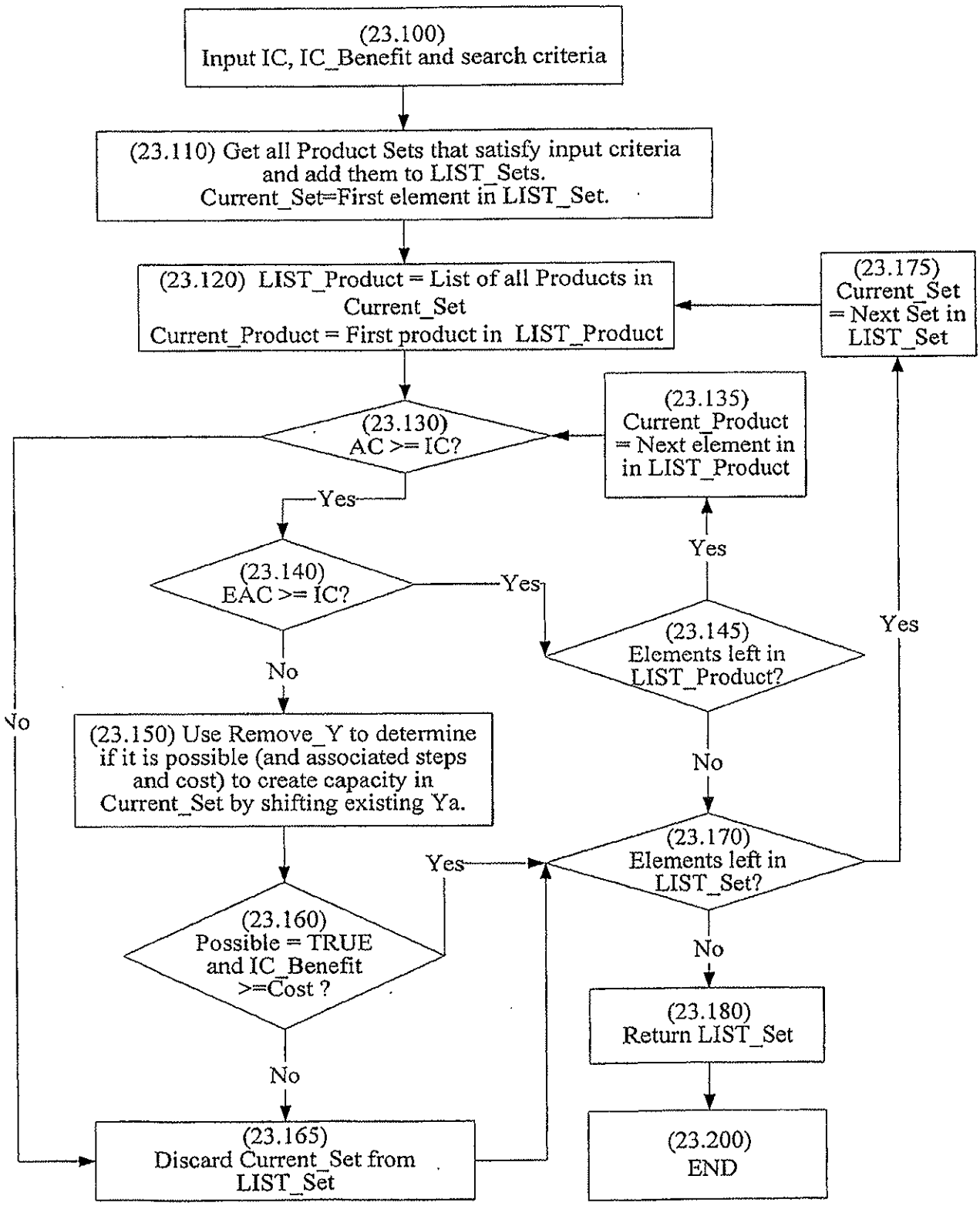

(23.100)
Input IC, IC_Benefit and search criteria
(23.110) Get all Product Sets that satisfy input criteria and add them to LIST_Sets.
Current_Set=First element in LIST_Set.
(23.120) LIST_Product = List of all Products in Current_Set
Current_Product = First product in LIST_Product
(23.175)
Current_Set = Next Set in LIST_Set
(23.135)
Current_Product = Next element in in LIST_Product
(23.130)
AC >= IC?
Yes
No
(23.140)
EAC >= IC?
Yes
No
Yes
(23.145)
Elements left in LIST_Product?
Yes
No
(23.150) Use Remove_Y to determine if it is possible (and associated steps and cost) to create capacity in Current_Set by shifting existing Ya.
(23.170)
Elements left in LIST_Set?
Yes
No
(23.160)
Possible = TRUE and IC_Benefit >=Cost ?
No
Yes
(23.180)
Return LIST_Set
(23.165)
Discard Current_Set from LIST_Set
(23.200)
END

Fig. 24

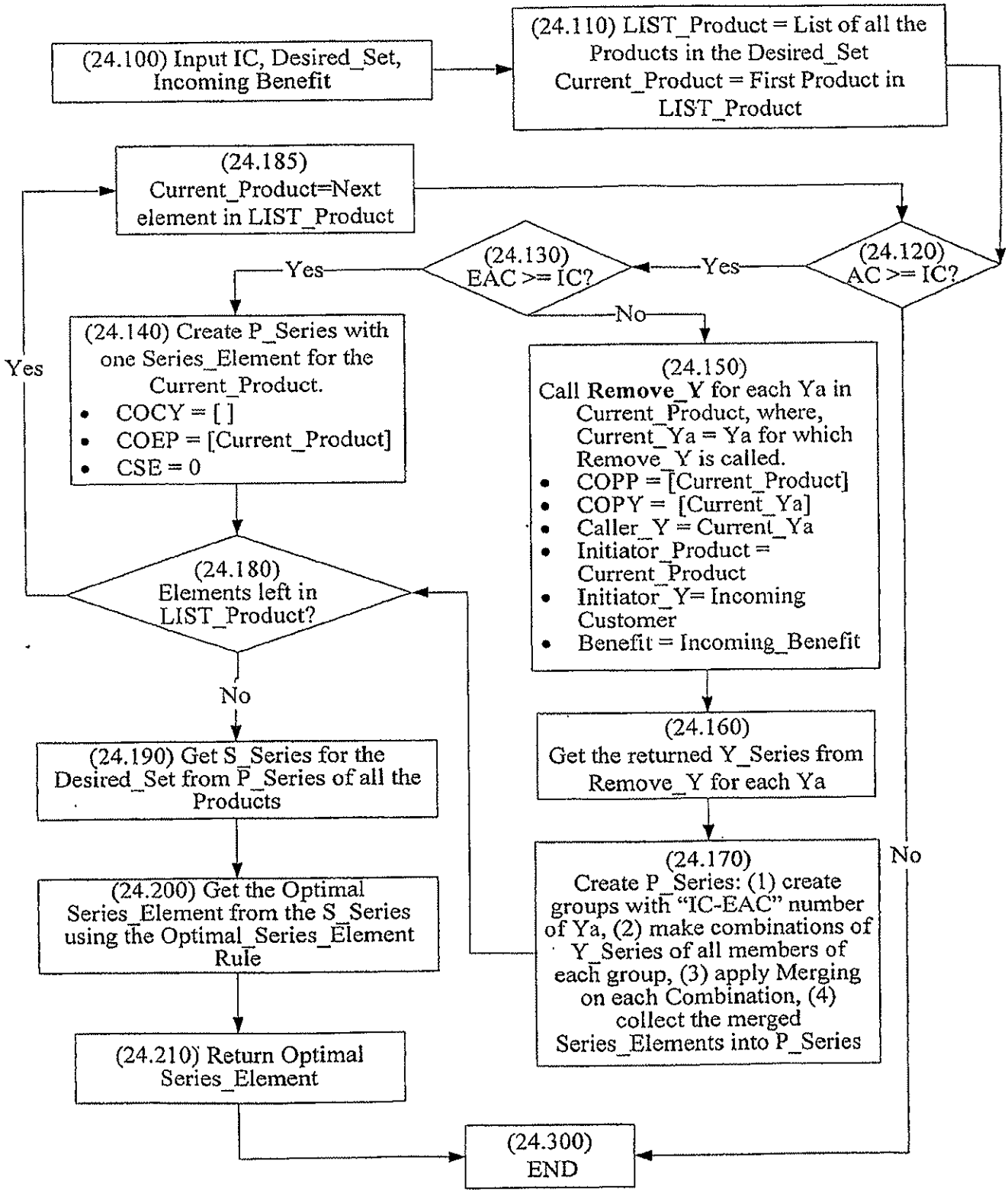

(24.100) Input IC, Desired_Set, Incoming Benefit
(24.110) LIST_Product = List of all the Products in the Desired_Set Current_Product = First Product in LIST_Product
(24.185) Current_Product=Next element in LIST_Product
(24.120) AC >= IC?
Yes
(24.130) EAC >= IC?
Yes
No
(24.140) Create P_Series with one Series_Element for the Current_Product.
• COCY = [ ]
• COEP = [Current_Product]
• CSE = 0
(24.150) Call Remove_Y for each Ya in Current_Product, where, Current_Ya = Ya for which Remove_Y is called.
• COPP = [Current_Product]
• COPY = [Current_Ya]
• Caller_Y = Current_Ya
• Initiator_Product = Current_Product
• Initiator_Y= Incoming Customer
• Benefit = Incoming_Benefit
Yes
(24.180) Elements left in LIST_Product?
No
(24.160) Get the returned Y_Series from Remove_Y for each Ya
(24.170) Create P_Series: (1) create groups with "IC-EAC" number of Ya, (2) make combinations of Y_Series of all members of each group, (3) apply Merging on each Combination, (4) collect the merged Series_Elements into P_Series
No
(24.190) Get S_Series for the Desired_Set from P_Series of all the Products
(24.200) Get the Optimal Series_Element from the S_Series using the Optimal_Series_Element Rule
(24.210) Return Optimal Series_Element
(24.300) END

Fig. 25

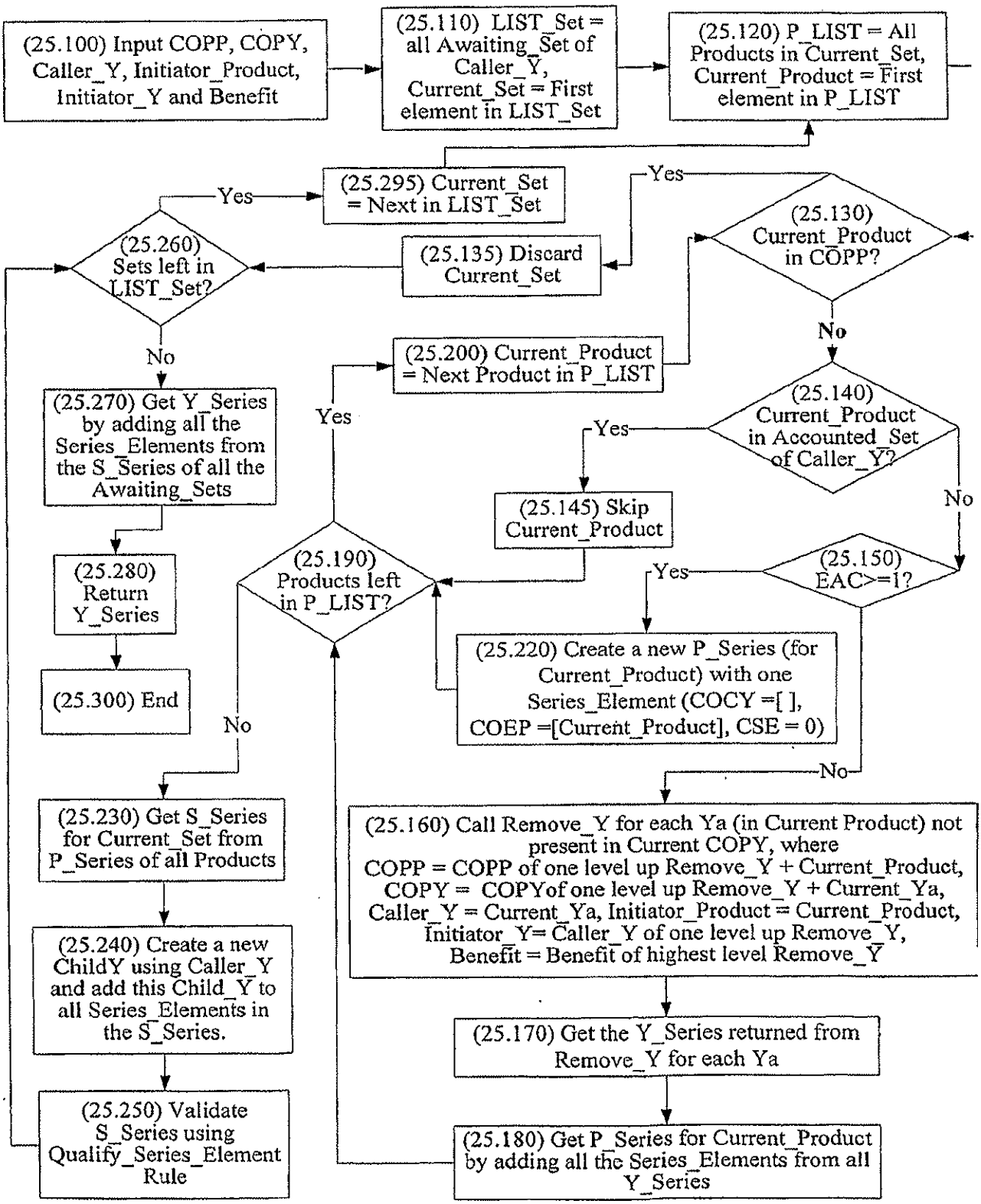

(25.100) Input COPP, COPY, Caller_Y, Initiator_Product, Initiator_Y and Benefit
(25.110) LIST_Set = all Awaiting_Set of Caller_Y, Current_Set = First element in LIST_Set
(25.120) P_LIST = All Products in Current_Set, Current_Product = First element in P_LIST
(25.295) Current_Set = Next in LIST_Set
Yes
(25.260) Sets left in LIST_Set?
(25.135) Discard Current_Set
Yes
(25.130) Current_Product in COPP?
No
(25.200) Current_Product = Next Product in P_LIST
Yes
(25.140) Current_Product in Accounted_Set of Caller_Y?
Yes
(25.145) Skip Current_Product
No
(25.150) EAC>=1?
Yes
(25.220) Create a new P_Series (for Current_Product) with one Series_Element (COCY =[ ], COEP =[Current_Product], CSE = 0)
No
(25.160) Call Remove_Y for each Ya (in Current Product) not present in Current COPY, where COPP = COPP of one level up Remove_Y + Current_Product, COPY = COPYof one level up Remove_Y + Current_Ya, Caller_Y = Current_Ya, Initiator_Product = Current_Product, Initiator_Y= Caller_Y of one level up Remove_Y, Benefit = Benefit of highest level Remove_Y
(25.170) Get the Y_Series returned from Remove_Y for each Ya
(25.180) Get P_Series for Current_Product by adding all the Series_Elements from all Y_Series
(25.190) Products left in P_LIST?
Yes
No
(25.270) Get Y_Series by adding all the Series_Elements from the S_Series of all the Awaiting_Sets
(25.280) Return Y_Series
(25.300) End
No
(25.230) Get S_Series for Current_Set from P_Series of all Products
(25.240) Create a new ChildY using Caller_Y and add this Child_Y to all Series_Elements in the S_Series.
(25.250) Validate S_Series using Qualify_Series_Element Rule

Fig. 26

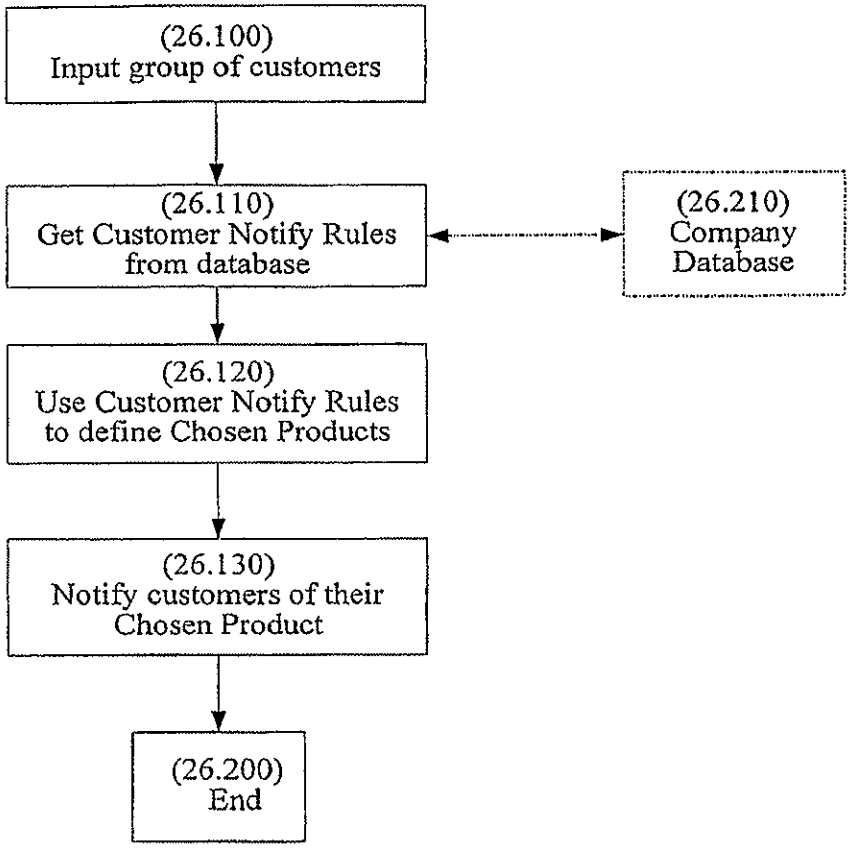

(26.100)
Input group of customers
(26.110)
Get Customer Notify Rules from database
(26.210)
Company Database
(26.120)
Use Customer Notify Rules to define Chosen Products
(26.130)
Notify customers of their Chosen Product
(26.200)
End

Fig. 27

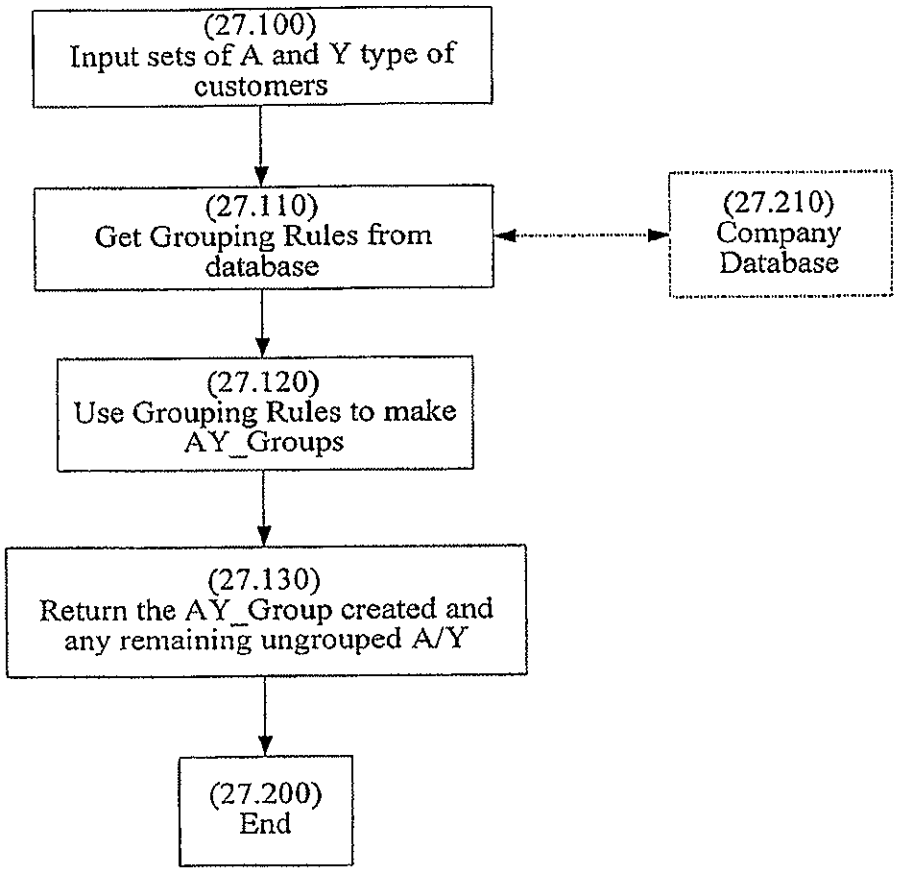

(27.100)
Input sets of A and Y type of customers
(27.110)
Get Grouping Rules from database
(27.210)
Company Database
(27.120)
Use Grouping Rules to make AY_Groups
(27.130)
Return the AY_Group created and any remaining ungrouped A/Y
(27.200)
End

Fig. 28

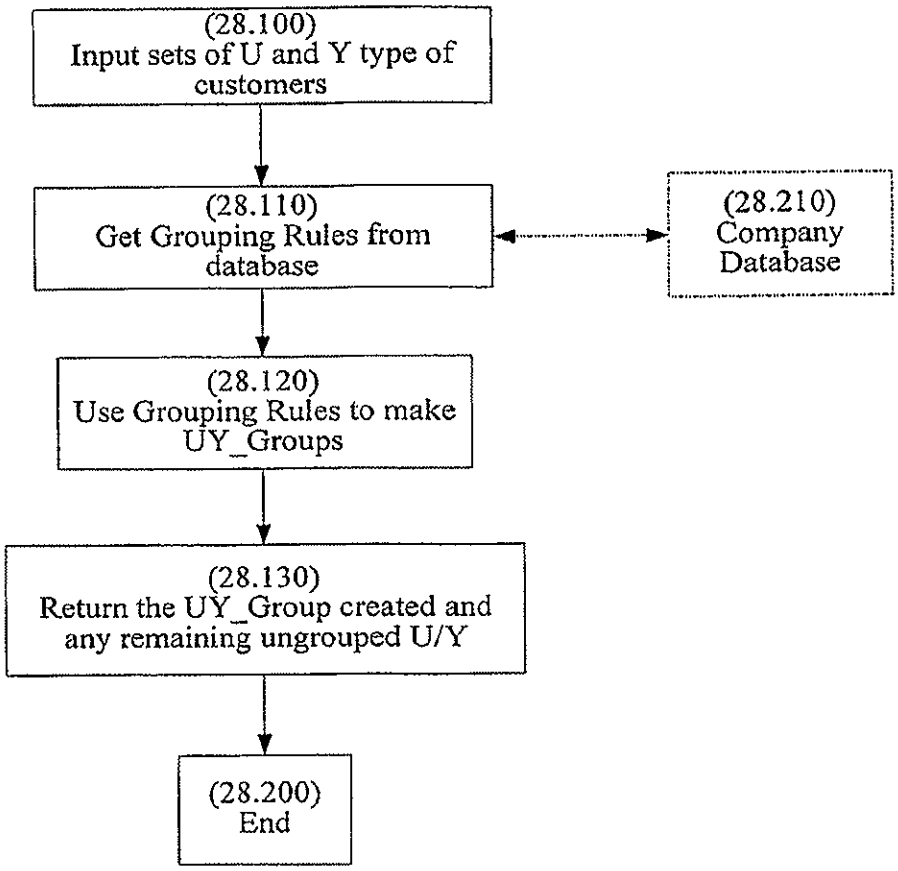

(28.100)
Input sets of U and Y type of customers
(28.110)
Get Grouping Rules from database
(28.210)
Company Database
(28.120)
Use Grouping Rules to make UY_Groups
(28.130)
Return the UY_Group created and any remaining ungrouped U/Y
(28.200)
End

Fig. 29

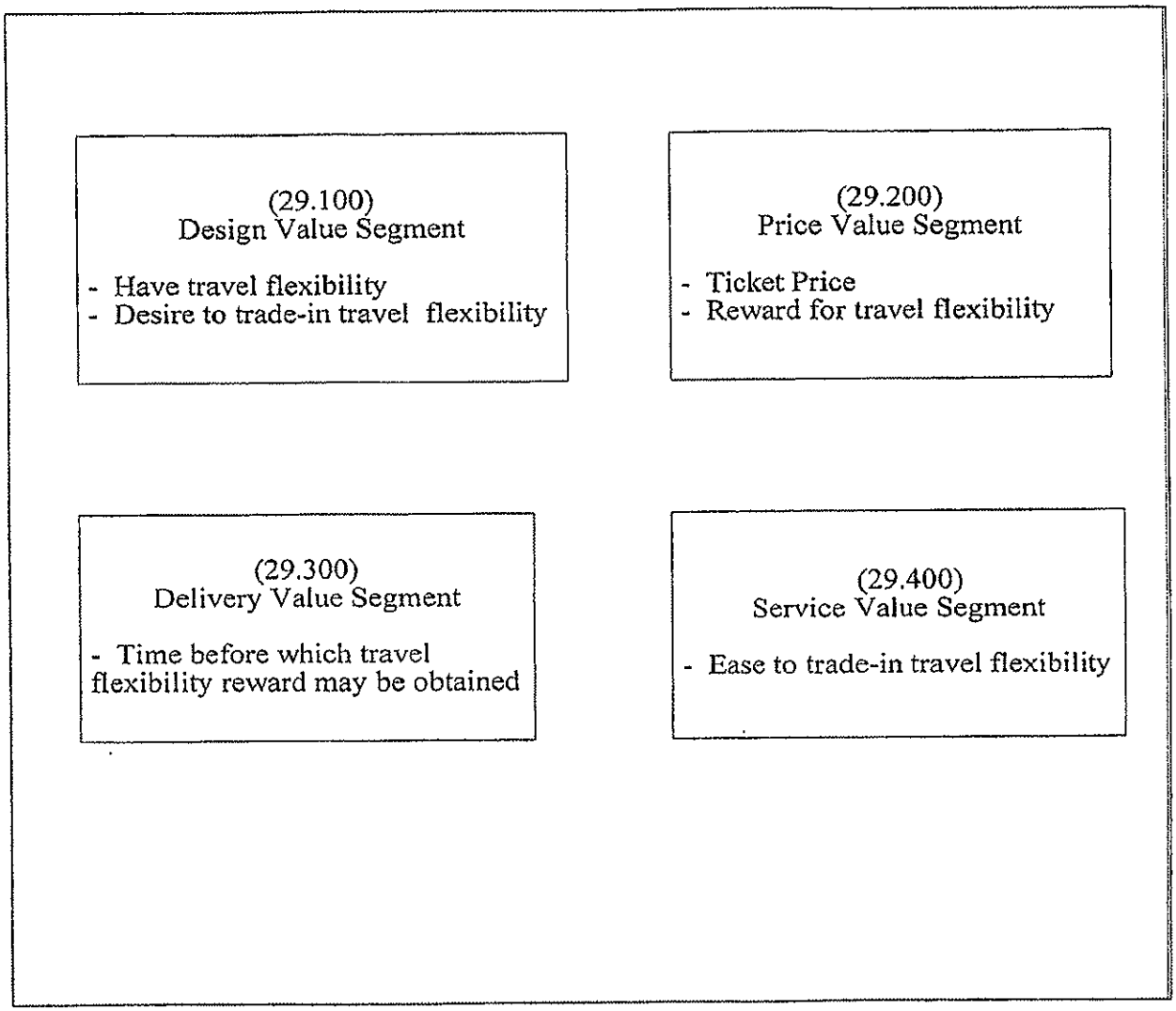

(29.100)
Design Value Segment
- Have travel flexibility
- Desire to trade-in travel flexibility
(29.200)
Price Value Segment
- Ticket Price
- Reward for travel flexibility
(29.300)
Delivery Value Segment
- Time before which travel flexibility reward may be obtained
(29.400)
Service Value Segment
- Ease to trade-in travel flexibility

Fig. 30

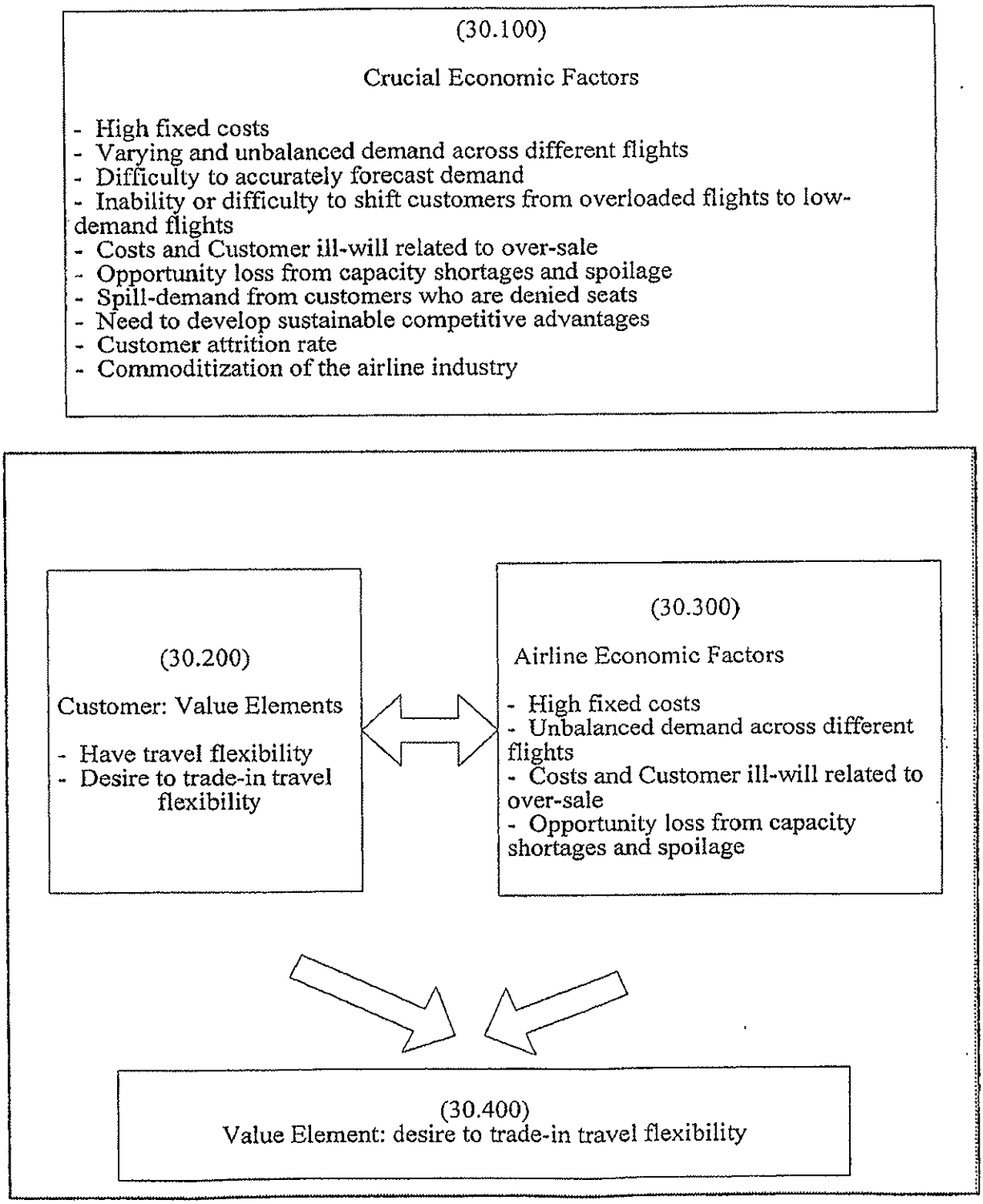

(30.100)
Crucial Economic Factors
- High fixed costs
- Varying and unbalanced demand across different flights
- Difficulty to accurately forecast demand
- Inability or difficulty to shift customers from overloaded flights to low-demand flights
- Costs and Customer ill-will related to over-sale
- Opportunity loss from capacity shortages and spoilage
- Spill-demand from customers who are denied seats
- Need to develop sustainable competitive advantages
- Customer attrition rate
- Commoditization of the airline industry
(30.200)
Customer: Value Elements
- Have travel flexibility
- Desire to trade-in travel flexibility
(30.300)
Airline Economic Factors
- High fixed costs
- Unbalanced demand across different flights
- Costs and Customer ill-will related to over-sale
- Opportunity loss from capacity shortages and spoilage
(30.400)
Value Element: desire to trade-in travel flexibility

Fig. 31

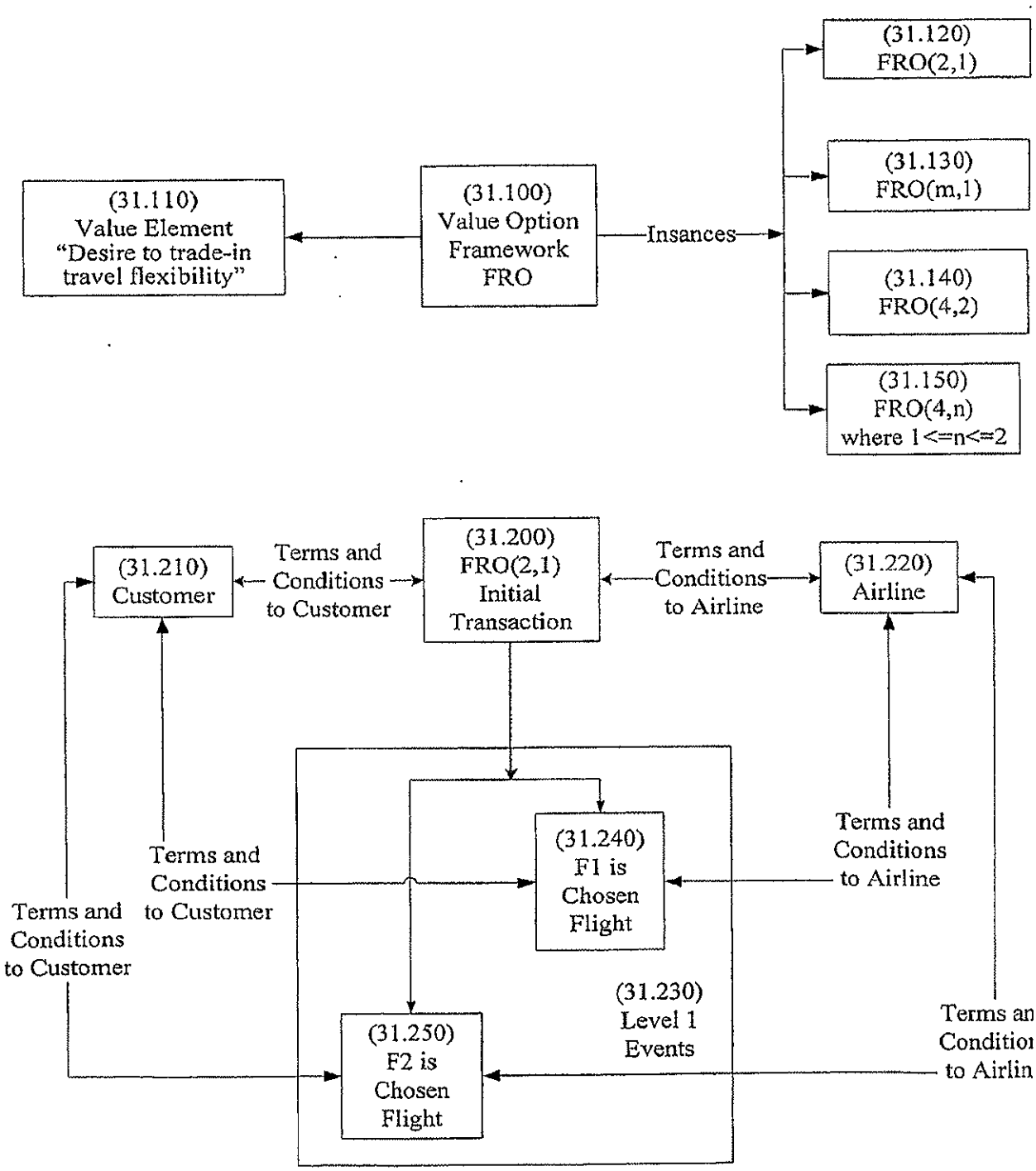

(31.110)
Value Element
"Desire to trade-in
travel flexibility"
(31.100)
Value Option
Framework
FRO
Insances
(31.120)
FRO(2,1)
(31.130)
FRO(m,1)
(31.140)
FRO(4,2)
(31.150)
FRO(4,n)
where 1<=n<=2
(31.210)
Customer
Terms and
Conditions
to Customer
(31.200)
FRO(2,1)
Initial
Transaction
Terms and
Conditions
to Airline
(31.220)
Airline
Terms and
Conditions
to Customer
Terms and
Conditions
to Customer
(31.240)
F1 is
Chosen
Flight
Terms and
Conditions
to Airline
(31.230)
Level 1
Events
(31.250)
F2 is
Chosen
Flight
Terms an
Conditio
to Airlin

Fig. 32

(32.100) Flight F1

Departs	Arrives	Flight	Cabin
BOS	ATL	Alpha 661	Coach
23rd May	23rd May	0 Stops	
2:00pm	5:00pm		

(32.200) Flight F2

Departs	Arrives	Flight	Cabin
BOS	ATL	Alpha 427	Coach
25th May	25th May	0 Stops	
4:30pm	7:30pm		

(32.300)	*(1)*	*(2)*
(1)	Ticket Purchase Date Time	14th April 5:24 pm
(2)	FRO Initial Transaction Date Time	14th April 5:27 pm
(3)	Ticket Price Paid	$500
(4)	Initial FRO Discount	$15

(32.400)

Default Flight F1

			(1)	*(2)*	*(3)*	*(4)*	*(5)*
(1)	Notify Deadline (DTD of F1)			30	7	3	1
(2)	FRO Exercise discount as a function of Chosen	F1		$5	$10	$15	$20
(3)	Flight	F2		$10	$25	$35	$50

Fig. 33

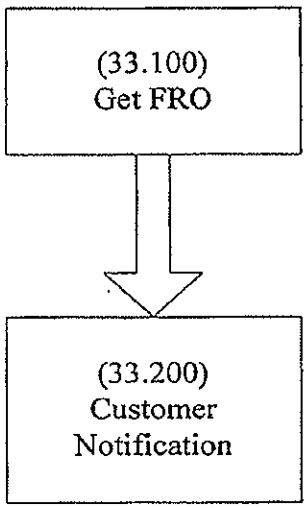


Fig. 34

FRO Itinerary 1 (34.100)

Onward journey

Flight	Departs	Arrives
Alpha993 Coach(U)	BOS 02 May 8:55am	ATL 02 May 11:57am
Alpha998 Coach(U)	ATL 02 May 1:00pm	LAX 02 May 3:07pm

Return journey

Flight	Departs	Arrives
Alpha893 Coach(U)	LAX 06 May 8:55am	BOS 06 May 12:15pm

FRO Itinerary 2 (34.200)

Onward journey

Flight	Departs	Arrives
Alpha963 Coach(U)	BOS 04 May 8:55am	ATL 04 May 11:57am
Alpha969 Coach(U)	ATL 04 May 2:00pm	LAX 04 May 4:07pm

Return journey

Flight	Departs	Arrives
Alpha793 Coach(U)	LAX 07 May 9:55am	BOS 07 May 1:15pm

Fig. 35

Onward journey (35.100)

	Flight	Departs	Arrives
(35.110) Initial Flight Segment	Alpha567 Coach(U)	BOS 02 May 8:55am	ATL 02 May 11:57am
	Alpha678 Coach(U)	ATL 02 May 1:00pm	LAX 02 May 3:07pm

(35.120)	Flight	Departs	Arrives
Option Flight Segment	Alpha123 Coach(U)	BOS 03 May 8:55am	LAX 03 May 12:07pm

Return journey (20.200)

(35.360)	Flight	Departs	Arrives
Initial Flight Segment	Alpha233 Coach(U)	LAX 06 May 8:55am	BOS 06 May 6:15pm

Fig. 36

Onward journey (36.100)

Leg 1

(36.110)	Flight	Departs	Arrives
Initial Flight Leg	Alpha493 Coach(U)	BOS 02 May 8:55am	ATL 02 May 11:57am

Leg 2 (36.200)

(36.210)	Flight	Departs	Arrives
Initial Flight Leg	Alpha398 Coach(U)	ATL 02 May 1:00pm	LAX 02 May 3:07pm

(36.220)	Flight	Departs	Arrives
Option Flight Leg	Alpha253 Coach(U)	ATL 02 May 6:00pm	LAX 02 May 8:07pm

Return journey (36.300)

Leg1

(36.310)	Flight	Departs	Arrives
Initial Flight Leg	Alpha343 Coach(U)	LAX 06 May 8:55am	BOS 06 May 6:15pm

Fig. 37

FRO Reservations

Alternate Flight Option
Know More... Buy AFO

Upgrade Ticket Option
Know More... Buy UTO

Flixibility Reward Option
Know More... Get FRO

Your Itinerary

Onward Journey

Flight	Departs	Arrives
Alpha 993 Coach (U)	BOS 8:55 am May 2	ORD 11:57 am May 2
Alpha 892 Coach (U)	ORD 1:00 pm May 2	ATL 3:07 pm May 2

Return Journey

Flight	Departs	Arrives
Alpha 964 Coach (U)	ATL 9:55 am May 7	ORD 11:50 pm May 7
Alpha 857 Coach (U)	ORD 1:00 pm May 7	BOS 4:07 pm May 7

Fig. 38

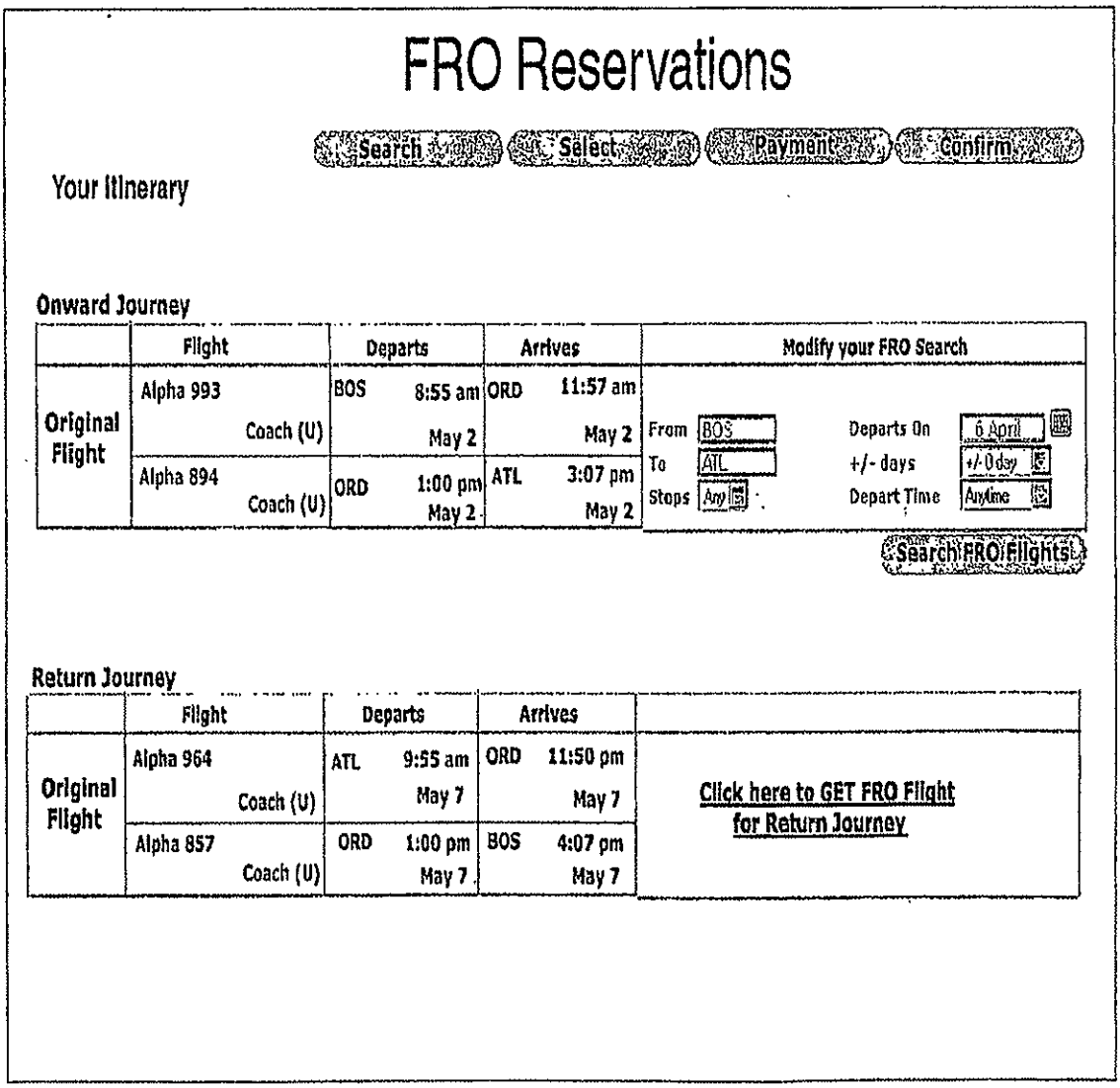

FRO Reservations
Search
Select
Payment
Confirm
Your Itinerary
Onward Journey
Flight
Departs
Arrives
Modify your FRO Search
Original Flight
Alpha 993 Coach (U)
BOS 8:55 am May 2
ORD 11:57 am May 2
Alpha 894 Coach (U)
ORD 1:00 pm May 2
ATL 3:07 pm May 2
From BOS
To ATL
Stops Any
Departs On 6 April
+/- days +/- 0 day
Depart Time Anytime
Search FRO Flights
Return Journey
Flight
Departs
Arrives
Original Flight
Alpha 964 Coach (U)
ATL 9:55 am May 7
ORD 11:50 pm May 7
Alpha 857 Coach (U)
ORD 1:00 pm May 7
BOS 4:07 pm May 7
Click here to GET FRO Flight for Return Journey

Fig. 39

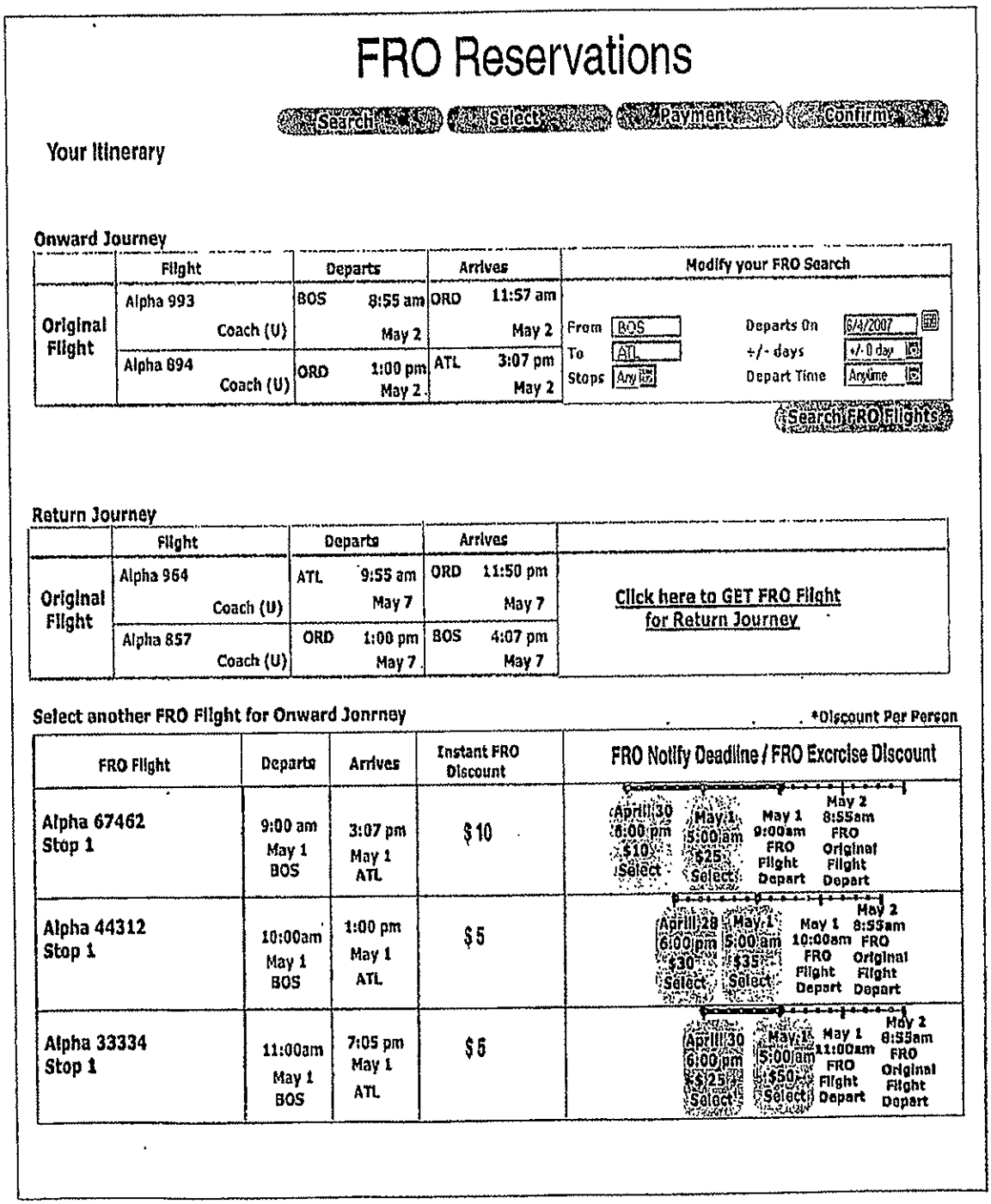

FRO Reservations
Search
Select
Payment
Confirm
Your Itinerary
Onward Journey
Flight
Departs
Arrives
Modify your FRO Search
Original Flight
Alpha 993 Coach (U)
BOS 8:55 am May 2
ORD 11:57 am May 2
Alpha 894 Coach (U)
ORD 1:00 pm May 2
ATL 3:07 pm May 2
From BOS
To ATL
Stops Any
Departs On 6/4/2007
+/- days +/- 0 day
Depart Time Anytime
Search FRO Flights
Return Journey
Flight
Departs
Arrives
Original Flight
Alpha 964 Coach (U)
ATL 9:55 am May 7
ORD 11:50 pm May 7
Alpha 857 Coach (U)
ORD 1:00 pm May 7
BOS 4:07 pm May 7
Click here to GET FRO Flight for Return Journey
Select another FRO Flight for Onward Journey
*Discount Per Person
FRO Flight
Departs
Arrives
Instant FRO Discount
FRO Notify Deadline / FRO Exercise Discount
Alpha 67462 Stop 1
9:00 am May 1 BOS
3:07 pm May 1 ATL
$10
April 30 6:00 pm $10 Select
May 1 5:00 am $25 Select
May 1 9:00am FRO Flight Depart
May 2 8:55am FRO Original Flight Depart
Alpha 44312 Stop 1
10:00am May 1 BOS
1:00 pm May 1 ATL
$5
April 28 6:00 pm $30 Select
May 1 5:00 am $35 Select
May 1 10:00am FRO Flight Depart
May 2 8:55am FRO Original Flight Depart
Alpha 33334 Stop 1
11:00am May 1 BOS
7:05 pm May 1 ATL
$5
April 30 6:00 pm $25 Select
May 1 5:00 am $50 Select
May 1 11:00am FRO Flight Depart
May 2 8:55am FRO Original Flight Depart

Fig. 40

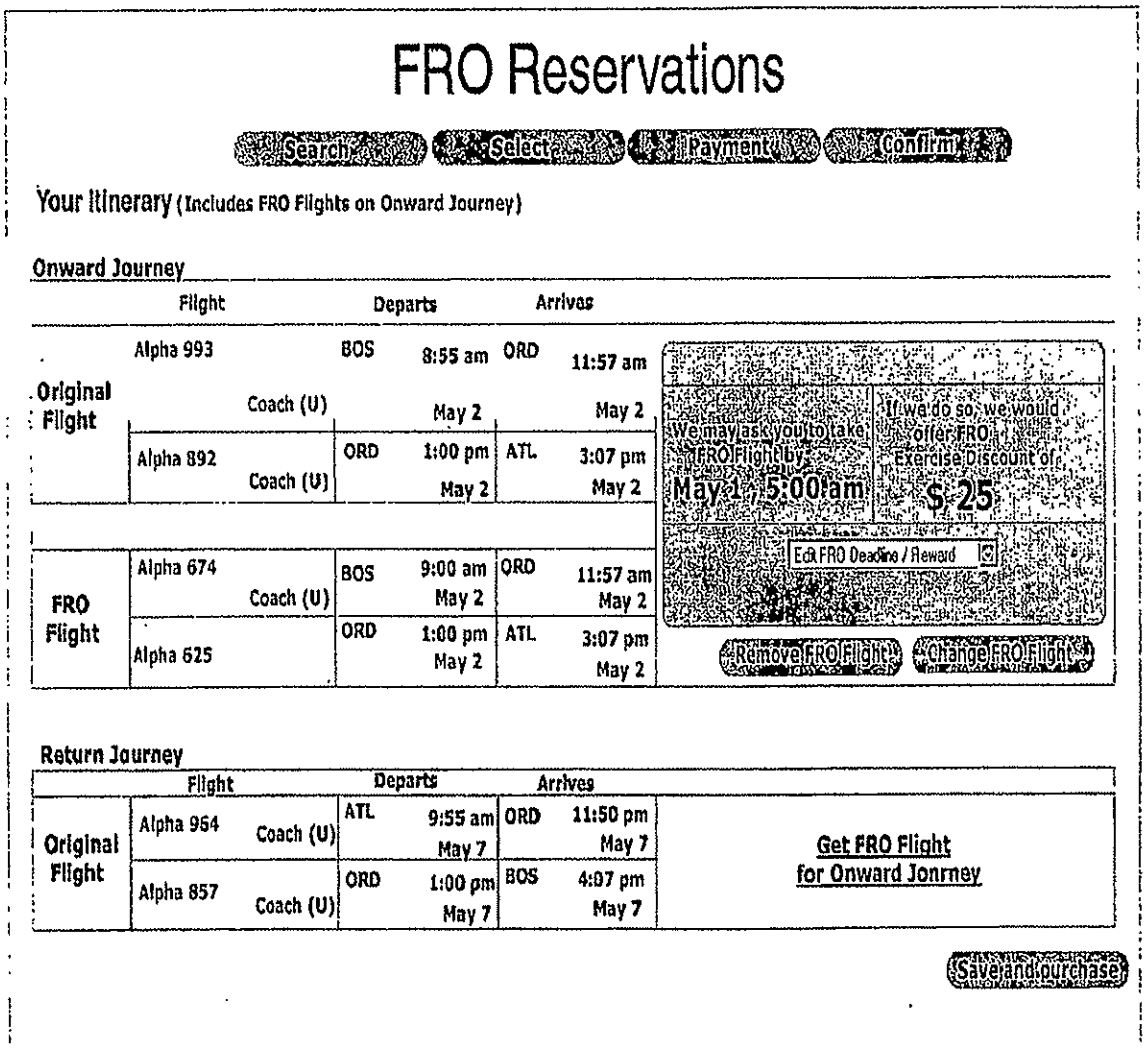

FRO Reservations
Search
Select
Payment
Confirm
Your Itinerary (Includes FRO Flights on Onward Journey)
Onward Journey
Flight
Departs
Arrives
Original Flight
Alpha 993 Coach (U)
BOS 8:55 am May 2
ORD 11:57 am May 2
Alpha 892 Coach (U)
ORD 1:00 pm May 2
ATL 3:07 pm May 2
FRO Flight
Alpha 674 Coach (U)
BOS 9:00 am May 2
ORD 11:57 am May 2
Alpha 625
ORD 1:00 pm May 2
ATL 3:07 pm May 2
We may ask you to take FRO Flight by:
May 1, 5:00 am
If we do so, we would offer FRO Exercise Discount of:
$ 25
Edit FRO Deadline / Reward
Remove FRO Flight
Change FRO Flight
Return Journey
Flight
Departs
Arrives
Original Flight
Alpha 964 Coach (U)
ATL 9:55 am May 7
ORD 11:50 pm May 7
Alpha 857 Coach (U)
ORD 1:00 pm May 7
BOS 4:07 pm May 7
Get FRO Flight for Onward Journey
Save and purchase

Fig. 41

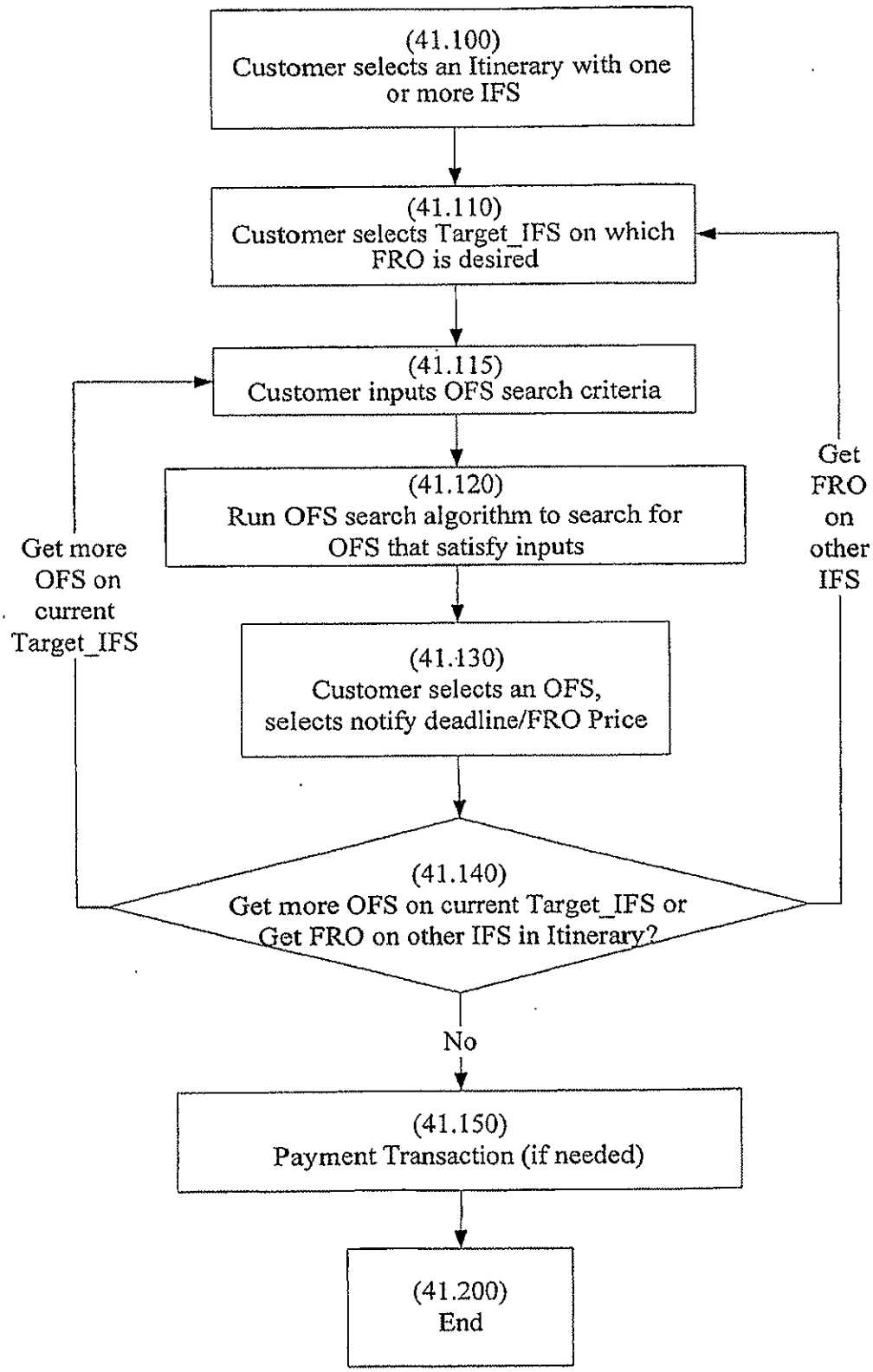

(41.100)
Customer selects an Itinerary with one or more IFS
(41.110)
Customer selects Target_IFS on which FRO is desired
(41.115)
Customer inputs OFS search criteria
(41.120)
Run OFS search algorithm to search for OFS that satisfy inputs
(41.130)
Customer selects an OFS, selects notify deadline/FRO Price
(41.140)
Get more OFS on current Target_IFS or Get FRO on other IFS in Itinerary?
Get more OFS on current Target_IFS
Get FRO on other IFS
No
(41.150)
Payment Transaction (if needed)
(41.200)
End

Fig. 42

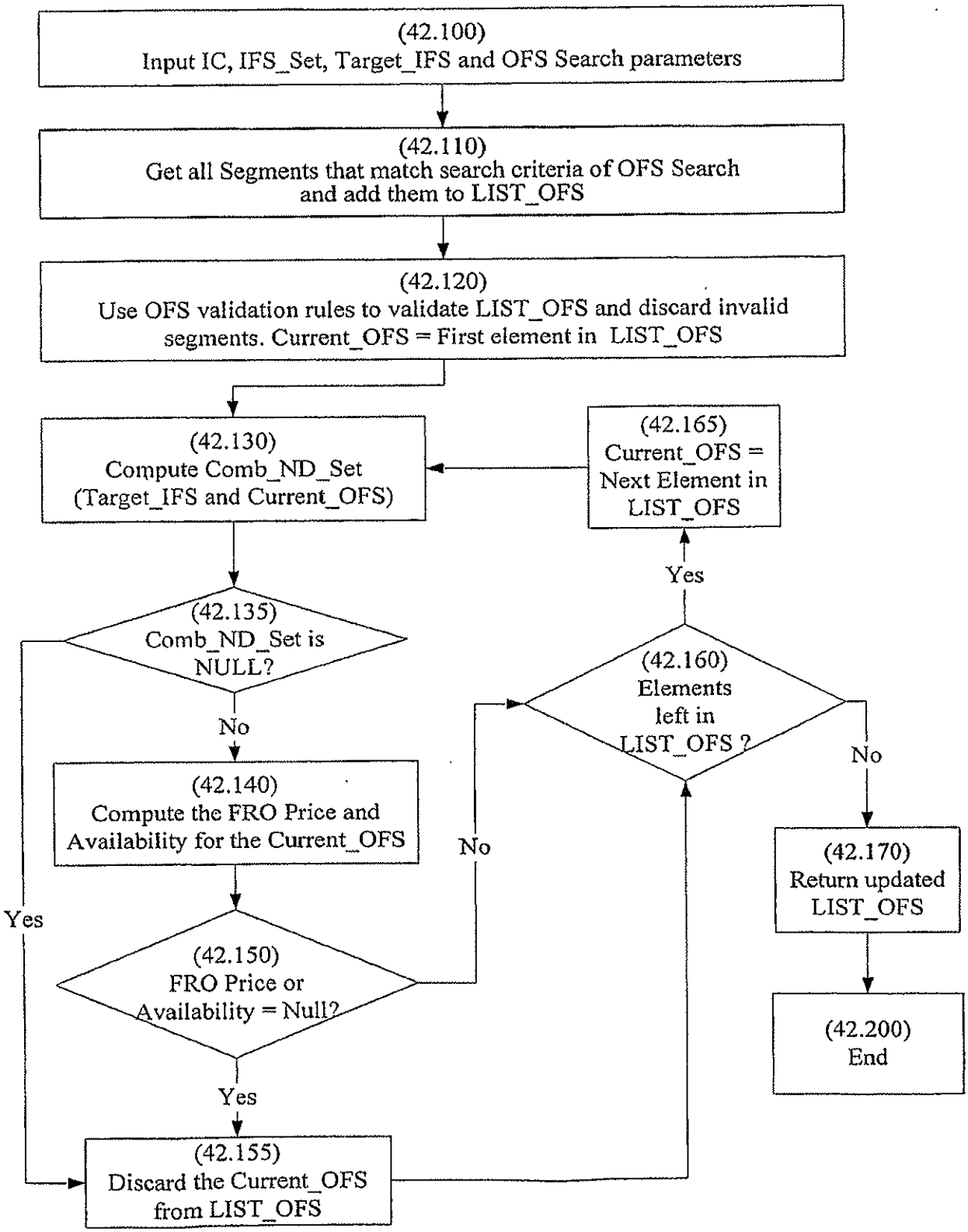

(42.100)
Input IC, IFS_Set, Target_IFS and OFS Search parameters
(42.110)
Get all Segments that match search criteria of OFS Search and add them to LIST_OFS
(42.120)
Use OFS validation rules to validate LIST_OFS and discard invalid segments. Current_OFS = First element in LIST_OFS
(42.130)
Compute Comb_ND_Set (Target_IFS and Current_OFS)
(42.165)
Current_OFS = Next Element in LIST_OFS
(42.135)
Comb_ND_Set is NULL?
Yes
No
(42.140)
Compute the FRO Price and Availability for the Current_OFS
(42.160)
Elements left in LIST_OFS ?
Yes
No
(42.150)
FRO Price or Availability = Null?
No
Yes
(42.155)
Discard the Current_OFS from LIST_OFS
(42.170)
Return updated LIST_OFS
(42.200)
End

Fig. 43

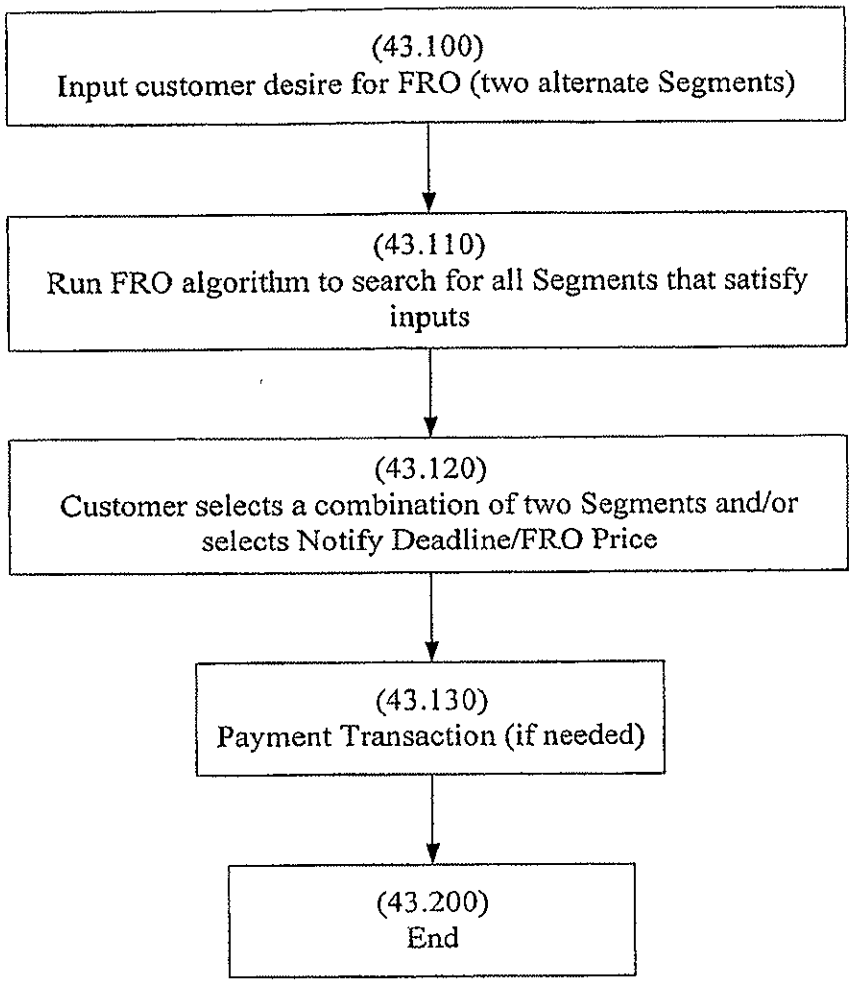

(43.100)
Input customer desire for FRO (two alternate Segments)
(43.110)
Run FRO algorithm to search for all Segments that satisfy inputs
(43.120)
Customer selects a combination of two Segments and/or selects Notify Deadline/FRO Price
(43.130)
Payment Transaction (if needed)
(43.200)
End

Fig. 44

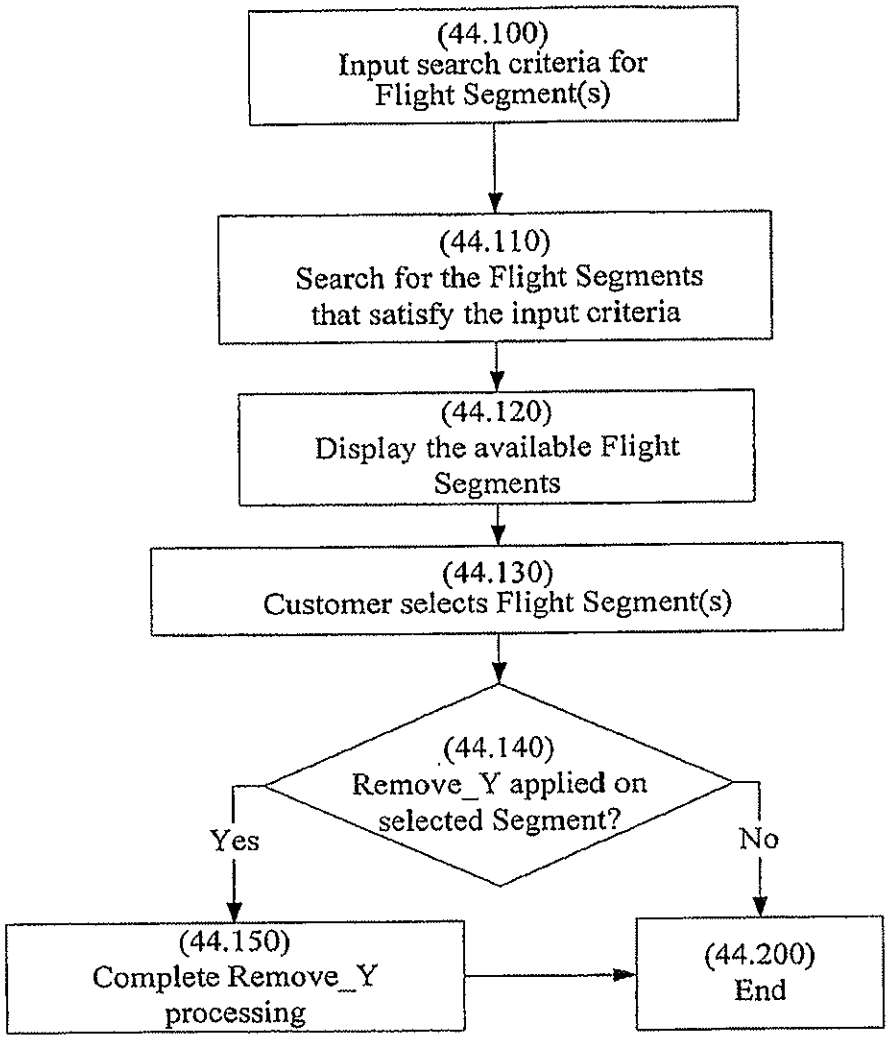

(44.100)
Input search criteria for Flight Segment(s)
(44.110)
Search for the Flight Segments that satisfy the input criteria
(44.120)
Display the available Flight Segments
(44.130)
Customer selects Flight Segment(s)
(44.140)
Remove_Y applied on selected Segment?
Yes
No
(44.150)
Complete Remove_Y processing
(44.200)
End

Fig. 45

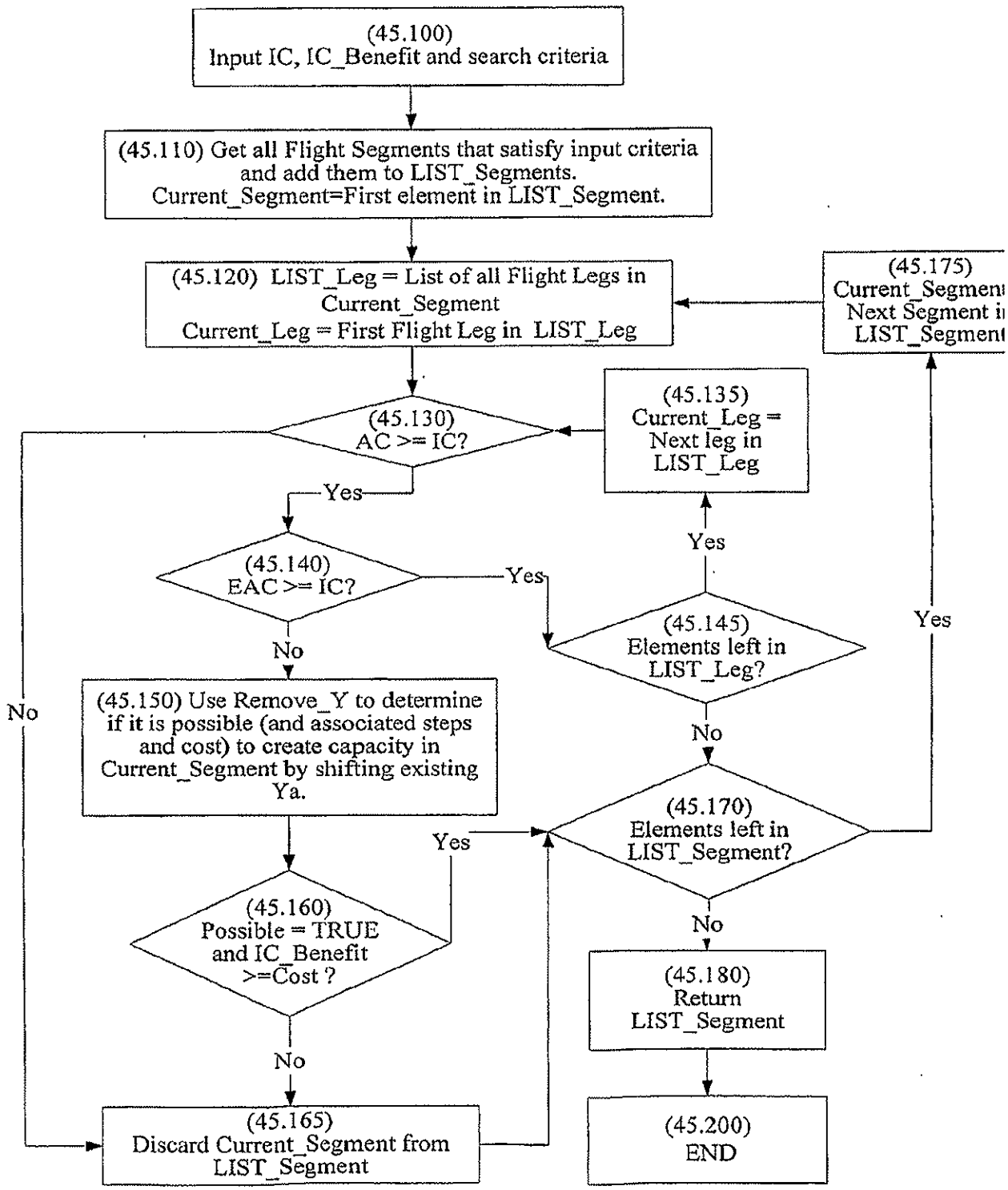

(45.100)
Input IC, IC_Benefit and search criteria
(45.110) Get all Flight Segments that satisfy input criteria and add them to LIST_Segments.
Current_Segment=First element in LIST_Segment.
(45.120) LIST_Leg = List of all Flight Legs in Current_Segment
Current_Leg = First Flight Leg in LIST_Leg
(45.175)
Current_Segmen
Next Segment i
LIST_Segmen
(45.135)
Current_Leg =
Next leg in
LIST_Leg
(45.130)
AC >= IC?
Yes
No
(45.140)
EAC >= IC?
Yes
No
(45.145)
Elements left in
LIST_Leg?
Yes
No
(45.150) Use Remove_Y to determine if it is possible (and associated steps and cost) to create capacity in Current_Segment by shifting existing Ya.
(45.170)
Elements left in
LIST_Segment?
Yes
No
(45.160)
Possible = TRUE
and IC_Benefit
>=Cost ?
Yes
No
(45.165)
Discard Current_Segment from
LIST_Segment
(45.180)
Return
LIST_Segment
(45.200)
END

Fig. 46

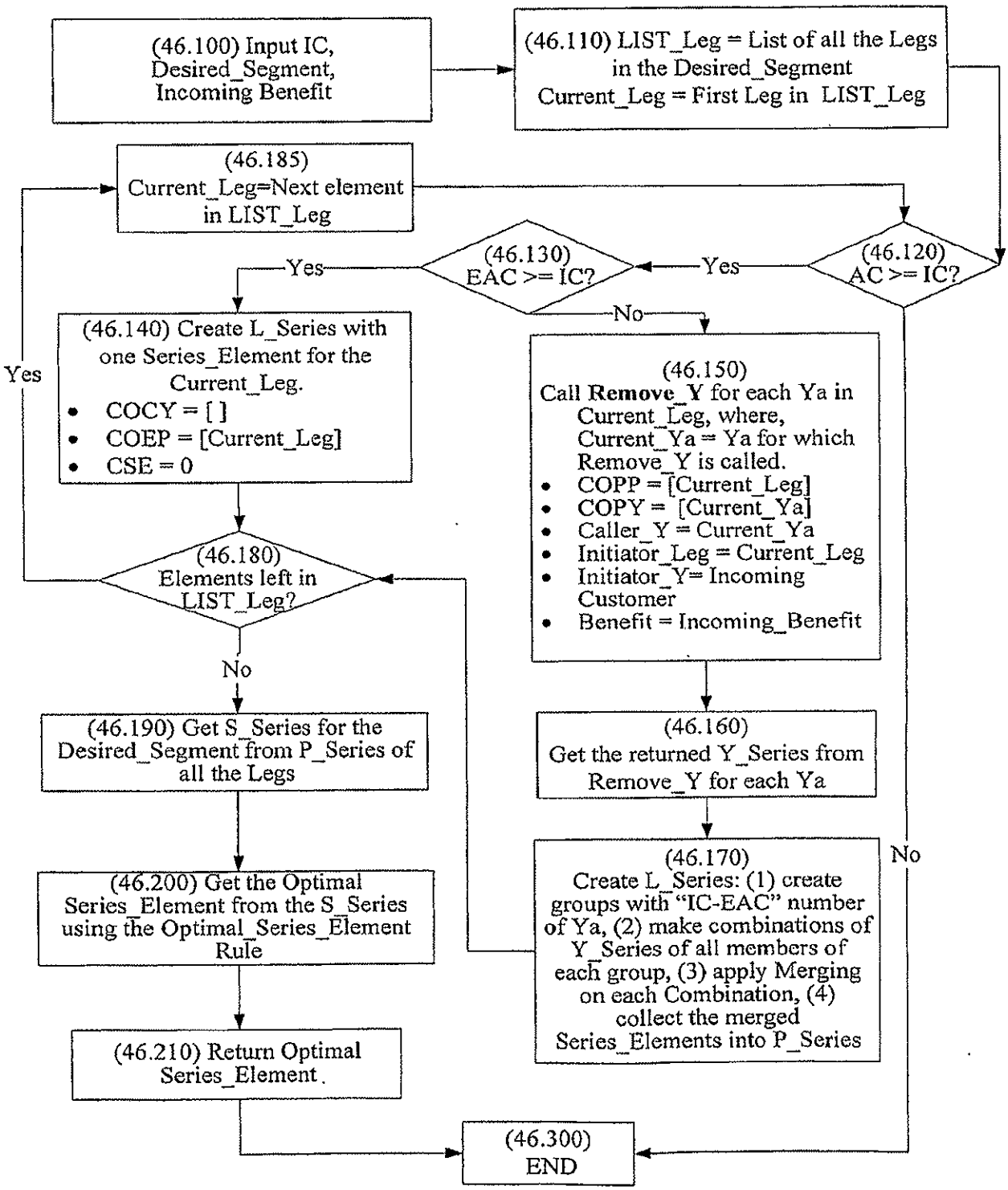

(46.100) Input IC, Desired_Segment, Incoming Benefit
(46.110) LIST_Leg = List of all the Legs in the Desired_Segment Current_Leg = First Leg in LIST_Leg
(46.185) Current_Leg=Next element in LIST_Leg
(46.120) AC >= IC?
Yes
(46.130) EAC >= IC?
Yes
No
Yes
(46.140) Create L_Series with one Series_Element for the Current_Leg.
COCY = [ ]
COEP = [Current_Leg]
CSE = 0
(46.150) Call Remove_Y for each Ya in Current_Leg, where, Current_Ya = Ya for which Remove_Y is called.
COPP = [Current_Leg]
COPY = [Current_Ya]
Caller_Y = Current_Ya
Initiator_Leg = Current_Leg
Initiator_Y= Incoming Customer
Benefit = Incoming_Benefit
(46.180) Elements left in LIST_Leg?
No
(46.190) Get S_Series for the Desired_Segment from P_Series of all the Legs
(46.160) Get the returned Y_Series from Remove_Y for each Ya
(46.170) Create L_Series: (1) create groups with "IC-EAC" number of Ya, (2) make combinations of Y_Series of all members of each group, (3) apply Merging on each Combination, (4) collect the merged Series_Elements into P_Series
No
(46.200) Get the Optimal Series_Element from the S_Series using the Optimal_Series_Element Rule
(46.210) Return Optimal Series_Element.
(46.300) END

Fig. 47

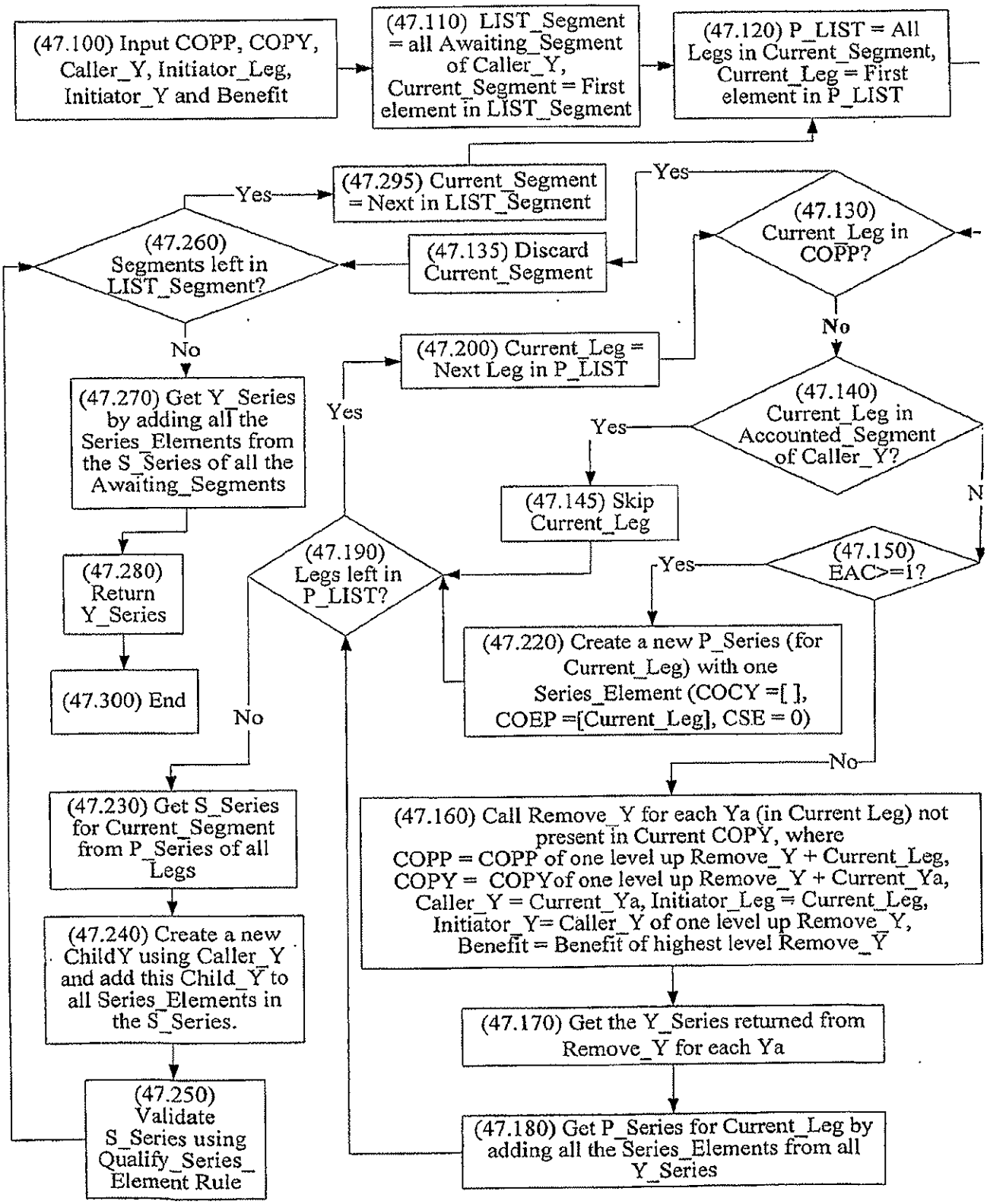

(47.100) Input COPP, COPY, Caller_Y, Initiator_Leg, Initiator_Y and Benefit
(47.110) LIST_Segment = all Awaiting_Segment of Caller_Y, Current_Segment = First element in LIST_Segment
(47.120) P_LIST = All Legs in Current_Segment, Current_Leg = First element in P_LIST
(47.295) Current_Segment = Next in LIST_Segment
Yes
(47.260) Segments left in LIST_Segment?
(47.135) Discard Current_Segment
Yes
(47.130) Current_Leg in COPP?
No
(47.200) Current_Leg = Next Leg in P_LIST
Yes
(47.140) Current_Leg in Accounted_Segment of Caller_Y?
Yes
(47.145) Skip Current_Leg
N
(47.150) EAC>=1?
Yes
(47.190) Legs left in P_LIST?
Yes
(47.220) Create a new P_Series (for Current_Leg) with one Series_Element (COCY =[ ], COEP =[Current_Leg], CSE = 0)
No
(47.270) Get Y_Series by adding all the Series_Elements from the S_Series of all the Awaiting_Segments
(47.280) Return Y_Series
(47.300) End
No
(47.230) Get S_Series for Current_Segment from P_Series of all Legs
(47.240) Create a new ChildY using Caller_Y and add this Child_Y to all Series_Elements in the S_Series.
(47.250) Validate S_Series using Qualify_Series_Element Rule
No
(47.160) Call Remove_Y for each Ya (in Current Leg) not present in Current COPY, where COPP = COPP of one level up Remove_Y + Current_Leg, COPY = COPYof one level up Remove_Y + Current_Ya, Caller_Y = Current_Ya, Initiator_Leg = Current_Leg, Initiator_Y= Caller_Y of one level up Remove_Y, Benefit = Benefit of highest level Remove_Y
(47.170) Get the Y_Series returned from Remove_Y for each Ya
(47.180) Get P_Series for Current_Leg by adding all the Series_Elements from all Y_Series

Fig. 48

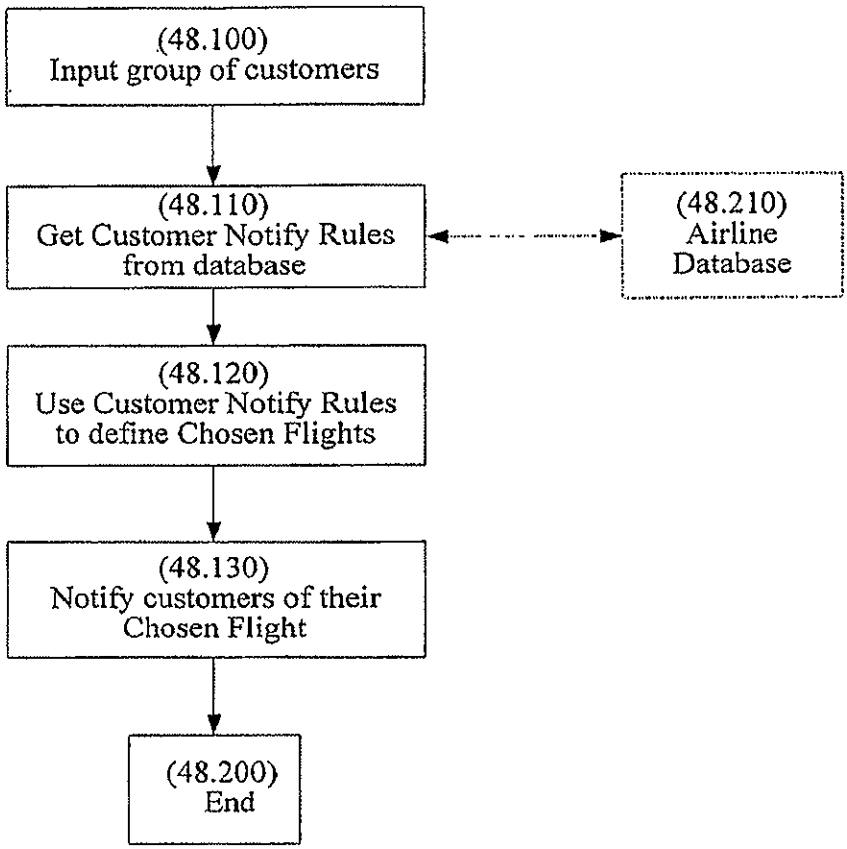

(48.100)
Input group of customers
(48.110)
Get Customer Notify Rules
from database
(48.210)
Airline
Database
(48.120)
Use Customer Notify Rules
to define Chosen Flights
(48.130)
Notify customers of their
Chosen Flight
(48.200)
End

Fig. 49

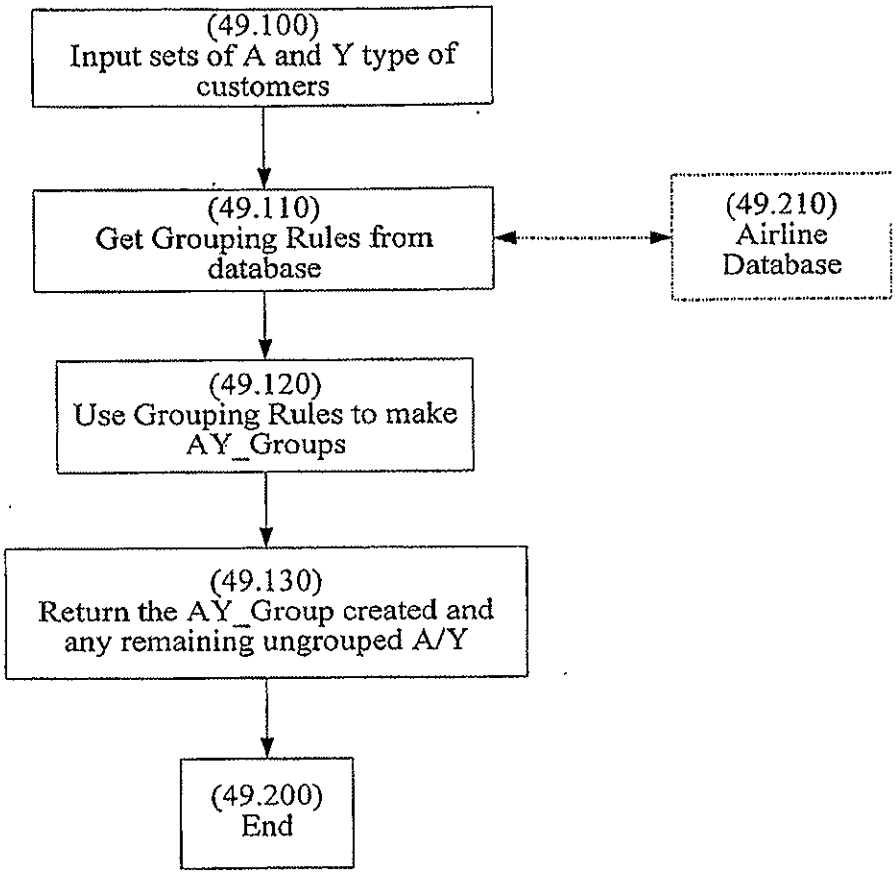

(49.100)
Input sets of A and Y type of customers
(49.110)
Get Grouping Rules from database
(49.210)
Airline Database
(49.120)
Use Grouping Rules to make AY_Groups
(49.130)
Return the AY_Group created and any remaining ungrouped A/Y
(49.200)
End

Fig. 50

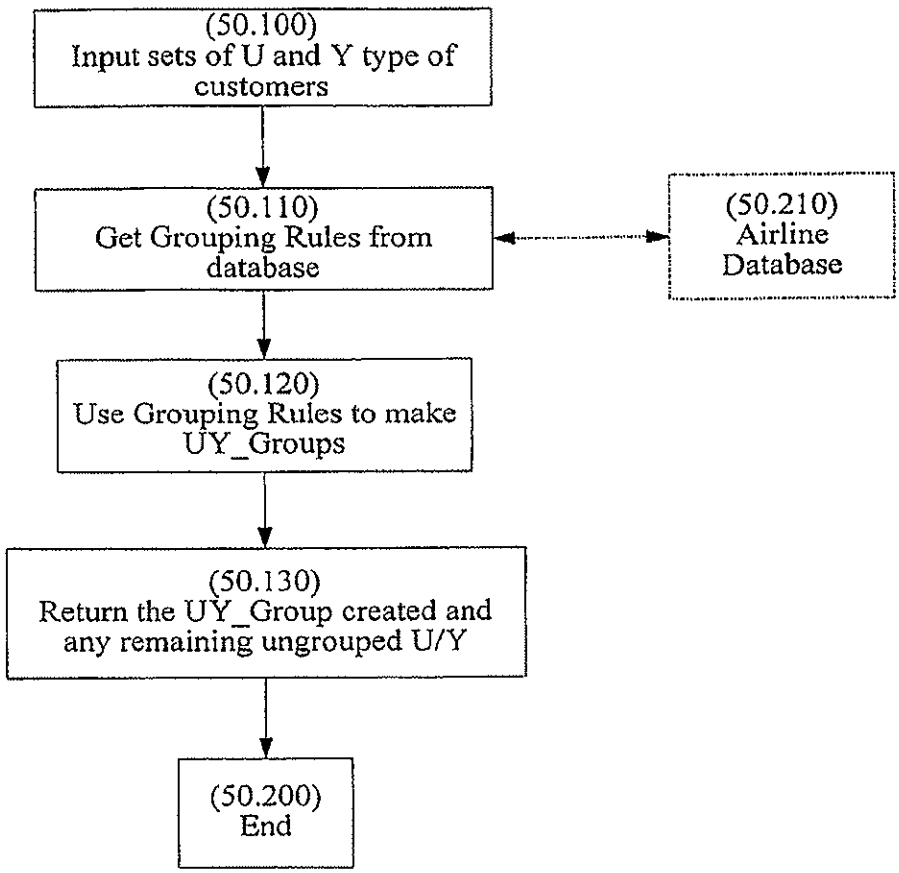

(50.100)
Input sets of U and Y type of customers
(50.110)
Get Grouping Rules from database
(50.210)
Airline Database
(50.120)
Use Grouping Rules to make UY_Groups
(50.130)
Return the UY_Group created and any remaining ungrouped U/Y
(50.200)
End

Fig. 51

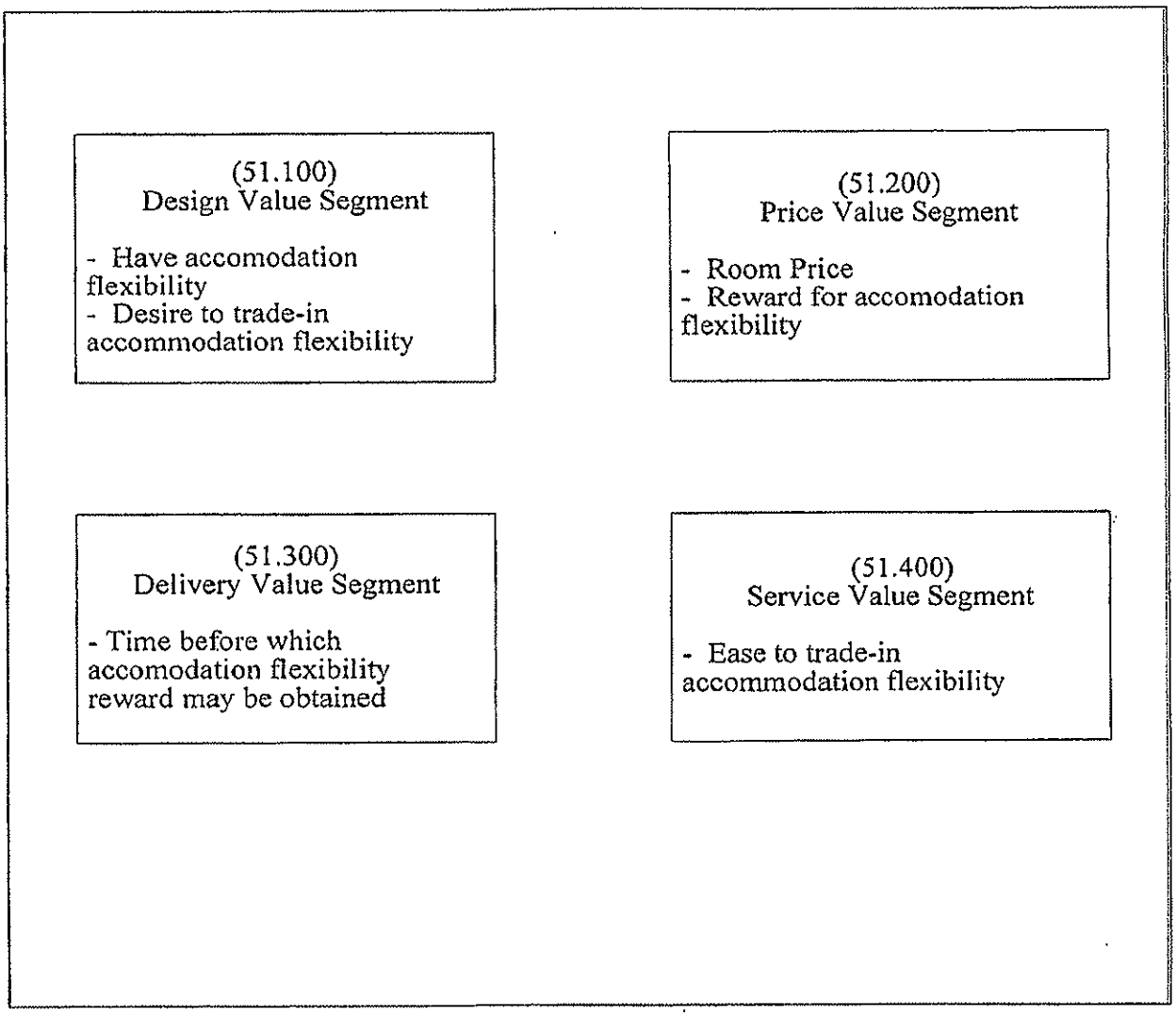

(51.100)
Design Value Segment
- Have accomodation flexibility
- Desire to trade-in accommodation flexibility
(51.200)
Price Value Segment
- Room Price
- Reward for accomodation flexibility
(51.300)
Delivery Value Segment
- Time before which accomodation flexibility reward may be obtained
(51.400)
Service Value Segment
- Ease to trade-in accommodation flexibility

Fig. 52

(52.100)

Crucial Economic Factors

- Fixed and variable costs
- Non-uniform distribution of demand across different rooms
- The difficulty to accurately forecast demand
- The inability or difficulty to shift customers from overloaded Rooms to low-demand Rooms
- Costs and Customer ill-will related to oversale
- Opportunity loss from capacity shortages and spoilage
- The need to develop sustainable competitive advantages
- Customer attrition rate
- Commoditization of the hotel industry

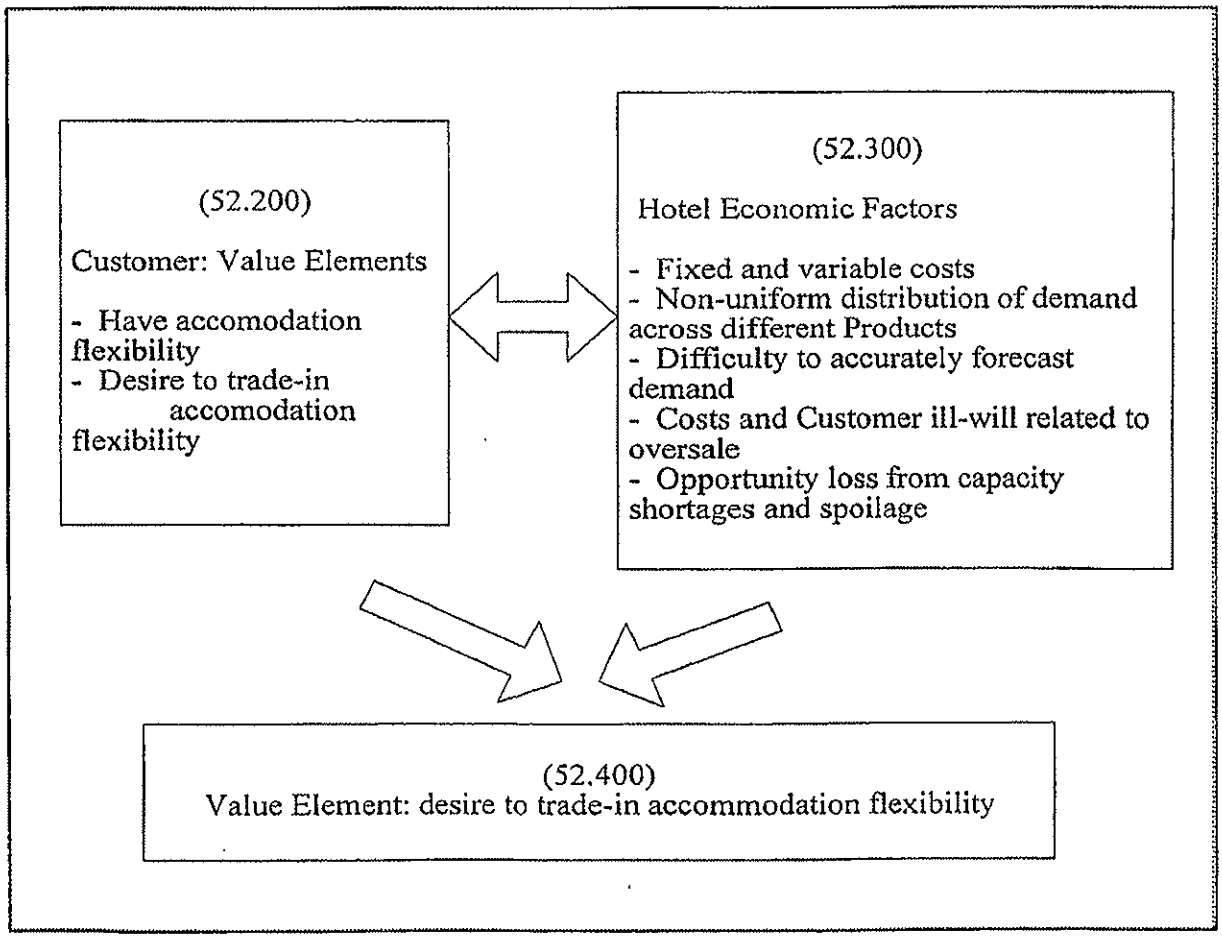


Fig. 53

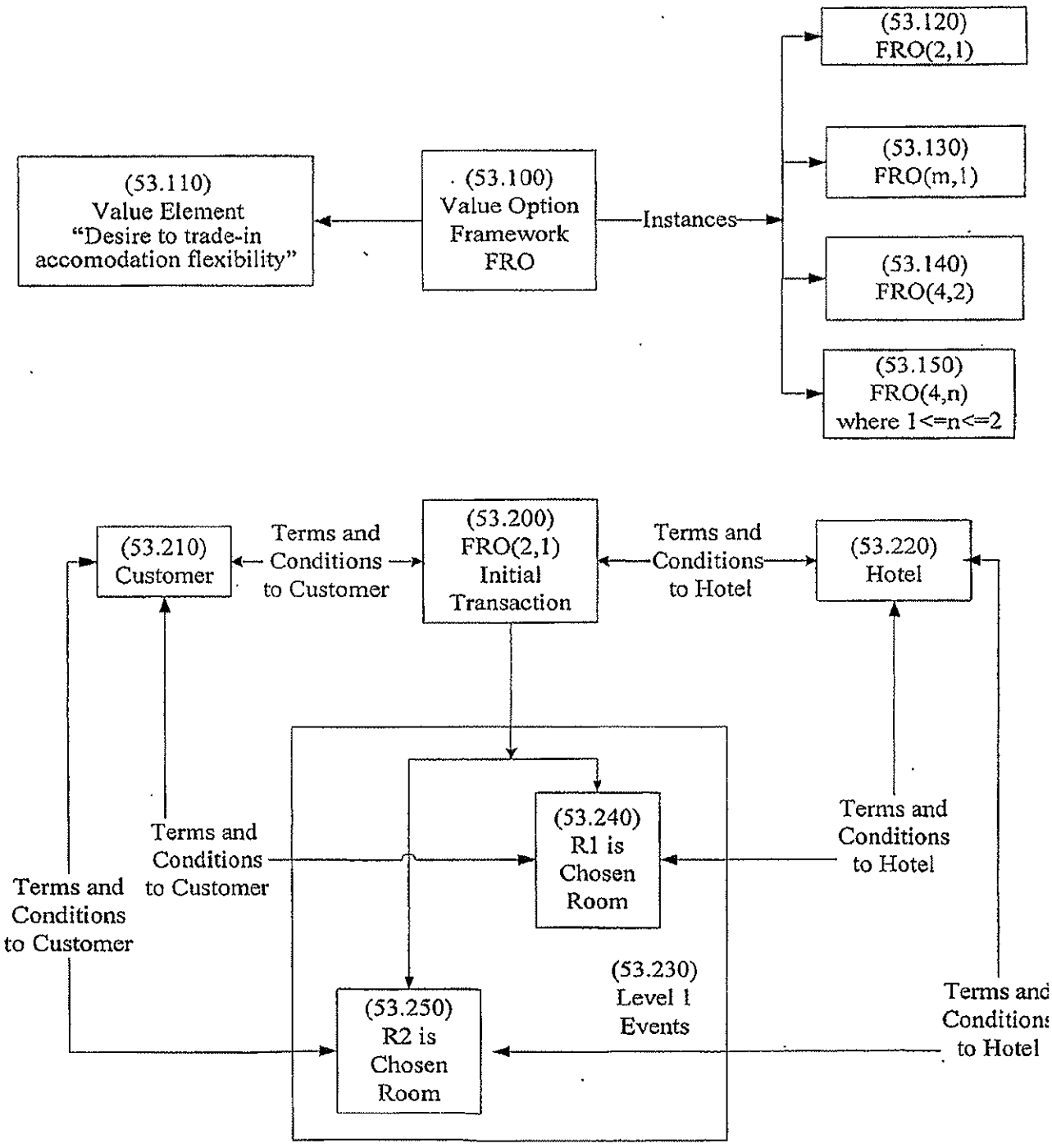

(53.110)
Value Element
"Desire to trade-in
accomodation flexibility"
(53.100)
Value Option
Framework
FRO
Instances
(53.120)
FRO(2,1)
(53.130)
FRO(m,1)
(53.140)
FRO(4,2)
(53.150)
FRO(4,n)
where 1<=n<=2
(53.210)
Customer
Terms and
Conditions
to Customer
(53.200)
FRO(2,1)
Initial
Transaction
Terms and
Conditions
to Hotel
(53.220)
Hotel
(53.240)
R1 is
Chosen
Room
Terms and
Conditions
to Customer
Terms and
Conditions
to Customer
Terms and
Conditions
to Hotel
(53.230)
Level 1
Events
(53.250)
R2 is
Chosen
Room
Terms and
Conditions
to Hotel

Fig. 54

(54.100) Room R1

XYZ International			
T Square		Check-in	Nov 20
New York		Check-out	Nov 23
Room Type	Guest Room		

(54.200) Room R2

XYZ International			
B Square		Check-in	Nov 20
New York		Check-out	Nov 23
Room Type	Guest Room		

(54.300)

	(1)	(2)
(1)	Room Reservation Date Time	7th Nov 5:24 pm
(2)	FRO Initial Transaction Date Time	10th Nov 5:27 pm
(3)	Room Price Paid	$450
(4)	Initial FRO Discount	$10

(54.400)
Default Room R1

			(1)	(2)	(3)	(4)	(5)
(1)	Notify Deadline (DTC of R1)			5	3	2	1
(2)	FRO Exercise discount as a	R1		$15	$20	$30	$40
(3)	function of Chosen Room	R2		$10	$25	$35	$50

Fig. 55

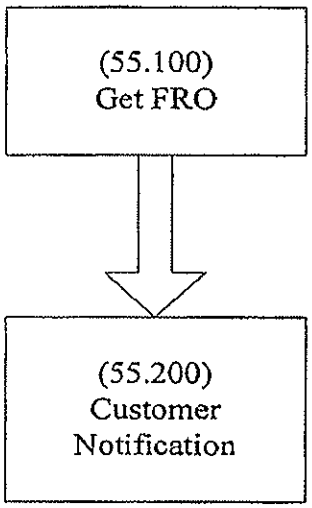


Fig. 56

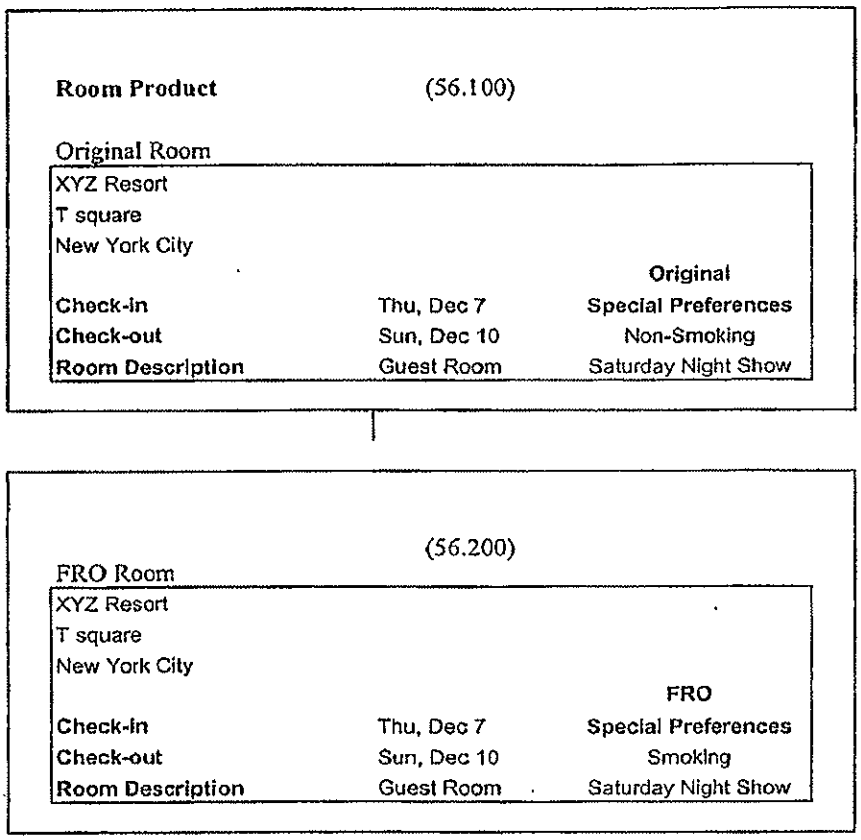

Room Product
(56.100)
Original Room
XYZ Resort
T square
New York City
Original
Check-in
Thu, Dec 7
Special Preferences
Check-out
Sun, Dec 10
Non-Smoking
Room Description
Guest Room
Saturday Night Show
(56.200)
FRO Room
XYZ Resort
T square
New York City
FRO
Check-in
Thu, Dec 7
Special Preferences
Check-out
Sun, Dec 10
Smoking
Room Description
Guest Room
Saturday Night Show

Fig. 57

Room Set (57.100)

Original Room

XYZ Resort		
T square		
New York City		
		Original
Check-in	Thu, Dec 7	**Special Preferences**
Check-out	Sun, Dec 10	Non-Smoking
Room Description	Guest Room	Meeting Room

(57.200)

FRO Room

XYZ Resort		
B Square		
New York City		
		FRO
Check-in	Thu, Dec 7	**Special Preferences**
Check-out	Sun, Dec 10	Smoking
Room Description	Guest Room	Meeting Room

Fig. 58

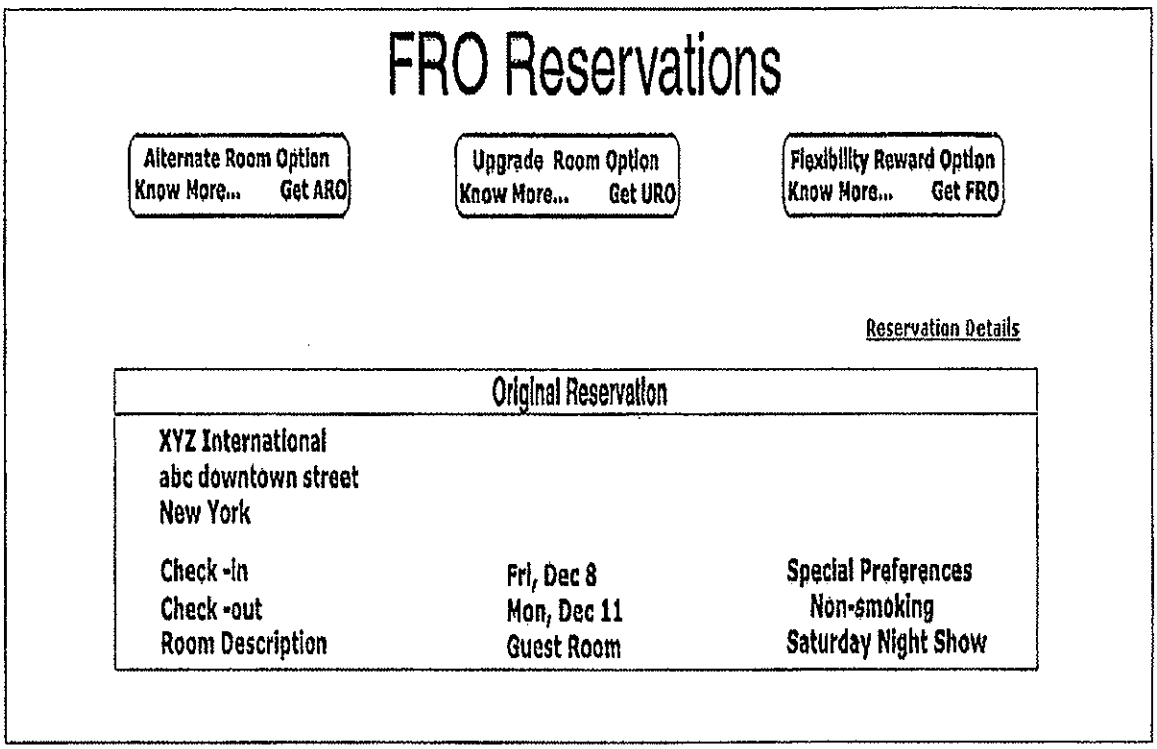

FRO Reservations
Alternate Room Option
Know More... Get ARO
Upgrade Room Option
Know More... Get URO
Flexibility Reward Option
Know More... Get FRO
Reservation Details
Original Reservation
XYZ International
abc downtown street
New York
Check -in
Check -out
Room Description
Fri, Dec 8
Mon, Dec 11
Guest Room
Special Preferences
Non-smoking
Saturday Night Show

Fig. 59

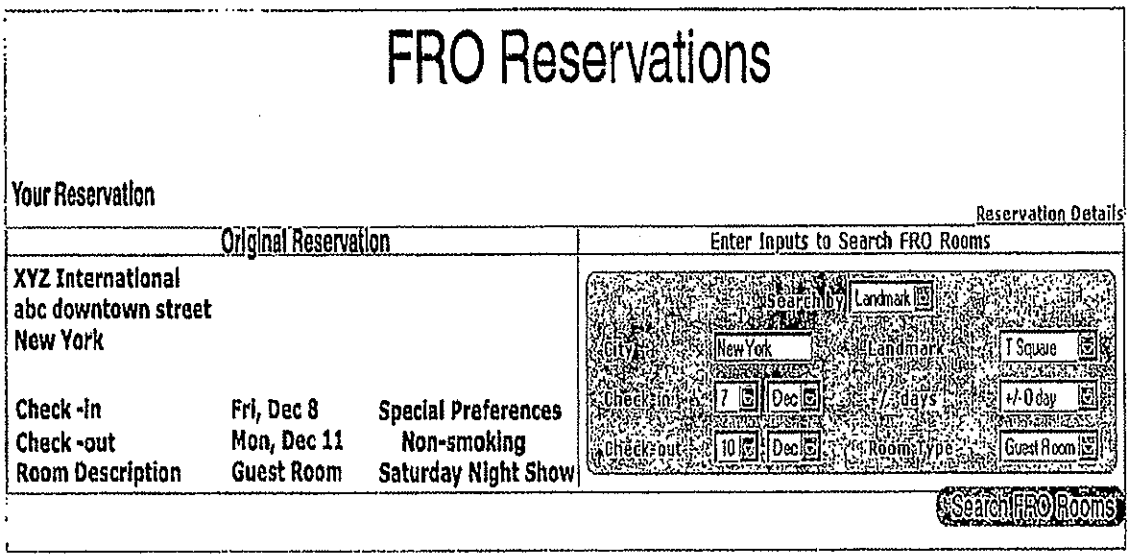

FRO Reservations
Your Reservation
Reservation Details
Original Reservation
Enter Inputs to Search FRO Rooms
XYZ International
abc downtown street
New York
Check -in
Fri, Dec 8
Check -out
Mon, Dec 11
Room Description
Guest Room
Special Preferences
Non-smoking
Saturday Night Show
Search by
Landmark
City
New York
Landmark
T Square
Check in
7
Dec
+/-days
+/-0 day
Check out
10
Dec
Room Type
Guest Room
Search FRO Rooms

Fig. 60

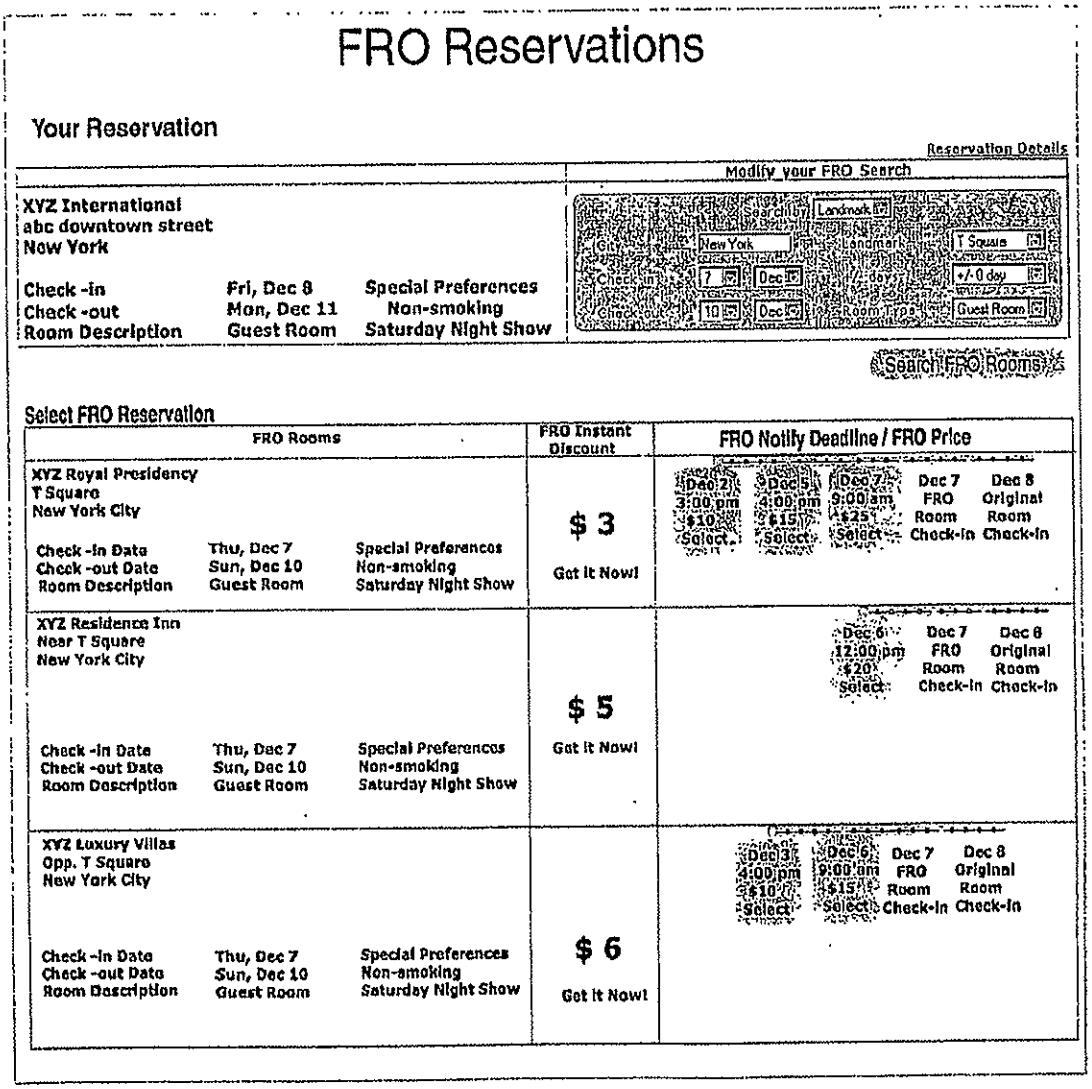

FRO Reservations
Your Reservation
Reservation Details
Modify your FRO Search
XYZ International
abc downtown street
New York
Check -in Fri, Dec 8
Check -out Mon, Dec 11
Room Description Guest Room
Special Preferences
Non-smoking
Saturday Night Show
Search FRO Rooms
Select FRO Reservation
FRO Rooms
FRO Instant Discount
FRO Notify Deadline / FRO Price
XYZ Royal Presidency
T Square
New York City
Check -In Date Thu, Dec 7
Check -out Date Sun, Dec 10
Room Description Guest Room
Special Preferences
Non-smoking
Saturday Night Show
$ 3
Get It Now!
Dec 2 3:00 pm $10 Select
Dec 5 4:00 pm $15 Select
Dec 7 9:00 am $25 Select
Dec 7 FRO Room Check-In
Dec 8 Original Room Check-In
XYZ Residence Inn
Near T Square
New York City
$ 5
Get It Now!
Dec 6 12:00 pm $20 Select
XYZ Luxury Villas
Opp. T Square
New York City
$ 6
Get It Now!
Dec 3 4:00 pm $10 Select
Dec 6 9:00 am $15 Select

Fig. 61

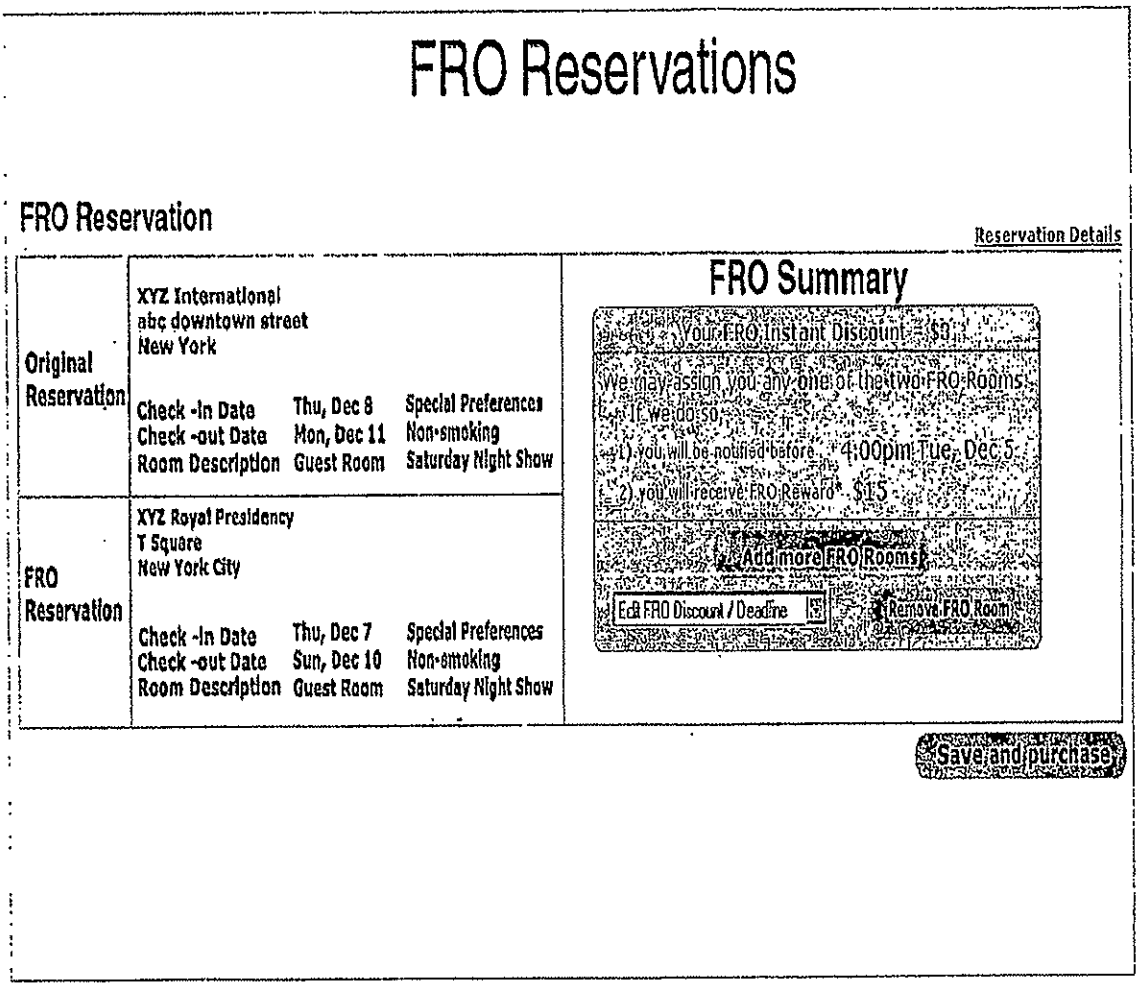

FRO Reservations
FRO Reservation
Reservation Details
Original Reservation
XYZ International
abc downtown street
New York
Check -In Date Thu, Dec 8
Check -out Date Mon, Dec 11
Room Description Guest Room
Special Preferences
Non-smoking
Saturday Night Show
FRO Reservation
XYZ Royal Presidency
T Square
New York City
Check -In Date Thu, Dec 7
Check -out Date Sun, Dec 10
Room Description Guest Room
Special Preferences
Non-smoking
Saturday Night Show
FRO Summary
Your FRO Instant Discount = $0
We may assign you any one of the two FRO Rooms
1) you will be notified before 4:00pm Tue, Dec 5
2) you will receive FRO Reward* $15
Add more FRO Rooms
Edit FRO Discount / Deadline
Remove FRO Room
Save and purchase

Fig. 62

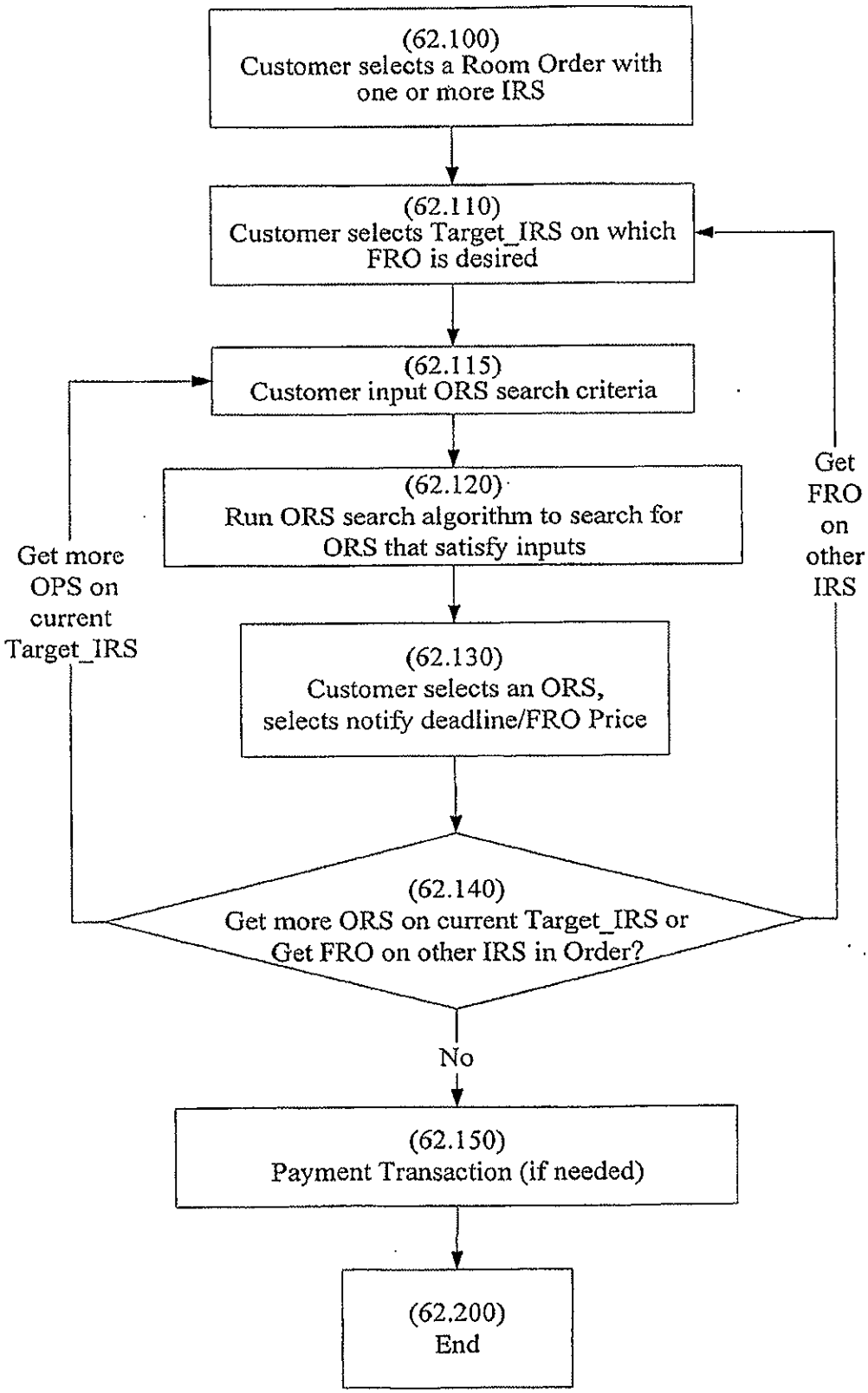

(62.100)
Customer selects a Room Order with one or more IRS
(62.110)
Customer selects Target_IRS on which FRO is desired
(62.115)
Customer input ORS search criteria
(62.120)
Run ORS search algorithm to search for ORS that satisfy inputs
(62.130)
Customer selects an ORS, selects notify deadline/FRO Price
(62.140)
Get more ORS on current Target_IRS or Get FRO on other IRS in Order?
Get more OPS on current Target_IRS
Get FRO on other IRS
No
(62.150)
Payment Transaction (if needed)
(62.200)
End

Fig. 63

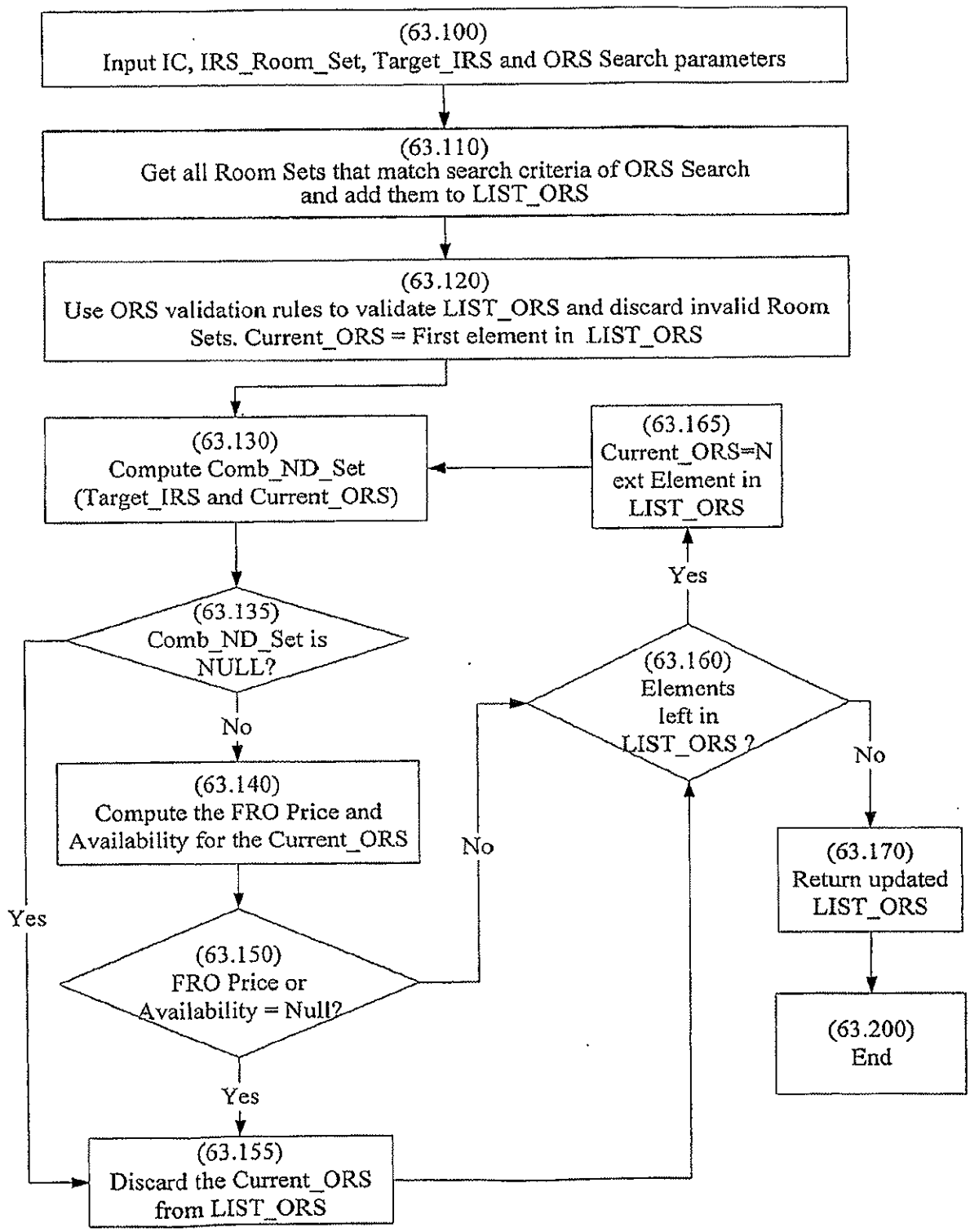

(63.100)
Input IC, IRS_Room_Set, Target_IRS and ORS Search parameters
(63.110)
Get all Room Sets that match search criteria of ORS Search and add them to LIST_ORS
(63.120)
Use ORS validation rules to validate LIST_ORS and discard invalid Room Sets. Current_ORS = First element in LIST_ORS
(63.130)
Compute Comb_ND_Set (Target_IRS and Current_ORS)
(63.165)
Current_ORS=Next Element in LIST_ORS
(63.135)
Comb_ND_Set is NULL?
Yes
No
(63.140)
Compute the FRO Price and Availability for the Current_ORS
(63.150)
FRO Price or Availability = Null?
Yes
No
(63.155)
Discard the Current_ORS from LIST_ORS
(63.160)
Elements left in LIST_ORS ?
Yes
No
(63.170)
Return updated LIST_ORS
(63.200)
End

Fig. 64

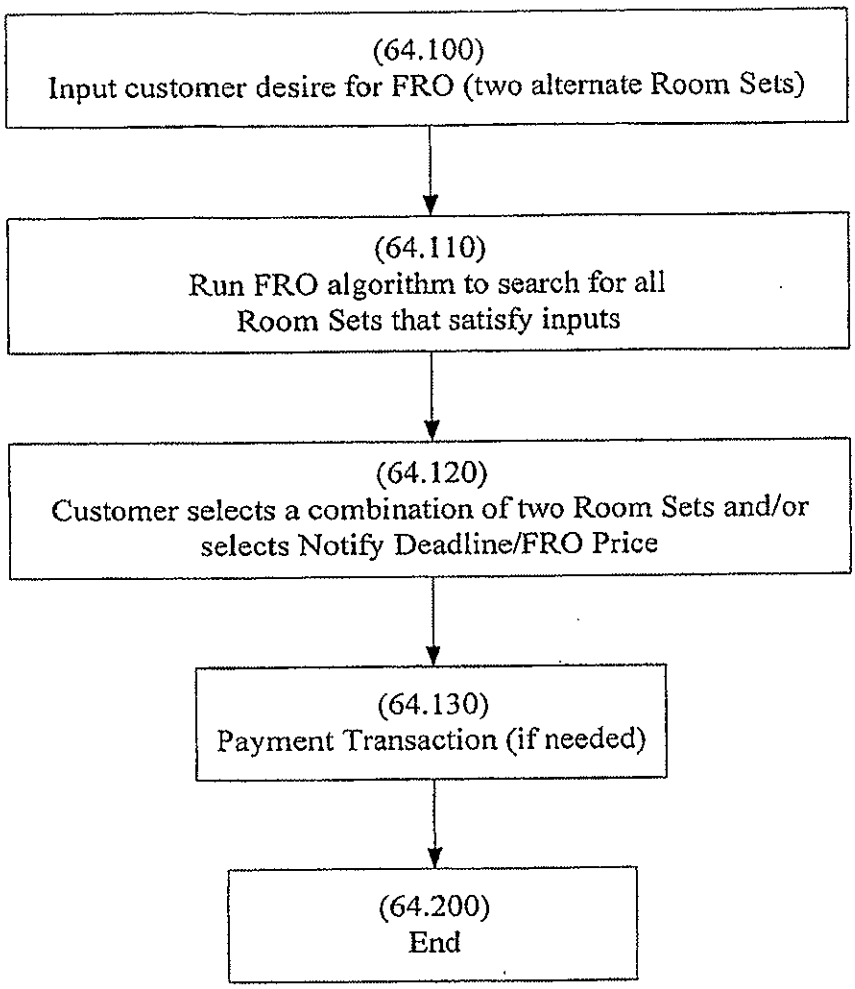

(64.100)
Input customer desire for FRO (two alternate Room Sets)
(64.110)
Run FRO algorithm to search for all
Room Sets that satisfy inputs
(64.120)
Customer selects a combination of two Room Sets and/or
selects Notify Deadline/FRO Price
(64.130)
Payment Transaction (if needed)
(64.200)
End

Fig. 65

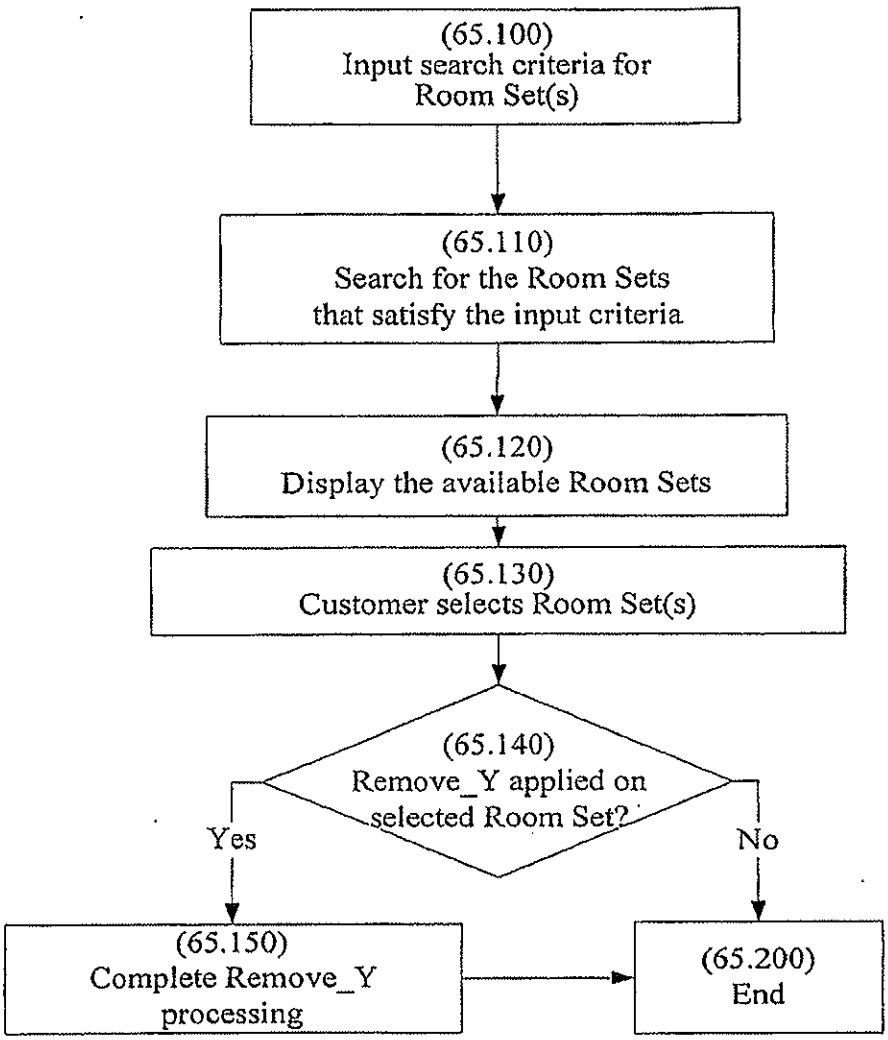

(65.100)
Input search criteria for
Room Set(s)
(65.110)
Search for the Room Sets
that satisfy the input criteria
(65.120)
Display the available Room Sets
(65.130)
Customer selects Room Set(s)
(65.140)
Remove_Y applied on
selected Room Set?
Yes
No
(65.150)
Complete Remove_Y
processing
(65.200)
End

Fig. 66

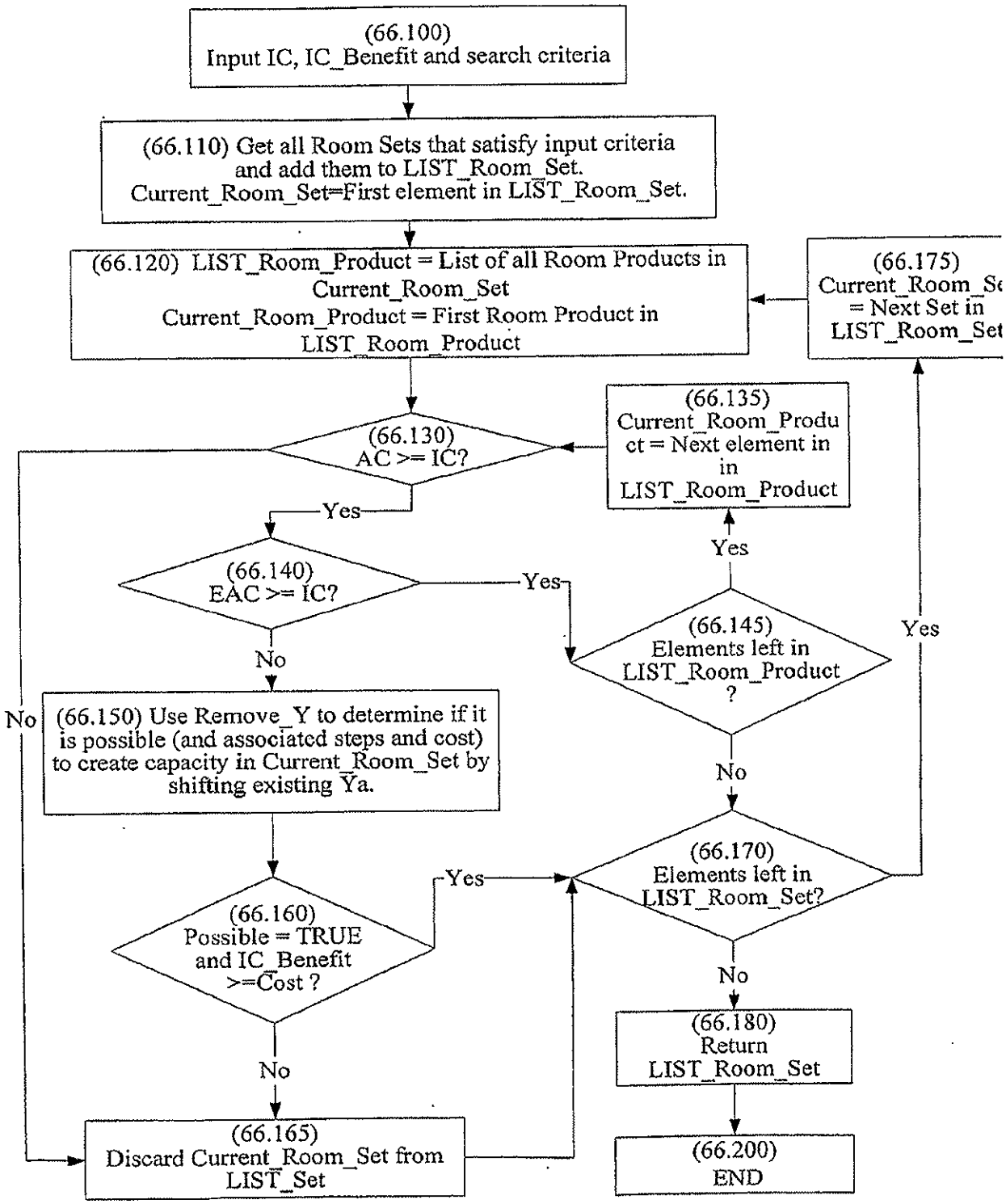

(66.100)
Input IC, IC_Benefit and search criteria
(66.110) Get all Room Sets that satisfy input criteria and add them to LIST_Room_Set.
Current_Room_Set=First element in LIST_Room_Set.
(66.120) LIST_Room_Product = List of all Room Products in Current_Room_Set
Current_Room_Product = First Room Product in LIST_Room_Product
(66.175)
Current_Room_Se
= Next Set in
LIST_Room_Set
(66.130)
AC >= IC?
(66.135)
Current_Room_Produ
ct = Next element in
in
LIST_Room_Product
Yes
Yes
(66.140)
EAC >= IC?
Yes
(66.145)
Elements left in
LIST_Room_Product
?
Yes
No
No
(66.150) Use Remove_Y to determine if it is possible (and associated steps and cost) to create capacity in Current_Room_Set by shifting existing Ya.
(66.170)
Elements left in
LIST_Room_Set?
Yes
(66.160)
Possible = TRUE
and IC_Benefit
>=Cost ?
No
No
(66.180)
Return
LIST_Room_Set
No
(66.165)
Discard Current_Room_Set from
LIST_Set
(66.200)
END

Fig. 67

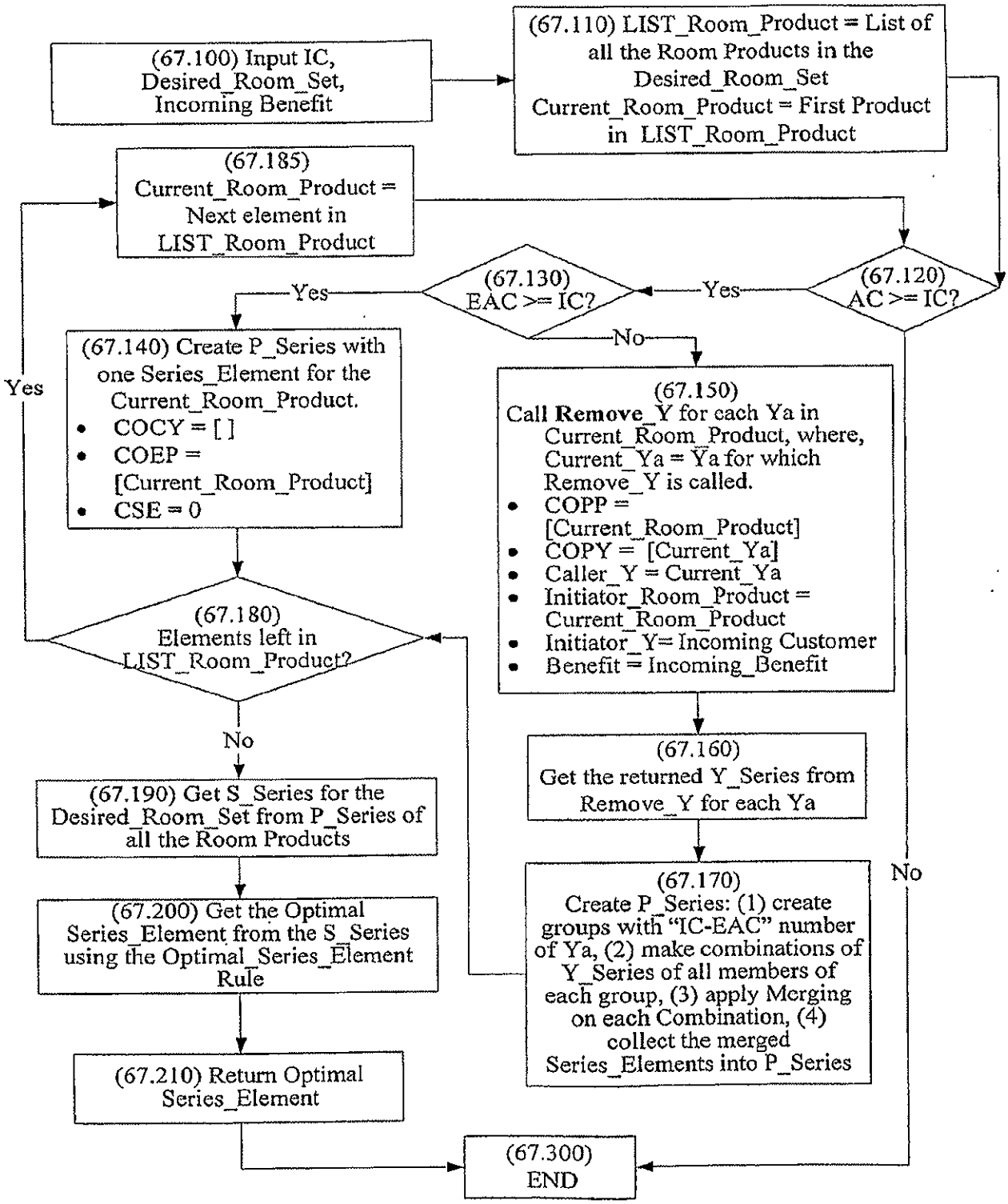

(67.100) Input IC, Desired_Room_Set, Incoming Benefit
(67.110) LIST_Room_Product = List of all the Room Products in the Desired_Room_Set
Current_Room_Product = First Product in LIST_Room_Product
(67.185) Current_Room_Product = Next element in LIST_Room_Product
(67.120) AC >= IC?
Yes
(67.130) EAC >= IC?
Yes
No
(67.140) Create P_Series with one Series_Element for the Current_Room_Product.
• COCY = [ ]
• COEP = [Current_Room_Product]
• CSE = 0
(67.150) Call Remove_Y for each Ya in Current_Room_Product, where, Current_Ya = Ya for which Remove_Y is called.
• COPP = [Current_Room_Product]
• COPY = [Current_Ya]
• Caller_Y = Current_Ya
• Initiator_Room_Product = Current_Room_Product
• Initiator_Y= Incoming Customer
• Benefit = Incoming_Benefit
Yes
(67.180) Elements left in LIST_Room_Product?
No
(67.160) Get the returned Y_Series from Remove_Y for each Ya
(67.170) Create P_Series: (1) create groups with "IC-EAC" number of Ya, (2) make combinations of Y_Series of all members of each group, (3) apply Merging on each Combination, (4) collect the merged Series_Elements into P_Series
No
(67.190) Get S_Series for the Desired_Room_Set from P_Series of all the Room Products
(67.200) Get the Optimal Series_Element from the S_Series using the Optimal_Series_Element Rule
(67.210) Return Optimal Series_Element
(67.300) END

Fig. 68

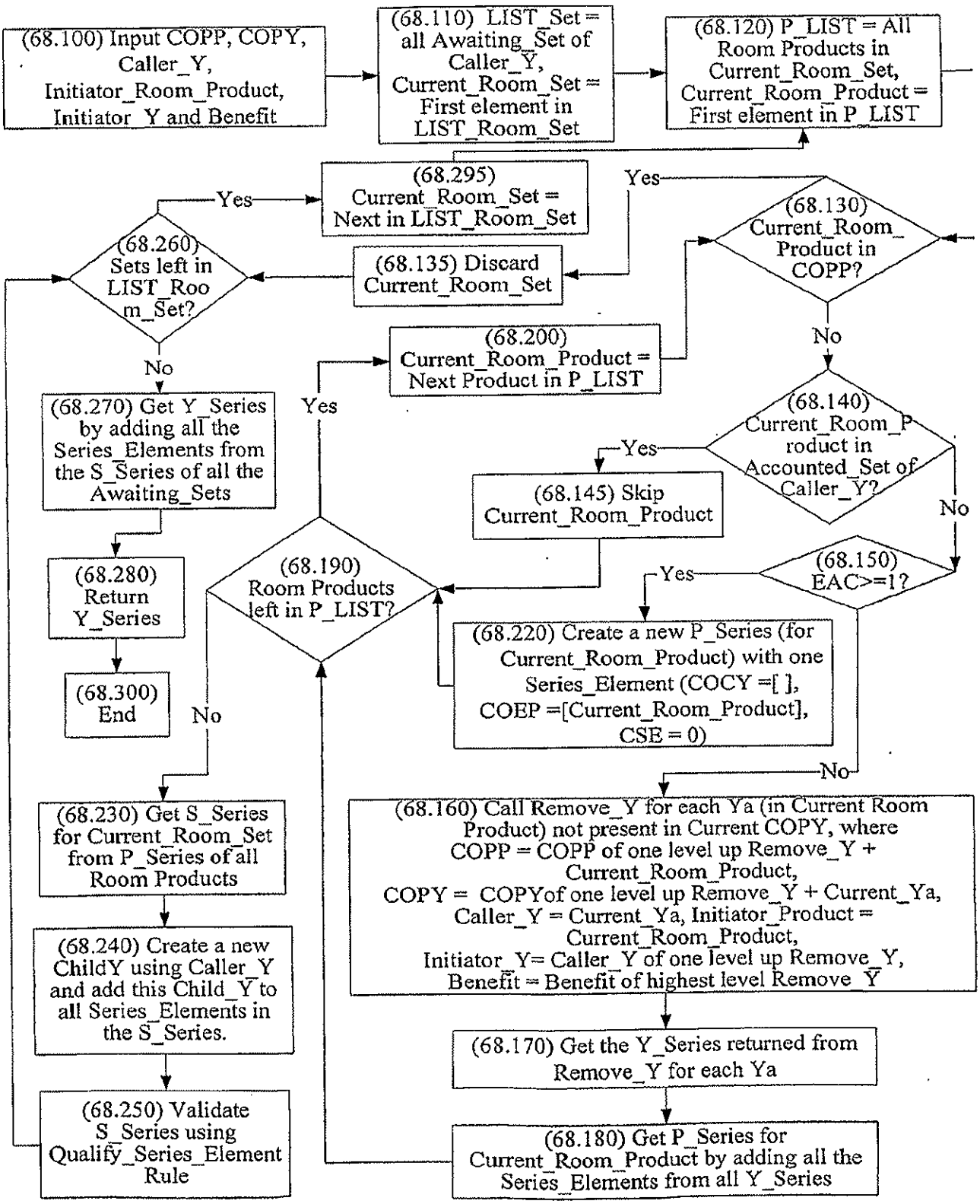

(68.100) Input COPP, COPY, Caller_Y, Initiator_Room_Product, Initiator_Y and Benefit
(68.110) LIST_Set = all Awaiting_Set of Caller_Y, Current_Room_Set = First element in LIST_Room_Set
(68.120) P_LIST = All Room Products in Current_Room_Set, Current_Room_Product = First element in P_LIST
(68.295) Current_Room_Set = Next in LIST_Room_Set
Yes
(68.260) Sets left in LIST_Room_Set?
(68.135) Discard Current_Room_Set
(68.130) Current_Room_Product in COPP?
No
(68.200) Current_Room_Product = Next Product in P_LIST
(68.140) Current_Room_Product in Accounted_Set of Caller_Y?
(68.145) Skip Current_Room_Product
(68.150) EAC>=1?
(68.270) Get Y_Series by adding all the Series_Elements from the S_Series of all the Awaiting_Sets
(68.280) Return Y_Series
(68.300) End
(68.190) Room Products left in P_LIST?
(68.220) Create a new P_Series (for Current_Room_Product) with one Series_Element (COCY =[ ], COEP =[Current_Room_Product], CSE = 0)
(68.230) Get S_Series for Current_Room_Set from P_Series of all Room Products
(68.240) Create a new ChildY using Caller_Y and add this Child_Y to all Series_Elements in the S_Series.
(68.250) Validate S_Series using Qualify_Series_Element Rule
(68.160) Call Remove_Y for each Ya (in Current Room Product) not present in Current COPY, where COPP = COPP of one level up Remove_Y + Current_Room_Product, COPY = COPYof one level up Remove_Y + Current_Ya, Caller_Y = Current_Ya, Initiator_Product = Current_Room_Product, Initiator_Y= Caller_Y of one level up Remove_Y, Benefit = Benefit of highest level Remove_Y
(68.170) Get the Y_Series returned from Remove_Y for each Ya
(68.180) Get P_Series for Current_Room_Product by adding all the Series_Elements from all Y_Series

Fig. 69

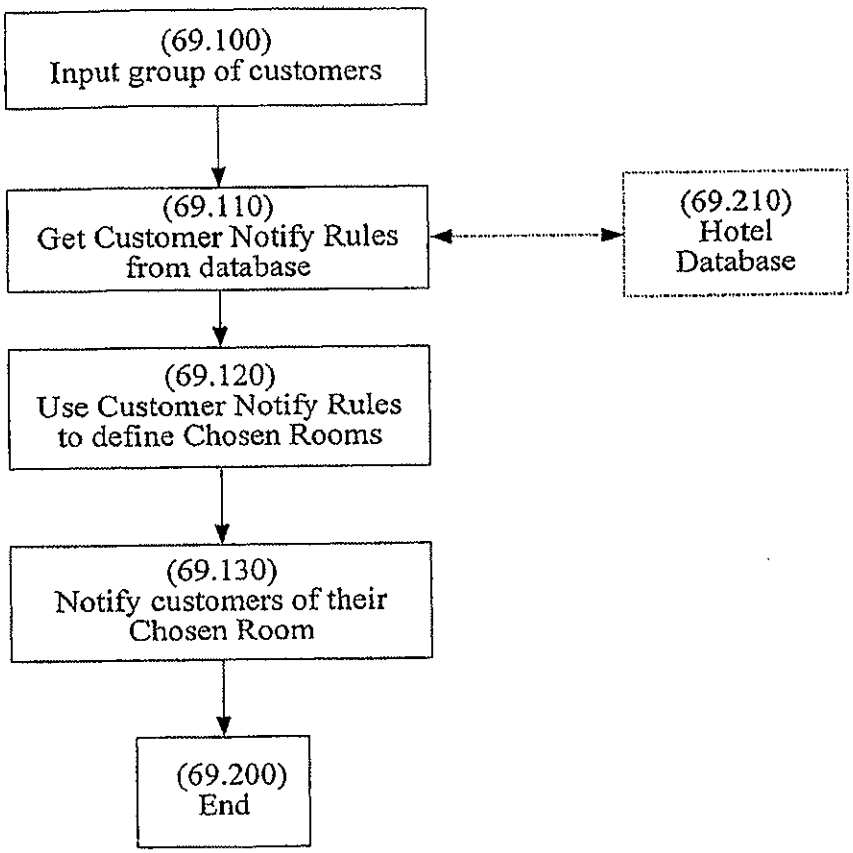

(69.100)
Input group of customers
(69.110)
Get Customer Notify Rules from database
(69.210)
Hotel Database
(69.120)
Use Customer Notify Rules to define Chosen Rooms
(69.130)
Notify customers of their Chosen Room
(69.200)
End

Fig. 70

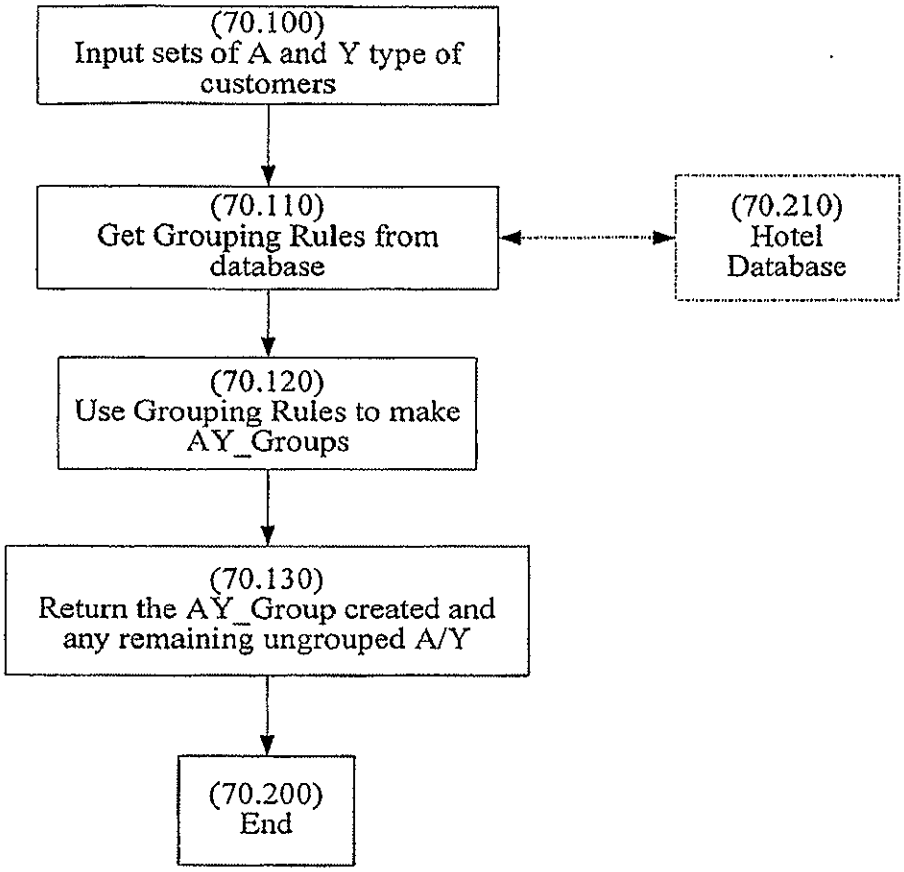
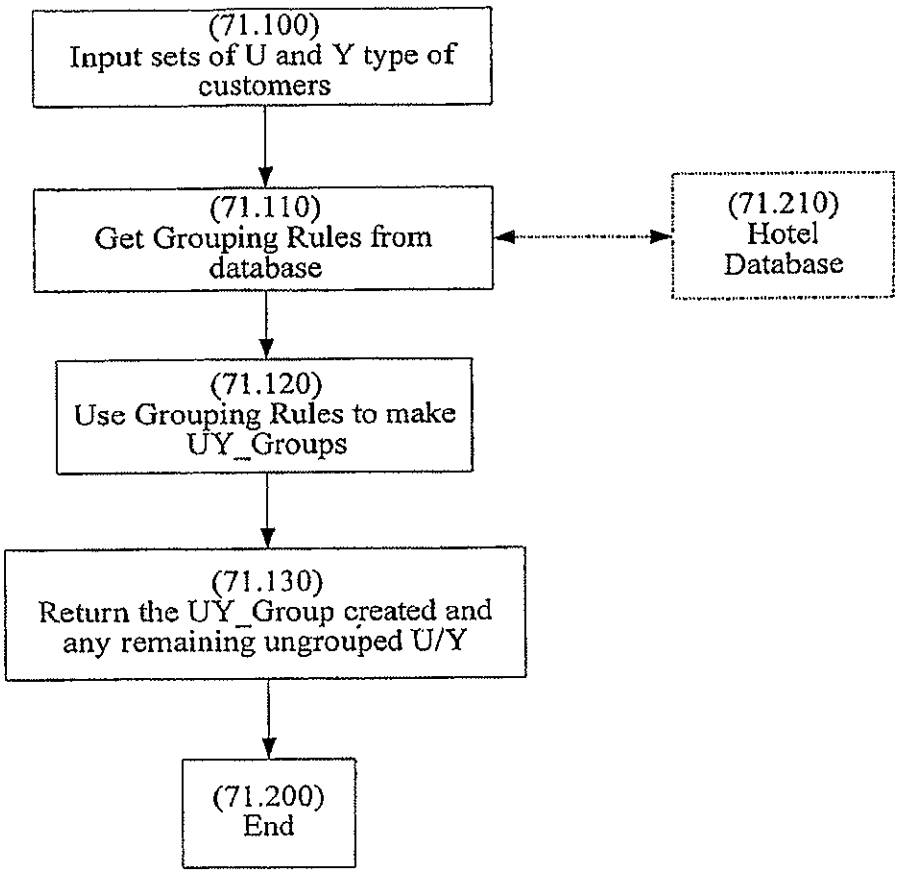

(70.100)
Input sets of A and Y type of customers
(70.110)
Get Grouping Rules from database
(70.210)
Hotel Database
(70.120)
Use Grouping Rules to make AY_Groups
(70.130)
Return the AY_Group created and any remaining ungrouped A/Y
(70.200)
End

Fig. 71


(71.100)
Input sets of U and Y type of customers
(71.110)
Get Grouping Rules from database
(71.210)
Hotel Database
(71.120)
Use Grouping Rules to make UY_Groups
(71.130)
Return the UY_Group created and any remaining ungrouped U/Y
(71.200)
End