MICHEL CALCIU

Associate professor, HDR

LEM IAE - Université des Sciences et Technologies de Lille

IAE - 104 Av. du Peuple Belge, 59043 Lille, France

Tél. : 03 20 12 34 09

e-mail : michel.calciu@univ-lille1.fr

IOAN POPA

Professor

REI- Academia de Studii Economice Bucuresti

Fac.REI - Bld. Dacia nr.41, sector 1, 010404 Bucharest, Romania

Tél. : +4021 319.19.90

e-mail : ioan.popa.rei@gmail.com

A document based marketing decision support and simulation framework. Application to a customer attraction/retention model.

Abstract

The paper suggests a document based framework for building marketing decision support and simulation systems. This framework transfers most of the marketing model implementation work to documents. It replaces many computer language programming tasks by human readable documents leading to increased separation of concerns and acceleration of model access to the market. The framework uses what we call document based programming. It essentially applies recent xml document technologies, like xslt transformations, xml data-binding, xml-schemas to replace programming tasks by standard automated processes. The application of this framework implements a marketing simulation with integrated decision support using a customer attraction/retention model.

Key words: marketing decision support, marketing simulation, document based programming, customer attraction, customer retention

Introduction and objectives

Despite a long tradition in quantitative research in marketing and important accumulations of models, little diffusion of models and their applications can be observed.

As to Little (1970) “the big problem with models is that managers practically never use them”. This situation seems not to have changed a lot ever since. "Even several decades after the earliest operational marketing models were first introduced, their impact on practice remains far below its potential" (Eliashberg & Lilien, 1993, p.19). “Many fewer models are actually used than are developed” (Lilien & Rangaswamy, 2000, p.233). The most frequently mentioned causes for this low adoption of marketing models are low productivity in building and implementing models and bad communication between researchers and mangers.

Although Marketing Decision Support Systems (MDSS) approaches have always been shaped by IT advances as can be seen from the Wierenga and van Bruggen (1997) classification of MDSS, marketing scientists seem to give little importance to information technologies (IT).

Lilien and Rangaswamy (2000) consider that newer IT based data collection and modelling techniques have taken too much time to be adopted placing marketing scientist in great risk to become marginalised as thought leaders in these fields. Tracking customer behaviour over the Internet or recording such behaviour through loyalty cards and real world experiments observing such behaviour are largely computer science dominated. Newer modelling techniques, such as Bayesian networks, neural networks, and data mining have also been actively developed and tested in other areas before being embraced by marketing modellers.

Most traditional marketing models have neglected factors that enhance how the models will be used. "Increasingly, models that do not design in features that take advantage of the distributed and data-rich context provided by the Internet ... will become irrelevant: they will not get used, and will have diminished importance to future developments in the modelling field. To develop models that do get used, modelers must pay attention to the IT-infrastructure under which their models will be used." (Lilien & Rangaswami, 2000, p.232). Among major IT evolutions that have apparently been neglected in marketing decision support literature are object orientation and xml technologies.

Object orientation is a way to analyse and build complex systems and decompose them into logical (classes and objects) and physical (processes module architectures) models with their static and dynamic interactions. As a systems analysis tool it marked an important evolution compared to the traditional procedure oriented approaches. A system can be decomposed into a reduced number of categories or classes using abstraction (highlighting essential properties), encapsulation (hiding detail), modularity and hierarchy. Object orientation is not only integrated in many modern computer programming languages but also in some data base management systems and even in document based approaches like XML (eXtensible Markup Language). XML technologies bring modelling logic much closer to documents. An XML document uses markup to identify content so that information can be easily classified and machine read. Well formed **XML** documents organise markup elements and the information they contain in a tree structure. They follow the Document Object Model (**DOM**) and can be parsed or read into memory to form a hierarchy of objects. Serialisation is the reverse process by which an objects' hierarchy from memory can be transformed and written into an XML document.

The analogy with object oriented programming goes further. As object types are defined by classes, mark-up and the hierarchical structure to be used in an XML document can be predefined in order to produce a valid document. These definitions can be written in XML Scheme language or in the know obsolete Data Type Definition (DTD) language and placed in a distinct document. An illustration will soon be given further in this paper.

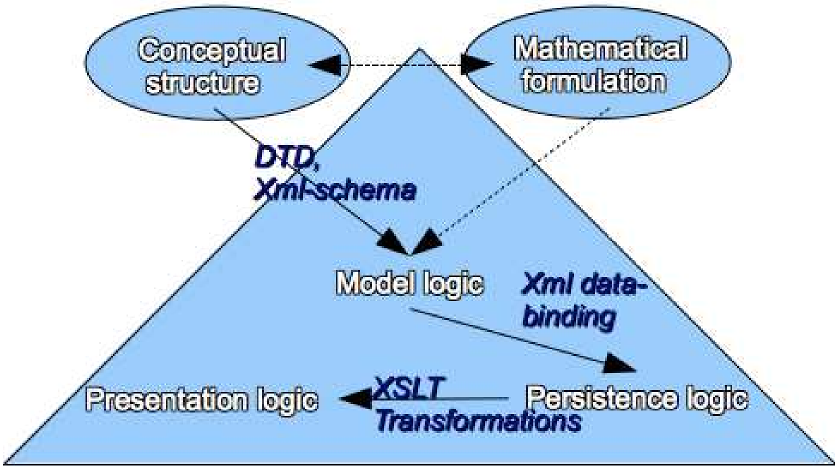
Recent evolutions of the above mentionned approaches and technologies have contributed to improve the quality and accessibility of Marketing Management Support Systems (MMSS) in general and of MDSS in particular.These can now easily be embedded in the organizational IT infrastructure that managers use (Wierenga, 2008).

The bulk of this research concentrates upon building and applying a framework that bases most of the marketing model implementation work on documents. It replaces many computer language programming tasks by human and machine readable documents leading to increased separation of concerns and acceleration of model access to the market. The framework uses what we call document based programming. It essentially applies recent technologies that came with xml documents, like xslt transformations, xml data-binding, xml-schemas to replace programming tasks by automated processes.

A document based marketing model implementation framework

Document based technologies can intervene in many stages of the development and implementation of a marketing model as can be seen in Figure 1. Each of them replaces programming tasks by standard automated processes.

Figure 1 - Document based framework for marketing models implementation



XML technologies like DTD (Document Type Definition) documents and XML-Schemas are very useful from the beginning when the model is created and the main entities it refers to (like the economy or industry, firms, brands, markets, segments etc.) need to be specified. Their main contribution will be in preparing model logic where all non-document based programming is concentrated. They define the hierarchical object oriented structure of the implementation program and generate bulk instructions. At a lesser degree they can help the mathematical formulation of the model. For the sake of separation of concerns we prefer to keep the mathematical formulation of the model apart and decide later which entities will integrate which calculations.

Although in our application, all the model logic is programmed using an object oriented computer language, here too some or all of the programming could have been replaced by dynamic xml documents (like eXtensible Server Pages) in order to make the development of the application more transparent.

As part of a classical three tired application framework, the model logic is closely linked with the persistence and presentation logic. Persistence logic refers to aspects dealing with persistent storage of information needed while executing the application while presentation logic deals with the user interface and extracts and presents available information. In our framework both persistence logic and presentation logic are entirely document based. While persistence logic uses mainly xml-binding techniques in order to store objects in xml documents, presentation logic uses XSLT (eXtensible Stylesheet Language Transformation) transformations in order to extract all needed information from those xml documents.

Our approach advocates separation of concerns. Modellers can concentrate on the mathematical formulation. Systems analysts and programmers can concentrate on the model logic. Web-designers, motivated students and auxiliary staff can concentrate on presentation logic.

In the rest of the paper we will apply our document based model implementation framework to a stylised and specially adapted strategic marketing simulation model. We begin with the mathematical formulation of the model as it gives a complete image of what the model is intended to do and subsequently develop all document based components of our implementation framework.

The dual attraction/retention Marketing Model

Sales, gains and budgets: Total sales of a brand i are given by summing up the number of customers (*n*) in each positional/relationship segment (jk) weighted with the average value of customers (*sv*) of the given relationship category (*k*), here key and non key customers.

$$s_i = \sum_{j}^{J} \sum_{k}^{K} n_{ijk} \cdot sv_k \qquad (1)$$

The gross marketing contribution (margin) results after deduction of the proportion of variable costs (*pcv*) and the fixed costs (*cf*) from the sales value :

$$gm_i = s_i \cdot (1 - pcv_i) - cf_i \ . \qquad (2)$$

The marketing budget for the next period (*E*) can be computed as a percentage (*pe*) of the net marketing contribution.

$$E_i^t = nm_i^{t-1} \cdot pe \qquad (3)$$

Marketing effort per customer (*e*) keeps relationship marketing mix inputs (efforts) comparable.

$$e_i^t = E_i^t / (ms_i^{t-1} n^{t-1}) \qquad (4)$$

Brand decisions: Each period the marketing budget is divided between offensive marketing mix attraction efforts (*xo*) and deffensive mix or retention efforts, the latter being subdivided in order to obtain retention through satisfaction (*xs*), switching costs (*xc*) and direct marketing communications (*xp*)

Segment Response: Key and non-key customers in all segments differ in response behaviour among each other and with regard to each mix ingredient (Colombo & Morrison, 1989). Standardised mix efforts generate attraction and retention measures (indexes) using mix element specific customer response functions based upon a decision calculus approach (Little, 1970,1975) .

Defensive mix effects: The retention indexes can also be seen as measures of the propensity of being loyal to a brand. They consist of

retention from satisfaction and quality (Rust, Zahorik & Keiningham, 1995; Rust & Zahorik, 1993).

$$r1_{ik} = fs_k(xs'_i) \text{ où } xs'_i = xs_i / \overline{xs}. \qquad (5)$$

retention by switching costs:

$$r2_{ik} = fc_k(xc'_i) \text{ où } xc'_i = xc_i / \overline{xc}. \qquad (6)$$

retention by direct marketing communications:

$$r3_{ik} = fp_k(xp'_i) \text{ où } xp'_i = xp_i / \overline{xp}. \qquad (7)$$

The brand loyalty rate (*l*) of a segment or the share of it's « hard core loyal » customers results from the combination (here multiplicative) of these defensive mix effects. The formula takes into account the proportion of customers leaving the market (q) and uses reference loyalty L^{o}.

$$l_{ijk} = (1-q) l_0 \prod_{h=1}^{3} r_{ijk}^{\beta l} \qquad (8)$$

Offensive mix effects: Besides standardised attraction efforts (budget), offensive marketing mix decision include also brand positioning objectives trying to target segments and there « ideal points ». Therefor attraction (A_{ij}) of a brand i on a segment j is a function of what we call intrinsic attractiveness of that brand (A_i) or its mass in physical terms and the distance (d_{ij}) between perceived positioning of the brand and the ideal point of segment j that represents expectancies of customers composing it.

$$A_{ijk} = f_{ok}(x'_{oi} / d'_{ij}) \text{ où } x'_{oi} = x_{oi} / \bar{x}_{o.} \quad d'_{ij} = d_{ij} / \bar{d}_{.j} \qquad (9)$$

As for the other mix element, key and non-key segments (value or relationship segments, k) have different responsiveness to attraction.

A brand's positioning (p) at time t depends on « offensive » advertising efforts and on the percentage of advertising budget affected to research which impacts (re)positioning precision and advertising effectiveness[1]. It also depends on the planned positioning (p'), on the actual positioning (p^{t-1}) and on the original (natural) positioning of that brand (p^{0}).

$$p_i^t = f(xo_i, xpr_i, p'^{t}_i, p_i^{t-1}, p_i^0) \qquad (10)$$

The distance (d_{ij}) between the perceived position of a brand i and the ideal point of segment j representing that segment's positional expectancies is given in a Q dimensional space (here Q=2) by the following formula :

$$d_{ij} = \sum_{q=1}^{Q} \sqrt{(p_{iq} - p_{jq})^2} \qquad (11)$$

The acquisition probability (a) of a brand affects remaining not hard core loyal customers, these include new customers entering the market and versatile (not loyal) customers from all brands. It is calculated using attractiveness indexes off all competing brands in the market.

$$a_{ijk} = A_{ijk} / \sum_{i=1}^{I} A_{ijk} \qquad (12)$$

Market Transitions: Finally customers transitions from a brand to another in each positional and relationship sub-segment depend on the interplay of two forces attraction and retention,

[1] The positioning sub-model is inspired from the Markstrat model (Larréché and Gatignon, 1990)

that are reflected by the acquisition probability and the loyalty rate.

These transitions can be synthetically represented using the following transition matrix

$$P=\begin{bmatrix} 0 & \boldsymbol{a}' \\ q-dg & [1-(q-dg)][\boldsymbol{L}+(1-\boldsymbol{l})\boldsymbol{a}'] \end{bmatrix} \qquad (13)$$

Its first state is the one of a non-customer. Therefore the first row indicates brands attraction probabilities (a) of new customers entering the market and the first column represents the constant proportion of each brand's customers leaving the market (q-dg), where *q* is the customer renewal rate*,* *g* the growth rate and *d* a dummy value equal to one when the growth rate is negative and zero otherwise. This means that when the growth rate is negative the proportion of customers leaving the market exceeds the renewal rate. The rest of the matrix represents customer transition probabilities between brands using an approach advocated by Bultez (1996,1997). It consists of the diagonal matrix of loyalty rates (L) that is added to the matrix resulting from the multiplication of the complementary versatility rates vector (*1*-l) with the brand acquisition probabilities vector (***a***). This matrix controls customer flows for all brands in each segment (*jk*)

Customer flows: The number of versatile customers or *switchors (ns)* attracted (acquired) by a brand i consists of a part of it's own switchors that have been « re-attracted » and switchors attracted from other brands i'.

$$ns_{ijk}^{t}=(1-q+dg)a_{ijk}\left[(1-L_{ijk})ms_{ijk}^{t-1}+\sum_{i'\neq i}(1-L_{i'jk})ms_{i'jk}^{t-1}\right]n_{jk}^{t-1} \qquad (14)$$

The number of *new customers (nn)* can be calculated using the market's customer renewal rate q (constant over time) and the segment's positive growth rate g_j.

The number of new customers attracted by brand i is

$$nn_{ijk}^{t}=[q+(1-d)g]\,a_{ijk}\,n_{jk}^{t-1} \text{ where } d=1 \text{ if } g<0 \text{ and } d=0 \text{ otherwise} \qquad (15)$$

The number of the brand's i hard core loyal customers is

$$nl_{ijk}^{t}=(1-q)l_{ijk}\,ms_{ijk}^{t-1}\,n_{jk}^{t-1} \qquad (16)$$

The total number of customers of brand i consists of the number of hard core loyals, switchors and new customers in segment jk

$$n_{ijk}^{t}=nl_{ijk}^{t}+ns_{ijk}^{t}+nn_{ijk}^{t} \qquad (17)$$

Market share: The numeric market share of brand i in segment jk is then:

$$ms_{ijk}=n_{ijk}/n_{jk} \qquad (18)$$

and the numeric market share of brand i over all segments is

$$ms_i = \sum_{j=0}^{J}\sum_{k=0}^{K} ms_{ikj}\, n_{jk} / \sum_{j=0}^{J}\sum_{k=0}^{K} n_{jk} \qquad (19)$$

Using the average value (sv_k) of key and non-key customers the value market share can also be computed. This process advances to the next period (t=t+1) and loops until the fixed number of periods is attained.

Prefiguring the conceptual structure of the model with DTD documents and XML-schemas

The mathematical formulation of the model uses the rather general mathematical notations and variable types. In implementation, for various reasons, it useful to adopt mnemonic notations for variables and use a domain specific vocabulary. Technologies aimed to define the vocabulary and the grammar for an xml document are particularly useful at this stage.

Less sophisticated but not xml documents themselves DTD's (Document Type Definition) are rather handy as a starting point . They define entities, their properties and hierarchical structure.

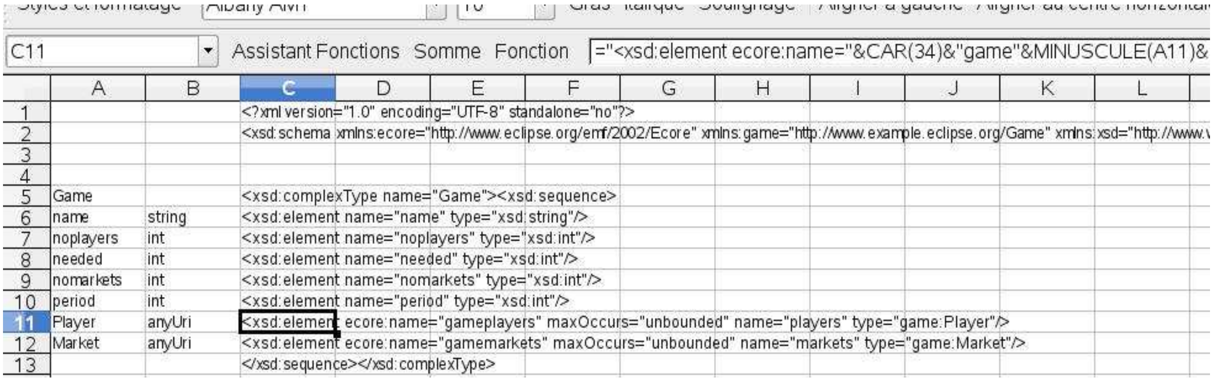
Figure 2 DTD (Document Type Definition)

```
<!ELEMENT Game (name,noplayers,needed,nomarkets,period,Player+,Market+)>
<!ELEMENT Player (name,password,eMail,budget,gain,period,Brand+)>
<!ELEMENT Brand (name,xNat,yNat,period, Decision+,BState+)>
<!ELEMENT Decision
(xViz,yViz,prcRechPub,budgAttract,budgSatisf,budgSwitchc,budgInteract,period)>
<!ELEMENT Bstate (budgTot,x,y,msTot,period,attr1,attr2,attr3,attr4,attr5,attr6,attr7,attr8,
fid1,fid2,fid3,fid4,fid5,fid6,fid7,fid8,ms1,ms2,ms3,ms4,ms5,ms6,ms7,ms8)>
<!ELEMENT Market (Segment+, Response+)>
<!ELEMENT Segment
(xI,yI,xF,yF,valIni,valgr02,valgr35,valgr6,sizeIni,sizegr02,sizegr35,sizegr6, period,BState+)>
<!ELEMENT SState (x,y,val,size,period,nCle,nNoncle,valCle,valNoncle)>
...
```

For example (see figure 2) the simulation (game) besides properties like name or period contains firms (Player) and markets (Market). Firms have brands (Brand) and markets are composed of customer segments (Segments). During the each simulation period firms take decisions (Decision). After each decisions round the simulation advances in the next period by calculating and adding new brand and segment states (Bstate, Sstate).

A more precise definition of the conceptual structure of the future application including the numeric type for variables not available in DTD can be given in an XSD (XML Schema Description) which additionally is itself an xml document. XSD's can be automatically generated form DTD's and numeric types for variable can be added with some external program (like spreadsheets) as in the following example:

Figure 3 - Updating and extending XSD using a spreadsheet



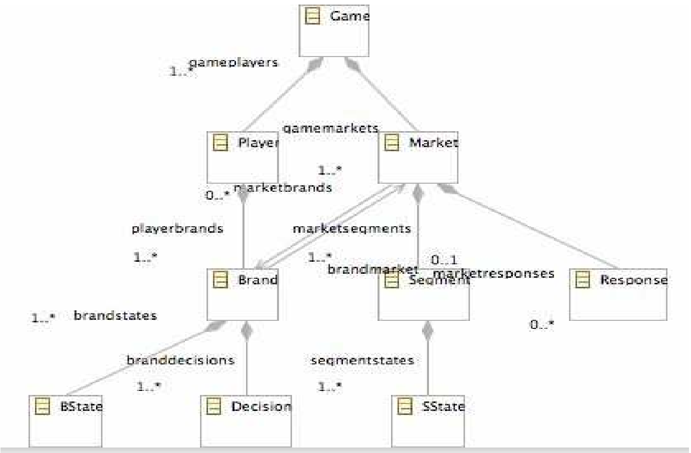
="<xsd:element ecore:name="&CAR(34)&"game"&MINUSCULE(A11)&

	A	B	C
1			<?xml version="1.0" encoding="UTF-8" standalone="no"?>
2			<xsd:schema xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" xmlns:game="http://www.example.eclipse.org/Game" xmlns:xsd="http://www.\
3			
4			
5	Game		<xsd:complexType name="Game"><xsd:sequence>
6	name	string	<xsd:element name="name" type="xsd:string"/>
7	noplayers	int	<xsd:element name="noplayers" type="xsd:int"/>
8	needed	int	<xsd:element name="needed" type="xsd:int"/>
9	nomarkets	int	<xsd:element name="nomarkets" type="xsd:int"/>
10	period	int	<xsd:element name="period" type="xsd:int"/>
11	Player	anyUri	<xsd:element ecore:name="gameplayers" maxOccurs="unbounded" name="players" type="game:Player"/>
12	Market	anyUri	<xsd:element ecore:name="gamemarkets" maxOccurs="unbounded" name="markets" type="game:Market"/>
13			</xsd:sequence></xsd:complexType>

Model logic and applications modelling framework

The XSD document which is an open standard can be used to automatically generate the class diagram of the application using standard notation defined in UML (Unified Modelling Language).

Figure 4 - Class diagram of the simulation in UML notation



The class diagram in figure 4, due to space limitations, is restrained to class names and associations while class properties and methods are hidden. Most explanation given for the DTD hold for the class diagram. Besides containment relations between entities (classes) that had been described in DTD, additional associations between brands and markets have been graphically specified using appropriate tools. While brands belong to players (firms) they are designed for and compete on specific markets. There is a one to many association between a market and brands.

With a combination of UML diagrams, a complete model of an application can be specified. Using appropriate tools that often are regrouped in a *modelling framework2* part of or, in simple cases, all of an application can be generated.

The only remaining programming effort needed to implement the marketing model consists in integrating the mathematical formulations as methods of specific classes. For simplicity reasons most of the mathematical formulations are in the market's “compute” method and in the segment's “advance” method. As can be seen from the DTD document (figure 2) the segment contains all information needed to compute it's state informations (size of demand, number and average value of key and non-key customers etc.) for the next period. As their states are predefined by a simulation scenario, segments (demand) advance independently of firms' decisions. Considering segments' and brands' state information and firms' brand decisions, the market compute method strictly applies the mathematical formulations of the model given above in order to populate brand state informations for the next period. Some visually appealing details will be given in the presentation logic section.

Although in our application, all the model logic is programmed using an object oriented computer language on which the modelling framework relies, here too some or all of the programming could have been replaced by dynamic xml documents (like eXtensible Server Pages) in order to make the development of the application more transparent.

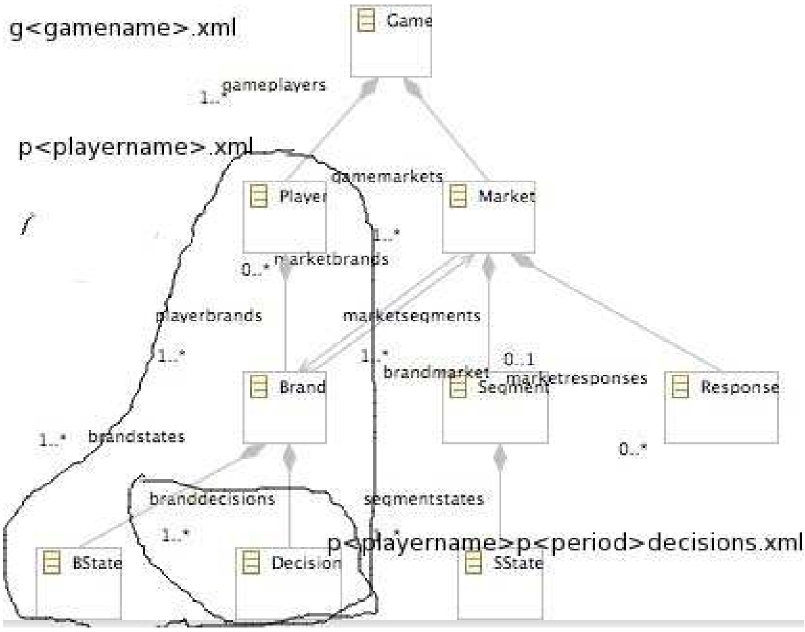
Persistence logic through xml data binding at various levels of the object hierarchy

As part of a three tired application the model logic is closely linked with the persistence and presentation logic. Persistence logic refers to aspects dealing with persistent storage of information needed while executing the application. Persistence logic can use as storage support documents or databases. Here we privilege document based persistence through XML data binding. XML data binding refers to the process of representing the information in an XML document as an object in computer memory and vice-versa. An XML data binder accomplishes this by automatically creating a mapping between elements of the XML schema of the document we wish to bind and members of a class to be represented in memory. In this way the whole complexity of using files to input and output data when building applications becomes transparent. Transferring memory objects to an XML document is called

[2] We used the Eclipse Modelling Framework.

marshalling, and the reverse action is called unmarshalling[3].

Figure 5 Three levels of xml data binding

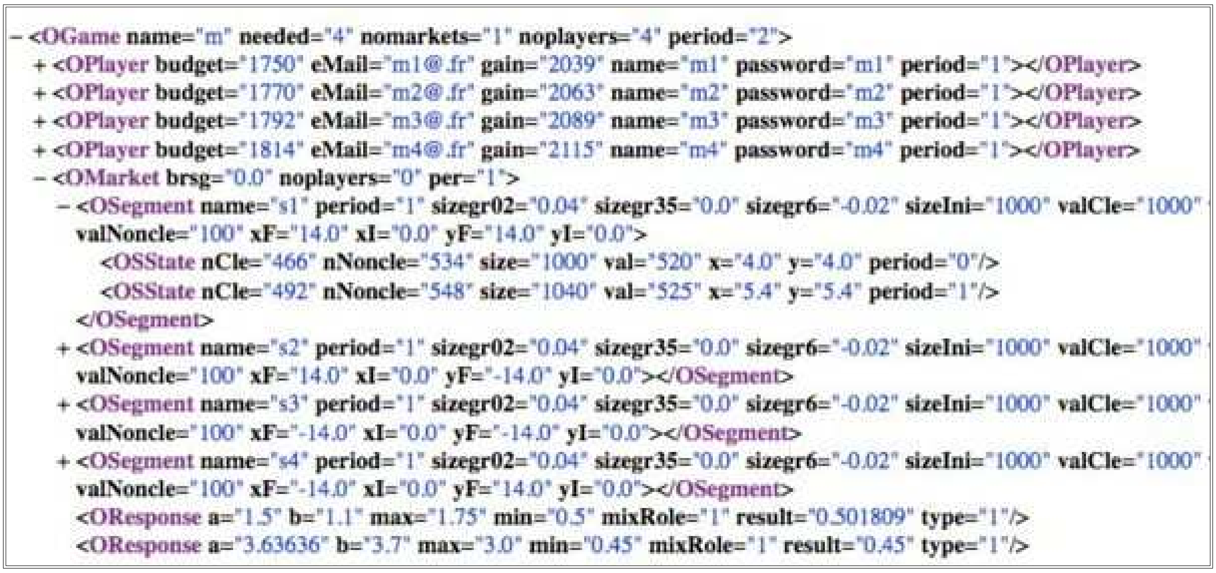


For the needs of our marketing simulation, XML binding has been organised at three levels in the game object hierarchy (see Figure 5).

At the first and *highest level* we have the whole game object containing the complete information of the industry. The game object is persistently stored into the g<gamename>.xml file. This file and the corresponding object is accessed and managed by the administrator (teacher) software.

[3] Marshalling is for a hierarchy of memory objects what parsing is for a hierarchy of xml tags. While marshalling and unmarshalling needs valid xml documents, parsing and serialisation uses only well formed xml documents

Figure 6 First level of xml data binding the g<gamename>.xml file

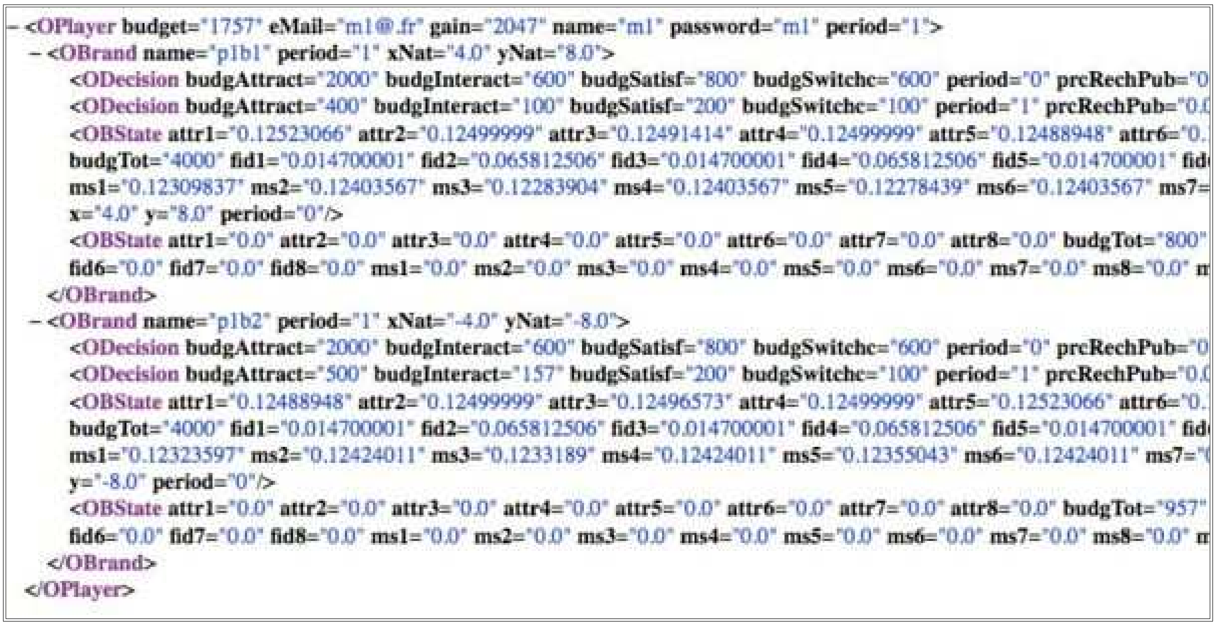


```
- <OGame name="m" needed="4" nomarkets="1" noplayers="4" period="2">
  + <OPlayer budget="1750" eMail="m1@.fr" gain="2039" name="m1" password="m1" period="1"></OPlayer>
  + <OPlayer budget="1770" eMail="m2@.fr" gain="2063" name="m2" password="m2" period="1"></OPlayer>
  + <OPlayer budget="1792" eMail="m3@.fr" gain="2089" name="m3" password="m3" period="1"></OPlayer>
  + <OPlayer budget="1814" eMail="m4@.fr" gain="2115" name="m4" password="m4" period="1"></OPlayer>
  - <OMarket brsg="0.0" noplayers="0" per="1">
    - <OSegment name="s1" period="1" sizegr02="0.04" sizegr35="0.0" sizegr6="-0.02" sizeIni="1000" valCle="1000"
      valNoncle="100" xF="14.0" xI="0.0" yF="14.0" yI="0.0">
        <OSState nCle="466" nNoncle="534" size="1000" val="520" x="4.0" y="4.0" period="0"/>
        <OSState nCle="492" nNoncle="548" size="1040" val="525" x="5.4" y="5.4" period="1"/>
      </OSegment>
    + <OSegment name="s2" period="1" sizegr02="0.04" sizegr35="0.0" sizegr6="-0.02" sizeIni="1000" valCle="1000"
      valNoncle="100" xF="14.0" xI="0.0" yF="-14.0" yI="0.0"></OSegment>
    + <OSegment name="s3" period="1" sizegr02="0.04" sizegr35="0.0" sizegr6="-0.02" sizeIni="1000" valCle="1000"
      valNoncle="100" xF="-14.0" xI="0.0" yF="-14.0" yI="0.0"></OSegment>
    + <OSegment name="s4" period="1" sizegr02="0.04" sizegr35="0.0" sizegr6="-0.02" sizeIni="1000" valCle="1000"
      valNoncle="100" xF="-14.0" xI="0.0" yF="14.0" yI="0.0"></OSegment>
      <OResponse a="1.5" b="1.1" max="1.75" min="0.5" mixRole="1" result="0.501809" type="1"/>
      <OResponse a="3.63636" b="3.7" max="3.0" min="0.45" mixRole="1" result="0.45" type="1"/>
```

As can be observed form figure 6 the game contains players and markets. The market contains segments and response. There are four positional segments each with changing temporal states with attributes like size, number of key and non key customers, ideal points (expectancies) with their x and y coordinates. As the segments have key and non key relational sub-segments which respond differently to the four marketing mix categories detailed below, the market also contains Response objects containing the coefficients of these segments' response functions.

The *second level* is the one of the player (or firm). The player object is persistently stored into the p<playername>.xml file. It contains all information about one player, that is everything a firm ignoring its competitors needs to know about its situation. This file and the corresponding object can be also accessed and managed by the player (firm)'s software.

Figure 7 Second level of xmlbinding the p<playername>.xml file



```
- <OPlayer budget="1757" eMail="m1@.fr" gain="2047" name="m1" password="m1" period="1">
  - <OBrand name="p1b1" period="1" xNat="4.0" yNat="8.0">
      <ODecision budgAttract="2000" budgInteract="600" budgSatisf="800" budgSwitchc="600" period="0" prcRechPub="0
      <ODecision budgAttract="400" budgInteract="100" budgSatisf="200" budgSwitchc="100" period="1" prcRechPub="0.0
      <OBState attr1="0.12523066" attr2="0.12499999" attr3="0.12491414" attr4="0.12499999" attr5="0.12488948" attr6="0.
      budgTot="4000" fid1="0.014700001" fid2="0.065812506" fid3="0.014700001" fid4="0.065812506" fid5="0.014700001" fid
      ms1="0.12309837" ms2="0.12403567" ms3="0.12283904" ms4="0.12403567" ms5="0.12278439" ms6="0.12403567" ms7=
      x="4.0" y="8.0" period="0"/>
      <OBState attr1="0.0" attr2="0.0" attr3="0.0" attr4="0.0" attr5="0.0" attr6="0.0" attr7="0.0" attr8="0.0" budgTot="800"
      fid6="0.0" fid7="0.0" fid8="0.0" ms1="0.0" ms2="0.0" ms3="0.0" ms4="0.0" ms5="0.0" ms6="0.0" ms7="0.0" ms8="0.0" m
    </OBrand>
  - <OBrand name="p1b2" period="1" xNat="-4.0" yNat="-8.0">
      <ODecision budgAttract="2000" budgInteract="600" budgSatisf="800" budgSwitchc="600" period="0" prcRechPub="0
      <ODecision budgAttract="500" budgInteract="157" budgSatisf="200" budgSwitchc="100" period="1" prcRechPub="0.0
      <OBState attr1="0.12488948" attr2="0.12499999" attr3="0.12496573" attr4="0.12499999" attr5="0.12523066" attr6="0.
      budgTot="4000" fid1="0.014700001" fid2="0.065812506" fid3="0.014700001" fid4="0.065812506" fid5="0.014700001" fid
      ms1="0.12323597" ms2="0.12424011" ms3="0.1233189" ms4="0.12424011" ms5="0.12355043" ms6="0.12424011" ms7="
      y="-8.0" period="0"/>
      <OBState attr1="0.0" attr2="0.0" attr3="0.0" attr4="0.0" attr5="0.0" attr6="0.0" attr7="0.0" attr8="0.0" budgTot="957"
      fid6="0.0" fid7="0.0" fid8="0.0" ms1="0.0" ms2="0.0" ms3="0.0" ms4="0.0" ms5="0.0" ms6="0.0" ms7="0.0" ms8="0.0" m
    </OBrand>
  </OPlayer>
```

As shown in figure 7 players have brands, brands have changing temporal states and decisions. Decisions refer to dividing budget between the four marketing mix ingredients (mentioned above) and fixing positioning objectives (xViz and yViz). States contain temporarily achieved brand positioning, attraction and loyalty indexes as well as market shares obtained by the brand on key and nonkey customers belonging to the four positional segments.

The third and last level is the collection of a player's brand decisions for the current period. This collection is persistently stored into a file called p<playername>p<period>decisions.xml. It is the document that gathers all information concerning the current decisions of a firm (here only brand decisions). This file and the corresponding collection can also be accessed and managed by the player. By editing this xml file directly or using an xml editor program this file offers a simple and quick way to collect decisions from players or allow the administrator (teacher) to take decisions on their behalf when some players are absent.

Presentation logic using XSLT transformations

The tree structure of XML documents can be very easily read and transformed by computer programs. One of the most important evolutions based upon this fact is the Extensible Stylesheet Language Transformation (XSLT) technology. XSLT is probably one of the most exciting XML technologies. It is a standard way to "transform" an XML document into another XML document by associating an XSL style-sheet that contains the transformation rules. Several linked transformations, where the output from one is input to the other, form a pipeline. Such pipelined transformations are seamlessly supported by some xml based application frameworks[4]. This makes it possible to use a Lego(tm)-like approach in building applications, hooking together components into pipelines without requiring programming.

Web browsers usually support simple not pipelined XSLT transformations. These are powerful enough to support the presentation logic of a marketing simulation or decision support application.

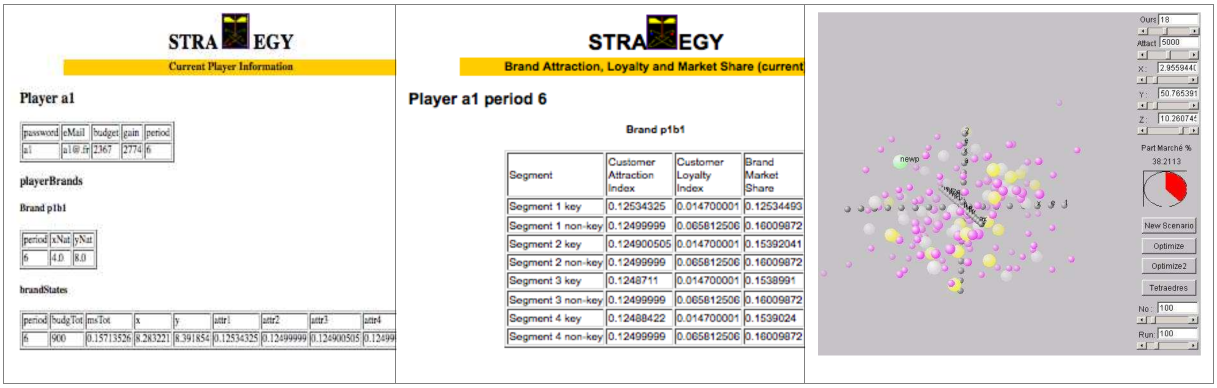
In the analysed application by associating various XSL style-sheets two key xml files are transformed into xhtml files that display on the browser media rich information extracted from them. Practically all the user interface that allows taking managerial decisions by getting all necessary information can be produced using such "simple" xslt transformations.

The two xml files are the player file containing information limited to each firm and the game file containing complete information about the industry. For both player and game the complete content can be extracted and displayed in a user friendly way in two versions: a historic version containing all brand and segments state information or a current version retaining only state information from the last period (as in figure 8a).

Each xml file is combined dynamically with several xsl files by specifying the names of those files as URL variables. The combination is obtained on the client side by using the browser's scripting language in a html file.

The player will be able to obtain information concerning his financial status, information concerning his brands like natural, intended and obtained positioning, or attraction, loyalty and market share (figure 8b), or even customer flows per segment.

Figure 8 information extracted by xsl stylesheets from a player's xml file



STRA EGY

Current Player Information

Player a1

password	eMail	budget	gain	period
a1	a1@.fr	2367	2774	6

playerBrands

Brand p1b1

period	xNat	yNat
6	4.0	8.0

brandStates

period	budgTot	msTot	x	y	attr1	attr2	attr3	attr4
6	900	0.15713526	8.283221	8.391854	0.12534325	0.12499999	0.124900505	0.12499

STRA EGY

Brand Attraction, Loyalty and Market Share (current)

Player a1 period 6

Brand p1b1

Segment	Customer Attraction Index	Customer Loyalty Index	Brand Market Share
Segment 1 key	0.12534325	0.014700001	0.12534493
Segment 1 non-key	0.12499999	0.065812506	0.16009872
Segment 2 key	0.124900505	0.014700001	0.15392041
Segment 2 non-key	0.12499999	0.065812506	0.16009872
Segment 3 key	0.1248711	0.014700001	0.1538991
Segment 3 non-key	0.12499999	0.065812506	0.16009872
Segment 4 key	0.12488422	0.014700001	0.1539024
Segment 4 non-key	0.12499999	0.065812506	0.16009872

Ours 18

Attact 5000

X: 2.9559440

Y: 50.765391

Z: 10.260745

Part Marché %

38.2113

No : 100

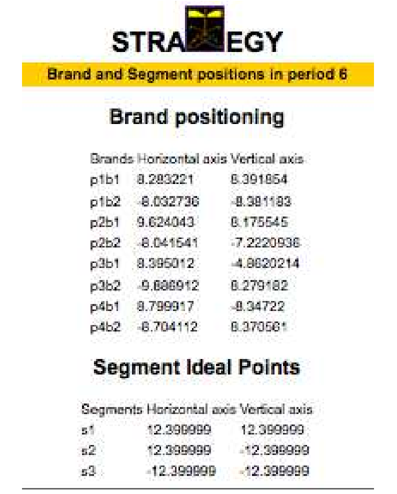
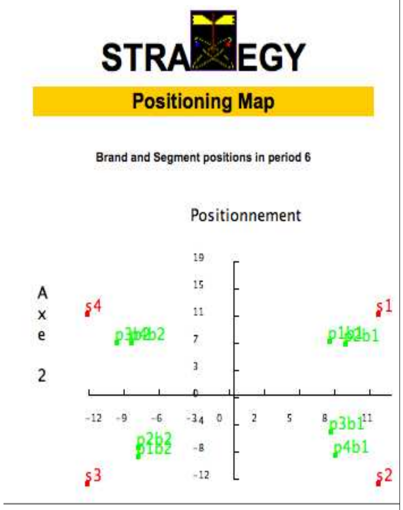
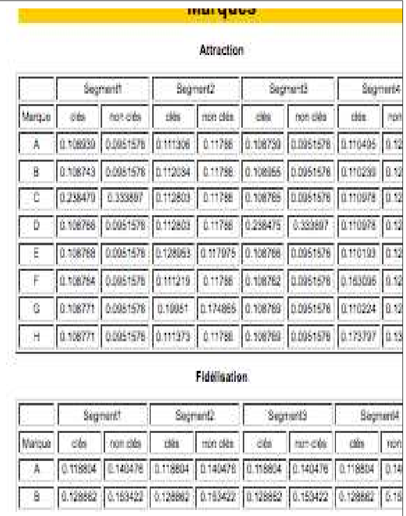
Run: 100

[4] see http://cocoon.apache.org

(a) Current information	(b) attraction, loyalty and share	(c) positioning decision support

The game administrator will be able to see information concerning all firms, brands, segments and customer flows. Some examples are shown in figure 9.

Figure 9 information extracted by xsl stylesheets from a game's xml file

		
(a) positioning tables	(b) positioning map	(c) attraction, loyalty index

STRATEGY

Brand and Segment positions in period 6

Brand positioning

Brands	Horizontal axis	Vertical axis
p1b1	8.283221	8.391854
p1b2	-8.032736	-8.381183
p2b1	9.624043	8.175545
p2b2	-8.041541	-7.2220936
p3b1	8.395012	-4.8620214
p3b2	-9.686912	8.279182
p4b1	8.799917	-8.34722
p4b2	-8.704112	8.370561

Segment Ideal Points

Segments	Horizontal axis	Vertical axis
s1	12.399999	12.399999
s2	12.399999	-12.399999
s3	-12.399999	-12.399999

Marques

Attraction

	Segment1		Segment2		Segment3		Segment4	
Marque	clés	non clés	clés	non clés	clés	non clés	clés	non
A	0.108939	0.0951576	0.111306	0.11788	0.108730	0.0951576	0.110495	0.12
B	0.108743	0.0951576	0.112034	0.11788	0.108955	0.0951576	0.110239	0.12
C	0.238470	0.333897	0.112803	0.11788	0.108785	0.0951576	0.110978	0.12
D	0.108766	0.0951576	0.112803	0.11788	0.238475	0.333897	0.110978	0.12
E	0.108768	0.0951576	0.128063	0.117975	0.108766	0.0951576	0.110193	0.12
F	0.108764	0.0951576	0.111219	0.11788	0.108762	0.0951576	0.163096	0.12
G	0.108771	0.0951576	0.19951	0.174865	0.108769	0.0951576	0.110224	0.12
H	0.108771	0.0951576	0.111373	0.11788	0.108769	0.0951576	0.173797	0.13

Fidélisation

	Segment1		Segment2		Segment3		Segment4	
Marque	clés	non clés	clés	non clés	clés	non clés	clés	non
A	0.118804	0.140476	0.118804	0.140476	0.118804	0.140476	0.118804	0.14
B	0.128862	0.153422	0.128862	0.153422	0.128862	0.153422	0.128862	0.15

Discussion and further developments

This paper introduces a document based marketing model implementation framework using object orientation and xml technologies that offers a quick way to bring marketing models to market. A marketing model that is well adapted to illustrate the approach has also been developed. The model is a stylised simulation that concentrates on essential marketing strategy aspects. Although this can be seen as a limitation, it serves the demonstrative purpose of this research. To our knowledge it is the only simulation model that combines both transactional and relationship marketing strategies in a dual approach producing varying attraction and loyalty effects on key and non-key customers.

We define document based programming as an approach that tends to replace most system implementation programming tasks with human and machine readable documents transformed by standard automated processes. While text or web html documents are known as being only human readable and binary files are only machine readable, xml documents are closest to what can be called standard human and machine readable documents.

The framework facilitates separation of concerns by allowing marketing analyst, managers and auxiliary staff to concentrate and use their specific skills in the project. It also favours openness and enables collaboration between potential participants.

The suggested implementation strategy is rather straightforward and natural. The mathematical formulation of the model is the starting point. Therefrom structural entities are defined and domain specific vocabularies are developed. Using standard mark-up this information is stored in human and machine readable documents. Based upon these documents, standard automatic processes generate the core model implementation logic. That logic is completed by human intervention which is essentially limited to the translation of the mathematical formulation to a computer language. The resulting application computes the model and mobilises another standard automatic process in order to persistently input and output all needed information again in human and machine readable documents. These documents are then dynamically combined by the web browser with human and machine readable transformation stylesheets using automated processes in order to organise the user interface of the application.

By using this framework we suggest a quick implementation strategy that produces a “thin” application which is closely sticking to the mathematical model and delegates most user interface and interaction aspects to widely available software and automated processes. The result is an offline application using the web browser on a local machine to support a document based user interface. This interface, that exploits the browser's continuously evolving hypermedia technologies, can become rather rich and satisfying for marketing decision support and simulation purposes.

The marketing model to which the framework is applied is a generic dual attraction/retention strategy model. It is implemented here as a simulation model, but it can also be used as a decision support model in order to optimise marketing mix allocation efforts in a dynamic programming approach. It can additionally integrate two decision support models for each period, one for the optimal positioning of a brand and one to find the optimal balance between attraction and retention efforts.

Both the model and it's implementation are and will be further developed in various ways. An online implementation can be found at http://claree.univ-lille1.fr/strateg_mono/. Some of these developments could occur collaboratively using some international inter-academic network by exploiting the openness of the suggested approach. The simulation model can evolve by extending both offensive and defensive strategy aspects. The first should include missing classical marketing mix aspects and the second should extend aspects concerning the impact of defensive mix on customer dynamics and value.

The model's implementation as a marketing simulation can use the “thin” application produced by the suggested framework as a common denominator and starting point for further extensions of the simulation logic. While basically an off-line simulation in which administrators have complete access and players have limited access to the persistently stored information, it can easily go on-line without any enhancements, by simply exchanging those files over the internet. Administrators can collect player files, run the simulation program for the next period and distribute the updated player files over the internet. This can be done by e-mail or by using some e-learning platform on which simulation sessions are scheduled and information files exchanged. Further enhancements can be easily built upon the approach presented here. The local offline application can evolve by automating the administrator's tasks towards a distributed client-server application. Document based persistency can be replaced with database persistency that supports highly interactive and multiple session simulations logic .

While the evoked framework and enhancements offer a quite satisfactory integration of model logic with the user interface and data access components of Little's (1979) now classical MDSS framework, the other components optimisation and statistics can also profit heavily from the document based programming approach. These components relay on the power of mathematical and statistical software and/or libraries which are often stand alone and need a document based programming approach to act as a glue between them and the core marketing model logic.

Marketing scientists while having good knowledge of statistics, econometrics, operations research, seem to have poor knowledge of modern programming and IT. This was not always so and is rather invalidating at least as concerns diffusion and adoption of marketing models. At the beginnings of MDSS, which somehow coincided with interactive computing, it was not rare to see MDSS implemented by the analysts themselves (see for example Choffray, 1985) using some programming language like BASIC. With the advent of micro- and personal computers and the era of office automation, spreadsheets and their “macro” programming language have often been and continue to be used to implement MDSS (Lilien, 1987; Lilien & Rangawamy, 2004). Paradoxically those technologies are far from having the power and elegance brought over by the object orientation paradigm nor the reach, ubiquity and openness of some of its embodyments like Java or Xml technologies. Despite these important evolutions, marketing scientists rarely implement their models using object orientation and to our knowledge Java and Xml technologies have not been much mobilised. Our research tries to fill this gap by retooling marketing model builders and aims to enforce research directions that favour IT aware marketing model implementations. In this way we hope to contribute to increased productivity in building model based marketing decision support and to facilitate adoption and use of marketing models by managers. Our document based framework and approach insists only on how model building and to a lesser extent model implementation can be facilitated. But the same technologies have much wider application fields. They enable marketing models to plug and "decision makers to tap into vast amounts of computing power and data sources on demand, thereby creating a new environment for the supply and demand for marketing analytics" (Lilien and Rangaswami, 2008, p.546).

References

Bultez, A. (1996), Mode de diagnostics de marchés concurrentiels, *Recherche et Applications en Marketing*, 11, 4, 3-34.

Bultez, A. (1997), Econométrie de la compétitivité: modèles et contre-exemples, *Recherche et Applications en Marketing*, 12, 1, 21-44.

Colombo, R. A., D.G. Morrison (1989), A Brand Switching Model with implications for Marketing Strategies, *Marketing Science*, 8,1, 89-100.

Choffray, J-M. (1985), *Marketing-expert: logiciels d'aide à la décision*. McGraw-Hill, Paris.

Eliashberg, J., G.L. Lilien(1993) Mathematical Marketing Models: Some Historical Perspectives and Future Projections, Eliashberg, J., G.L. Lilien, Eds. *Handbooks in Operational Research and Management Science: Marketing Vol. 5.* Elsevier Science Publishers, Amsterdam, 3-23.

Laréché, J-C., H. Gatignon (1990), *Markstrat 2 - A Marketing Strategy Simulation. Instructors Manual*, The Scientific Press, Redwood City;

Lilien, G.L. (1987), *Analyse des Décisions Marketing avec LOTUS 1-2-3*, (Traduit de l'americain par Pierre Y. Desmet), Economica, Paris.

Lilien, G.L., A. Rangaswamy (2000), Modeled to bits: Decision models for the digital, networked economy, Intern. J. of Research in Marketing, 17, 227–235

Lilien, G. , Rangawamy A. (2004), *Marketing Engineering: Computer-Assisted Marketing Analysis and Planning*, Revised Second Edition**,** Trafford Publishing, 538 p.

Lilien, G., A. Rangawamy (2008), Marketing Engineering: Models that Connect with Practice, Wierenga B., Ed. *Handbook of Marketing Decision Models*, Springer,527-559

Little, J.D.C. (1970), Model and Managers: The Concept of a Decision Calculus, *Management Science*, 16,8 (April), 467-485

Little, J.D.C. (1975), BRANDAID: A Marketing Mix Model, Part I: Structure; Part II: Implementation, *Operations Research*, 23, 628-673.

Little, J.D.C. (1979), Decision Support Systems for Marketing Managers, *Journal of Marketing*, Vol. 43, no. 3 (Summer), 9-27.

Rust, R.T., A.J. Zahorik (1993), Customer Satisfaction, Customer Retention and Market Share, *Journal of Retailing*, 69 (Summer), 193-215.

Rust, R.T., A. J. Zahorik, T.L. Keiningham (1995), Return of Quality (ROQ): Making Service

Quality Financially Accountable, *Journal of Marketing*, 59 (April), 58-70

Wierenga B., van Bruggen, G.H. (1997), The integration of marketing problem solving modes and marketing management support systems. *Journal of Marketing*, 61 (July), 21–37.

Wierenga B., van Bruggen G.H. And Althuizen N.A.P (2008) Advances in Marketing Management Support Systems, Wierenga B., Ed. *Handbook of Marketing Decision Models*, Springer, 561-592.