MODELLING iMODE SUCCESS WITH SYSTEM DYNAMICS

By

ADIL GEBORY

Masters Thesis
in
Information and Communication Technology

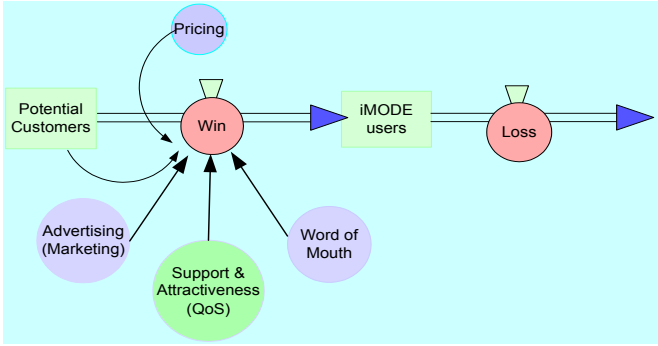
Agder University College
Faculty of Engineering & Science

Grimstad, Norway
May 2003

MODELLING iMODE SUCCESS WITH SYSTEM DYNAMICS

By

Adil Gebory



Supervisors

Prof. Dr. tech. *Jose J. Gonzalez,* Agder University College
Dr.degree st. *Agata Sawicka*, Agder University College

Published by

Agder University College, Communications & Information Technology
Grimstad, Norway
May 2003

Dedicated to My Parents, Sisters and Brothers

Declaration

I hereby declare that this work presented in this thesis is my own and that it has not been previously used to obtain a degree in this institution or elsewhere

_____

Adil Gebory

May 2003

FOREWORD

This paper is the whole part of my M. SC. Degree research in Information and Communication technology Department, Agder University College.

The work discusses system dynamics models for the dynamics of successful market growth for *iMODE*, mobile Internet services, which has been launched by *NTTDoCoMo* in Japan since February 1999.

The models in this report address the success growth of the total market for *iMODE* services for a newly launched mobile Internet service that is adopted by customers.

The work provides illustrative models for *iMODE* services based on the dynamics of *Word of Mouth*, advertising, pricing and quality of services.

Both differential equations and *PowerSim Studio* dynamics models notations are presented for these models in this report.

Using system dynamics thinking in my work leads to equations of a model, which has been simulated to understand various dynamic behaviours of alternatives policies.

I would like to thank my mother for her love, which has allowed this work to come into existence. Additionally, I thank my supervisors for their encouragement and advice, which provided me with the necessary direction to enable the realisation of this work.

Finally, I would like to thank all the staff in Agder University College for their help.

Adil Gebory
Oslo
Norway
May, 2003

CONTENTS

1. INTRODUCTION 7

2. PROBLEM DESCRIPTION: 8

2.1 IMODE SERVICES 9

2.2 TECHNICAL PERSPECTIVE 10

2.3 IMODE VERSUS WAP: 11

2.4 IMODE -SUCCESS: 12

2.5. SUCCESS FACTORS OF IMODE MARKET GROWTH MODEL: 13

3. MODEL PURPOSE AND USES: 14

4. REFERENCE BEHAVIOUR: 15

4.1 IMODE USERS GROWTH REFERENCE BEHAVIOUR 15

4.2 USAGE REFERENCE BEHAVIOUR 17

4.3 FINANCIAL REFERENCE BEHAVIOUR. 19

5. CAUSAL LOOP DIAGRAM & MATHEMATICAL BACKGROUND. 20

5.1 CAUSAL LOOP DIAGRAM OF ADOPTION FROM WORD OF MOUTH 23

5.2 ADOPTION FROM ADVERTISING EFFECT 23

5.3 PRICING STRATEGY CAUSAL LOOP DIAGRAM 24

5.4 SUPPORT & ATTRACTIVENESS CAUSAL LOOP DIAGRAM 25

5.5 FINANCIAL SEGMENT CAUSAL LOOP DIAGRAM 25

5.6 THE INFLUENCE DIAGRAM 27

6. STOCK & FLOW DIAGRAM (LEVELS & RATES) 32

6.1. POTENTIAL MARKET STOCK & FLOW DIAGRAM 34

6.2. ADVERTISING STRATEGY STOCK & FLOW DIAGRAM 37

6.3. PRICING STRATEGY STOCK & FLOW DIAGRAM 38

6.4. SUPPORT AND ATTRACTIVENESS (QOS) STOCK & FLOW DIAGRAM 38

6.5. FINANCIAL SEGMENT STOCK & FLOW DIAGRAM 39

7. MODELLING BY POWERSIM STUDIO 40

7.1 THE MIGRATION OF IMODE USERS 40

7.2 ADVERTISING EXPENDITURES EFFECT 41

7.3 PRICING STRATEGY 42

7.4 SUPPORT CUSTOMERS & ATTRACTIVENESS EFFORTS....42

7.5 REVENUE STREAM AND SEGMENTATION OF IMODE USERS....44

7.6 TOTAL COSTS (JPY)....45

7.7 NET INCOME & INCOME (JPY)....46

7.8 REVENUE, COSTS & INCOME (NOK) & (EURO)....46

7.9 SIMULATION THE MODEL....47

7.10 PRESENTATION MODE IN POWERSIM STUDIO....47

8. GRAPHICAL USER INTERFACE (GUI) BY JAVASWING....49

9. RESULTS....51

10. CONCLUSIONS....52

11. RESOURCES....53

11.1 REFRENCES....53

11.2 WEB SITES....53

11.3 APPLICATIONS....53

12 APPENDIXES....54

13 WHAT'S ON THE CD-ROM?....54

1. INTRODUCTION

This work is the whole part of a M. SC. Degree research in Information and Communication technology Dep., Agder University College, involving the study and analysis of *iMODE*, the wireless Internet cellular system. *iMODE* was launched in Japan in February 1999 by *NTT DoCoMo*, and is responsible for significant growth in the market. *NTT DoCoMo* is Japan's largest wireless carrier and the second largest in the World. During the research I discovered that, to date, there are more than 36 million users in Japan of *iMODE*-service; the majority of users are young people. However, a significant number of middle aged and elderly people also use the *iMODE* service.

NTT DoCoMo has created an attractive value chain and a mutually reinforcing ecosystem around the *iMODE* mobile Internet services.

Choosing *System dynamics* approach for this purpose has many advantages including:

- *System dynamics* is a powerful method to analyse why the managed systems do not behave as the users or vendors wish.
- *System dynamics* is supporting a strategic point of view.
- *System dynamics* consists of two phases, qualitative and quantitative.

System dynamics implies modelling a real-life phenomenon in terms of mathematical equations. Through millions of computations per second, the computer model can simulate the behaviour of given system through time. This makes it easy to understand and interpret how the system behaves.

Using *System dynamics* allows the more complex development models to investigate the innovation diffusions process. *System dynamics* is a methodology for studying and managing complex feedback systems.

The steps of our project are:

- Definition of the problem.
- Developing a dynamic hypothesis explaining the cause of the problem.
- Building a simulation model.
- Testing the model to establish if it reproduces the behaviour seen in the real world.
- Analysis the success factors.
- Policy recommendations.

2. Problem description:

System dynamics modelling simulation approach is used in scenario analysis of *iMODE* success, to study and understand the influence of uncertainty and time delays on how and at what speed they affect customer take-up of *iMODE* services. "i" in *iMODE* stands for information, internet and interactive.

The model investigates the success factors of *iMODE*, as a new product, which had been developed, tested and successfully introduced to the market on Feb, 1999.

Important factors relating to the take-up of *iMODE* services that are taken into account are:

- Word of mouth effect.
- Advertising effect.
- Pricing strategy.
- Quality of services, attractiveness and support.

iMODE is one of the most advanced services in the wireless Internet market, which was established in Japan. It was launched by *Nippon Telephone & Telegraph NTT DoCoMo* for the first time in February 1999. *NTT DoCoMo* is the only vendor offering the *iMODE* service in Japan.

NTT DoCoMo is a subsidiary of Japan's incumbent telephone operator NTT. The majority of *NTT DoCoMo* shares are owned by NTT, and the majority of NTT's shares are owned by Japanese government.

According to the research, I discovered that there are 828 companies providing information services based on *iMODE*, in addition there are about 1500 official web sites, which are compatible with *iMODE* protocol, and there are more than 60000 web sites that *IMODE* users can easy get access to. Currently there are over 36 millions people using *iMODE*-handsets.

iMODE offers access to fixed-line Internet services of several *Internet Service Providers* "*ISP*". *iMODE* users receive news and stock prices tailored to their personal needs.

iMODE subscribers can send and receive email, shop online, look at the weather forecast, look at sports results, load ringing melodies, play games, do online banking, online stock trading, purchase train and air tickets, download cartoon and images, look for restaurants and search for new friends.

iMODE is the network that allows continuous access to the Internet via an *iMODE* cellular phone.

2.1 *iMODE* services

iMODE offers e-mail services and access to web sites, and these web sites are two types: the official and unofficial web sites. The official web sites are most popular as they appear in the start menu when the cell phone connected. These official web sites can be segmented into four categories: transactional, databases, entertainment and others.

iMODE has the ability to offer numerous services to customers. E-mail and packet transmission services which include:

- Transactions capabilities such as:
  - Ticket reservation.
  - Airline information and reservation.
  - Purchasing books.
  - Mobile banking.
  - Mobile trading
  - Credit card information.

- Database access such as:
  - Restaurant guide.
  - Dictionary search and services.
  - Cooking recipe.
  - Mobile telephone directory.

- Entertainment such as:
  - Games download.
  - FM on-air.
  - Lottery.
  - Audio, music and songs.

- There are many others services such as:
  - Weather forecasts.
  - Sports news.
  - News.
  - Business technology news.
  - Horce racing information.
  - City information.
  - Transportation.
  - Rental opening.

2.2 Technical perspective

WAP, Wireless Application Protocol is a specification for presenting and interacting with information on wireless and other devices. *WAP* is an enabling technology based on the Internet client server, for presentation and transmission of information from the *World Wide Web* "*WWW*" and other applications utilising the *Internet Protocol* "*IP*" to a cellular phone device.

WAP is a protocol uses a special programming language is called *Wireless Mark-up Language* "*WML*", which enables the communication between the cellular phone device and the content on the Internet through a Gateway Centre, referred to as a *WAP Gateway*.

WAP Gateway is responsible for the conversion between *WML* and the wide used programming language on the Internet which is called *Hyper Text Mark-up Language* "*HTML"*. Figure 1.b illustrates the WAP network structure.

iMODE protocol is a layer over the ordinary Mobile voice system. The mobile voice system is based on "*Circuit-Switching*", meaning that in order to establish an Internet connection it is necessary to dial-up.

The *iMODE*-system is based on "*Packet-Switching*". This is advantageous in that the *iMODE* user is always connected. What is important to note here is that there is no delay to set up the connection; however, there is a delay period while the data is reaching the *iMODE* user. These delays depend on the size of the information and the network load.

When discussing the Internet, we may refer to the encoding of the web sites, which are compatible with *iMODE*-protocol. The web sites that are accessed by the *iMODE*-protocol use *Compact Hyper Text Mark-Up Language "cHTML",* which is a subset of ordinary "*HTML"*, and there is no need for a *Gateway Centre* to communicate with *Internet Content Provider*. Figure 1.a, illustrates the *iMODE* Network Structure.

Cellular phone cell for *WAP* users must support a WAP browser, but the *iMODE* users must be able to fetch and display information from *Compact Hyper Text Mark-Up Language "cHTML"*

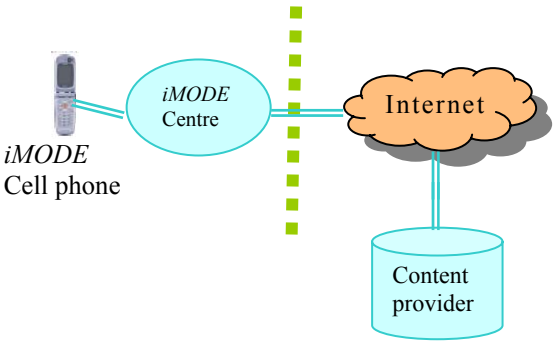


Figure 1.a *iMODE* Network Structure

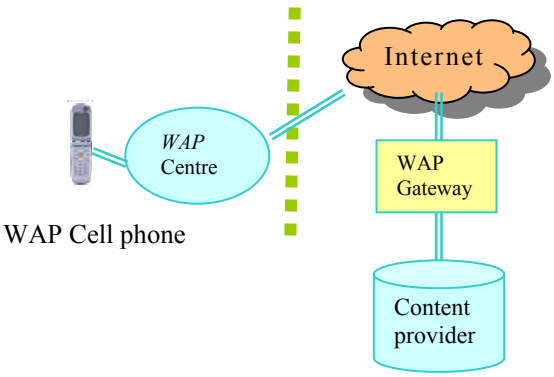


Figure 1.b *WAP* Network Structure

2.3 *IMODE* versus *WAP*:

IMODE provides high speed Internet access to mobile phone users. It had become a mass-market success. *WAP* had proved a disappointment for many, due to slow connections and limited content.

WAP describes a technology, which makes it possible to transfer information between Internet and mobile handsets, while *iMODE* is a business concept. The comparison between these two services leads us to three conceptions.

First, *iMODE* web sites use *cHTML* as an encoding language, which is a subset of ordinary *HTML*, and relatively is easier to learn and develop than the encoding language had been used in the *WAP*-protocol, which was called *wm l," Wireless application protocol mark-up language"*.

Not only is "*cHTML"* for *iMODE*, *simpler* than "*WAP"* for cell phones, but the *iMODE* website developers need only make one version of the website for all *iMODE* cell phones. WAP developers need to develop multiple versions of their website for different *WAP* cell phones.

Second, *iMODE* is implemented in "*Packet-Switched system*", i.e. is always on, eliminating the need to dial-up, while *WAP* is implemented in "*Circuit-Switched system*", i.e. it is necessary to dial-up.

Third and finally, *iMODE* subscribers are charged for the amount of information downloaded, whereas *WAP* users are charged by the connection time.

2.4 iMODE -success:

Arguably, there may be several factors attributing to *iMODE's* success in the Japanese market, and its eager acceptance by the Japanese people. Firstly, *NTT DoCoMo* built the *iMODE*-system to provide developers with an easy means of building web sites using this protocol.

The local access charges are very expensive in Japan, therefore, people preferred to access Internet services using their mobile handsets rather than a personal computers in their homes.

The report assumes that the possible reasons for *iMODE* success include:

- Japanese people love niceties.
- Mobile phone penetration.
- *Killer application*, email services.
- Costs to subscribers for *iMODE* enabled handsets at point of purchase.
- *IMODE* uses "Packet-Switched system", meaning it is always on.
- *IMODE*-protocol uses *cHTML* as a programming language for the *iMODE* web sites.
- Fashion and efficient marketing.

iMODE mobile Internet services had accumulated more than 35 millions subscribers by the end of year, 2002. *NTT DoCoMo*, the vendor of these services, still enjoys phenomenal growth rates.

iMODE services have an average signing rate greater than 50000 new *iMODE* users per day.

All the outlined and discussed thus far lead to several questions:

- Why has *iMODE* mobile Internet services been so successful?
- What are the factors that impact on the market growth success of these services?
- What attracts new people to adopt *iMODE* mobile Internet services?
- How does *NTT DoCoMo* serve their *iMODE* users, and retain them, so they do not abandon the service?

The report examines this success and the foundations which make the market growth of *iMODE* services so obvious.

2.5. Success factors of *iMODE* market growth model:

The success factors that arguably influence *iMODE*-product diffusion processes are:

1. The structure of the market is *monopolistic*.
2. Success factors such as pricing, quality, advertising or delivery delays have a direct impact of the purchase probability.
3. The substitution among successive *iMODE*-product and time varying market potential or negative word of mouth.
4. Using *iMODE* to access the Internet is cheaper than using the fixed network in Japan, i.e. the e-commerce cost higher than m-commerce, which is provided by *iMODE*.

Figure.2 illustrates these success factors:

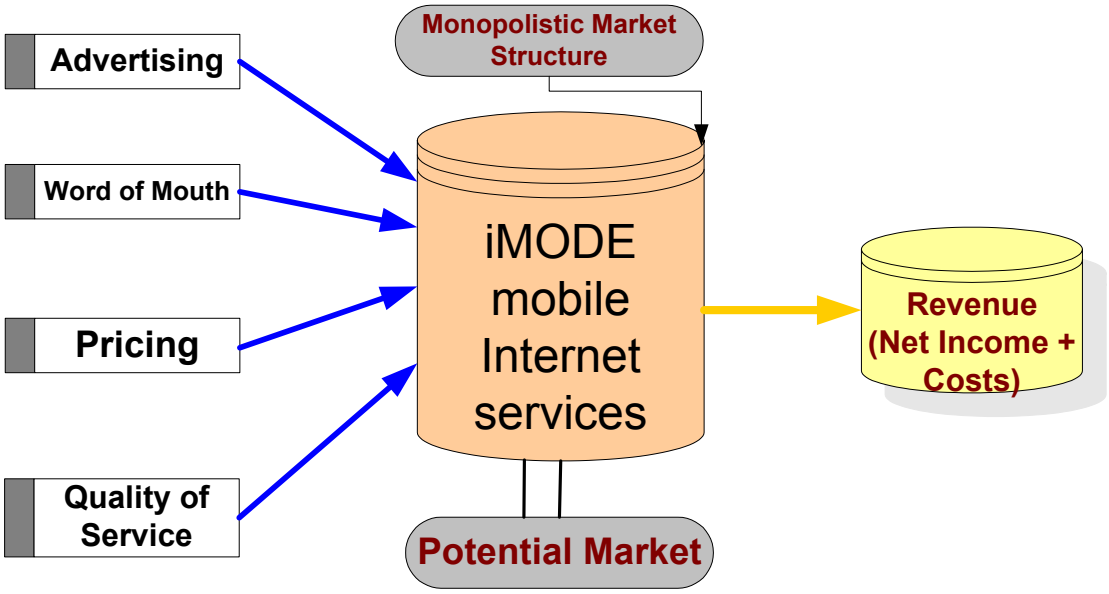


Figure 2 The success factors that influence *iMODE* market growth

3. Model purpose and uses:

Telenor and *NetCom* are going to launch the third generation mobile services in Norway. The analysis and model highlights the issues of the *iMODE*-system which could provide a useful tool in order to identify and adopt the relevant *iMODE* success factors.

Journalists have written many articles about *iMODE* and *WAP,* however, they have failed to identify practical products to support their comments. Their writings have been based according to theoretical knowledge and have been based on speculations.

In the case, i.e. *iMODE*-analysis, the thesis is going to study and analyse the success factors dynamically. The efforts on this problem will potentially provide a good base for vendors who are introducing new products into the market, particularly in the telecommunications field. The key objective here is to establish how and at what speeds will the customer take-up these new products.

4. Reference behaviour:

The thesis is dealing with a market, which means it is dealing with two sides. The supply side that is responsible for delivering services, and the demand side that is using these services.

The market has different perceptions. The supply side distinguishes between long-term and short-term solutions, while the demand side does not distinguish between these solutions. Perceptions of the market are based on determining the success of technology. Promising too much too soon can have serious adverse effects on market development.

4.1 *iMODE* users growth reference behaviour

Table 1 shows the growth of *iMODE* in Japan as relationship between time and *iMODE* users number.

Time	*iMODE* users
22.02.1999	*IMODE* service launched
31.03.1999	100000
30.04.1999	200000
31.05.1999	300000
28.06.1999	500000
31.07.1999	900000
08.08.1999	1300000
15.10.1999	2300000
15.03.2000	5000000
06.08.2000	10000000
22.11.2000	15000000
11.02.2001	19115000
18.02.2001	19398000
25.02.2001	19675000
04.03.2001	20015000
25.03.2001	21356000
01.04.2001	21753000
08.04.2001	22086000
15.04.2001	22412000
22.04.2001	22681000
09.09.2001	27139000
16.09.2001	27239000
23.09.2001	27553000
30.09.2001	27768000
07.10.2001	27925000
01.02.2002	31300000
01.03.2002	32200000
01.04.2002	32600000
01.05.2002	33000000
01.06.2002	33500000
01.07.2002	34100000
25.08.2002	34385000
01.09.2002	34449000
08.09.2002	34530000

Table. 1 *iMODE* user's progress

[Ref. http://www.nttdocomo.com/new/contents/01/whatnew0305.html]

In just three years since *iMODE* services has been launched by *NTTDoCoMo* in February, 1999. *NTTDoCoMo* has signed more than 35 millions accounts, i.e. there are more than 35 millions people use *iMODE* mobile Internet services. This number represents more than 25% of the whole population in Japan which is approximately 115 millions people.

The following milestones might be addressed as a reference to this work, and a good base to build the market growth of *iMODE* mobile Internet services.

- *NTTDoCoMo* has accumulated more than 60000 web sites which can be accessed by *iMODE* mobile handsets.
- The stream of Revenue and the robust business model encourages the usage which is represented by data packet transmission.
- The solid profitability to appeals to prospective customers to adopt *iMODE* mobile Internet services.

The graph in figure **3.a**, illustrates the growth of *iMODE* market in Japan, and shows the relationship between *iMODE* user's number and time.

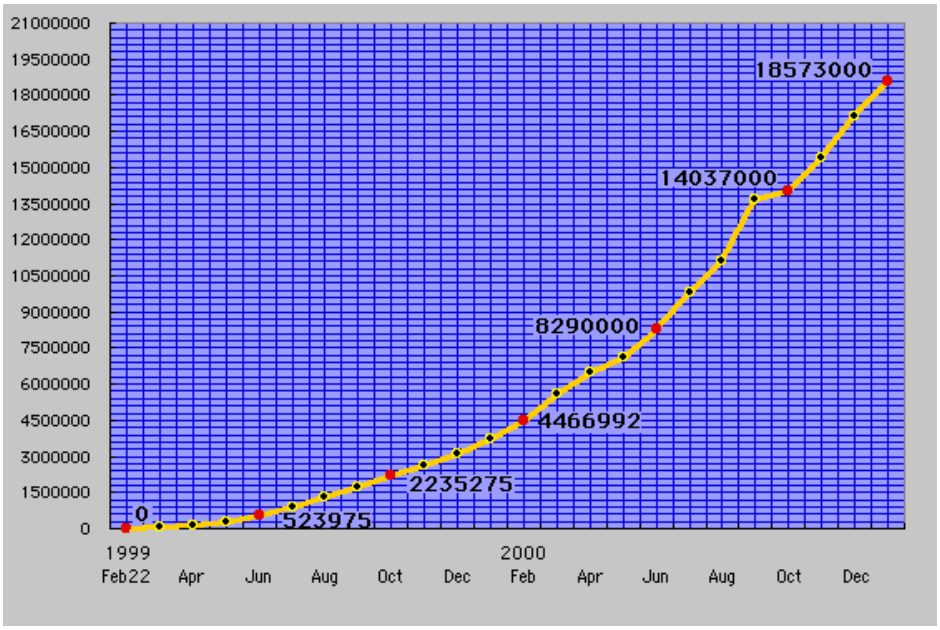


Figure 3.a *iMODE* users as a function of time

[Ref. http://www.nttdocomo.com/new/contents/01/whatnew0305.html]

4.2 Usage reference behaviour

Data packet transmission for contents and applications might be a foundation for the *iMODE* services growth market.

Data packet transmission gives *iMODE* users the ability to send and receive emails on mobility, as well as they can check news headlines and trade stocks, and as well many other entertainment and *Killer applications* services.

iMODE users can be segmented by their age in three categories:

1. Those who are younger than 30 years old and they represent 50% of the whole *iMODE* market.
2. Those who are between 30 and 39 years old and they address 20% of the whole *iMODE* market.
3. Those who are older than 40 years old and they address 30% of the whole *iMODE* market.

The diagram in figure **3.b** illustrates the segmentation of *iMODE* users by their age.

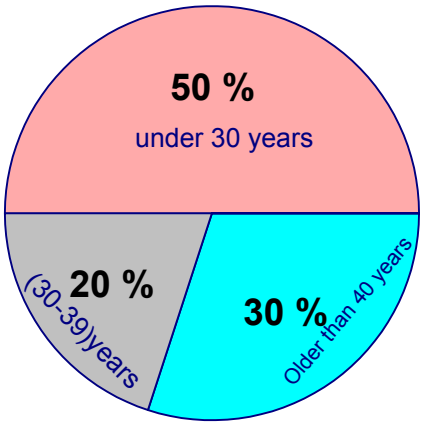


Figure 3.b The segmentation of *iMODE* users by their age

An alternative view of statistical data is the number of *iMODE* users classified by the usage of *iMODE* data packet transmission which comprises of the following categories:

- Mail accounts; represent 36% of the whole usage.
- *NTTDoCoMo* official web sites; represents 32% of the whole usage.
- Unofficial web sites which can be accessed by *iMODE* handsets; represents 32% of the whole usage.

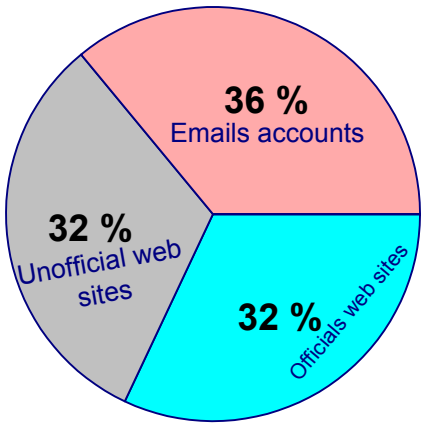


Figure 3.c The segmentation of *iMODE* users by their usage

The official web sites and email accounts represent the core of *iMODE* data packet transmission services, and address approximately 70% of the whole stream of revenue as shown in the diagram of figure **3.c**.

The official web sites services can be classified into four categories as:

1. Transactions.
2. Database.
3. Information.
4. Entertainment.

These categories had been described specifically earlier in our report, in section 2.1 .

The diagram in figure 3.d shows the segmentation of official web sites usage services.

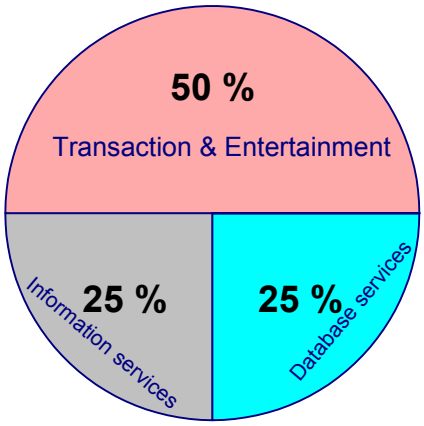


Figure 3.d The segmentation of official web sites usage services

4.3 Financial Reference Behaviour.

As mentioned earlier in this report, *NTT DoCoMo* is one of the mobile Internet services providers in Japan, which has become the largest by providing iMODE mobile Internet services from the wireless Internet.

The main idea is the flat rate charge, such as monthly fixed fee and pay per usage. When it talks about usage, we mean the amount of data (1 packet = 128 byte).

There are three main revenue streams for *NTT DoCoMo*'s *iMODE* services which are driven from *iMODE* users.

The revenue stream can be addressed by the different fees that *iMODE* users pay. These fees can be divided into three categories:

1. Fixed fee each month to access the services itself.
2. Data traffic fees that *iMODE* user generate.
3. A monthly fee for any content that *iMODE* users subscribe, from which *NTT DoCoMo* receive commission fees.

The basic subscription fee an *iMODE* user pay to access *iMODE* services is 300 yen per month. (1 YEN = 60 Norwegian Krone).

The data traffic fees or the usage fees depend on how much traffic an iMODE user has generated by accessing the mobile Internet network.

The data traffic fees occur when an *iMODE* user accesses a web site, sends an email, or downloads data. The price a user pays for such traffic is 0.3 YEN per packet.

The last category is the commission fees which are accumulated from the unofficial web sites providers; the commission percentage is 9% which is streaming into the revenue of *NTT DoCoMo*. Figure 3.e illustrates the stream of iMODE services revenue.

Fixed Fee	**300 YEN per USER per MONTH**
Usage Unit Fee	**0.3 YEN per PACKET**
Commission Fraction Fee	**9%**

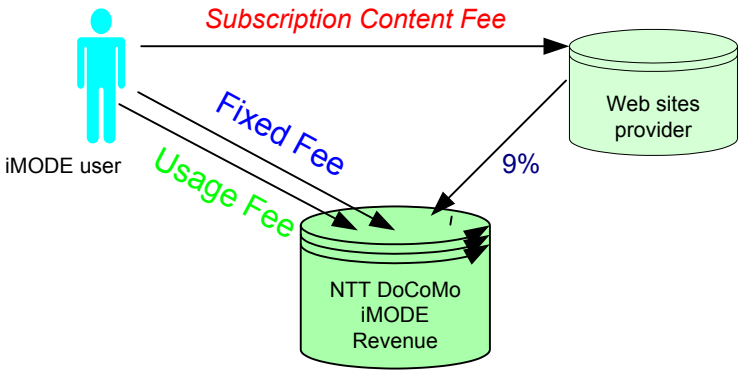


Figure 3.e Stream of iMODE services revenue

5. Causal Loop Diagram & Mathematical Background.

Causal Loop Diagram "*CLD*" is the method deployed to improve ones comprehension of how the business processes in our model work. The *CLD* illustrated in figure 4.

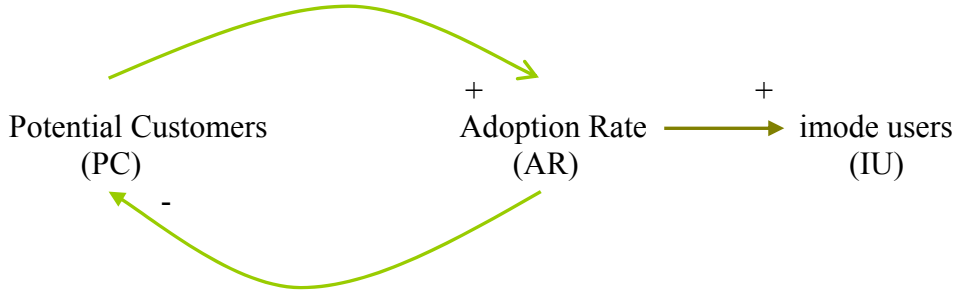


Figure 4, Causal Loop Diagram, Potential Customers

In this specific case, there is a pool of Potential Customers "PC" who may adopt *iMODE* services, are turned into *iMODE* users "IU" by an Adoption Rate "AR".

Potential Customers and Adoption Rate are connected in a negative feedback loop with the goal of driving Potential Customers to zero.

An increase in the number of Potential Customers, leads to an increase in Adoption Rate. This is illustrated by the positive sign between PC and AR in figure 4.

Since the Potential Customers are connected to *iMODE* users by Adoption Rate, then an increase in Adoption Rate leads to decrease in Potential Customers as is showed by the negative sign between Potential Customers and Adoption Rate.

The model assumes that for a short period of time ($\Delta\tau$) each *iMODE* users converts ($\beta.\Delta\tau$) Potential Customers into *iMODE* users, where β is a constant that encodes the effectiveness of *iMODE* users are at converting Potential Customers.

If at time (τ) there are $\{\lambda(\tau)\}$ *iMODE* users, and each of which converts ($\beta.\Delta\tau$) Potential Customers into *iMODE* users during the next period of time ($\Delta\tau$), then at time equal to ($\tau + \Delta\tau$), the number of *iMODE* users will be:

$$\lambda(\tau + \Delta\tau) = \lambda(\tau) + \lambda(\tau).(\beta.\Delta\tau)$$

By the calculus solution methods, the equation above can be converted to the following differential equation:

$$d\lambda(\tau) / d\tau = \beta.\lambda(\tau)$$

Then, and by integral both sides of this equation,

$$\lambda(\tau) = \beta_o + \int_0^{\tau} \beta . \lambda(\tau)\, d\tau$$

Where $\beta_{o\ is}$ the number of *iMODE* users at $(\tau = 0)$.

And the solution of the integral is the well known exponential function:

$$\lambda(\tau) = \beta_o . e^{\beta\tau} \quad , \qquad \text{for } \tau \geq 0$$

Figure 5, illustrates this Exponential growth function.

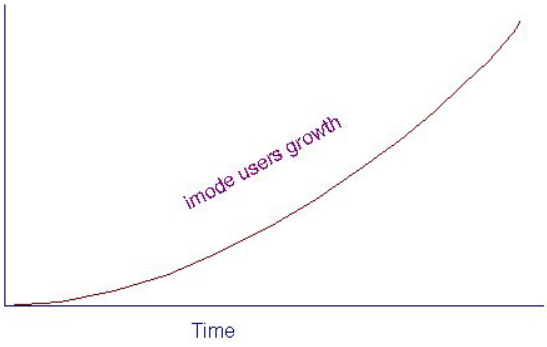


Figure 5 Exponential Growth of imode users

The above analysis can not be applied in realistic markets, because the exponential growth can not continue forever in the real world. [*See figure 7*]

Then, it considers that the model to be more realistic by assuming that the total number of Potential Customers of *iMODE* services is (Γ). That leads to the remaining number of Potential Customers at time (τ) will be $\{\Gamma - \lambda(\tau)\}$, where $\{\lambda(\tau)\}$ is the number of *iMODE* users at time (τ).

Assuming that (β) represents the Adoption Rate, then the number of Potential Customers converted by each *iMODE* users in a time period ($\Delta\tau$) is:

$$[\{\Gamma - \lambda(\tau)\} / \Gamma] . (\beta.\Delta\tau)$$

And Potential Customers by all $\lambda(\tau)$, *iMODE* users are:

$$[\{\Gamma - \lambda(\tau)\} / \Gamma] . (\beta.\Delta\tau) . \lambda(\tau)$$

By the calculus solution methods, the equation above can be converted to the following differential equation:

$$d\lambda(\tau) / d\tau = \beta . [\{\Gamma - \lambda(\tau)\} / \Gamma] . \lambda(\tau)$$

Then, and by integrating both sides of this equation,

$$\lambda(\tau) = \beta_o + \int_0^{\tau} \beta.\,[\{\Gamma - \lambda(\tau)\} / \Gamma]\,.\,\lambda(\tau)\,d\tau$$

Where β_o is the number of *iMODE* users at $(\tau = 0)$.

And the result of this integration produces the well known exponential function:

$$\lambda(\tau) = \Gamma / \{1 + [(\Gamma - \beta_o) / \beta_o]\, e^{-\beta\tau}\}, \qquad \text{for } \tau \geq 0$$

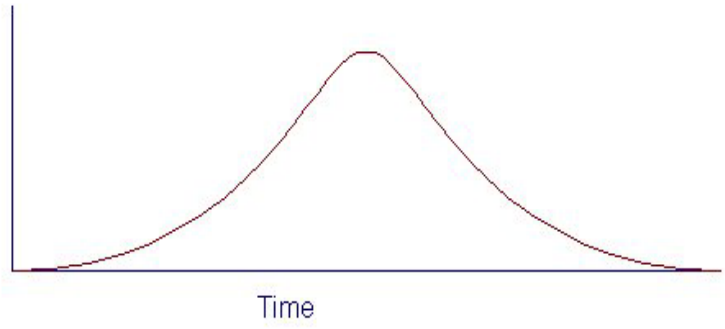


Figure 6 Adoption Rate of imode users

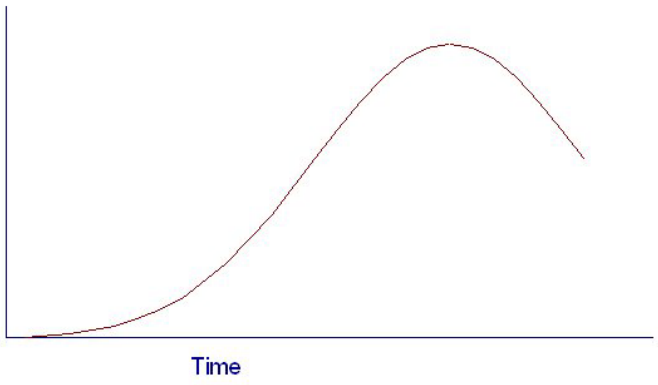


Figure 7 The S-Shaped growth of imode market

5.1 Causal Loop Diagram of Adoption from Word of Mouth

CLD of Adoption from Word of Mouth shows the feedback loop that control the Adoption Rate due to Adoption from WOM (Word of Mouth),

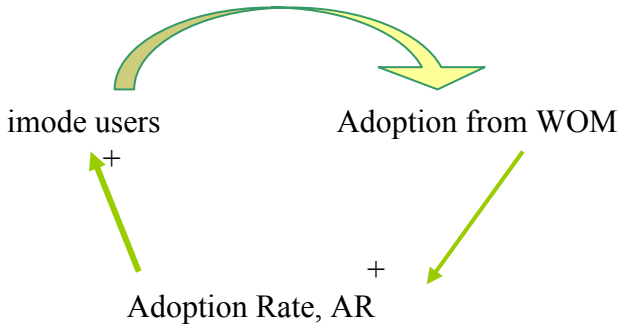


Figure 8, Causal Loop Diagram, Adoption from WOM

CLD in figure 8 illustrates the positive feedback loop affecting Adoption Rate due to adoption from WOM. The Adoption Rate increases the number of *iMODE* users, resulting in an increase in Adoption from WOM. An increase in Adoption from WOM, in turn increases the Adoption Rate.

5.2 Adoption from Advertising Effect

Adoption from Advertising has a direct link to Adoption Rate which is affected by Advertising Expenditures Effect.

Advertising Expenditures Effect is the efforts *NTTDoCoMO* spend to get their service well known by the Potential Customers, in the hope of leading them to adopt services.

Figure **9.a** illustrates the Adoption from Advertising CLD.

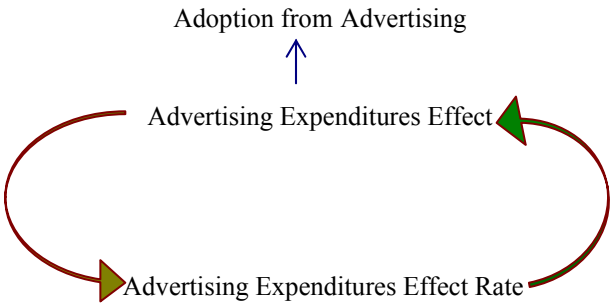


Figure 9.a, Causal Loop Diagram, Adoption from Advertising

5.3 Pricing Strategy Causal Loop Diagram

CLD of pricing strategy shows the feedback loop that controls the Pricing Rate due to Price Control.

Figure 9.b illustrates the positive feedback loop affecting Pricing Rate due to Price Control. Pricing Rate increases Simulated Packet's Price result an increase in Price Control. An increase in Price Control, in turn increases Pricing Rate.

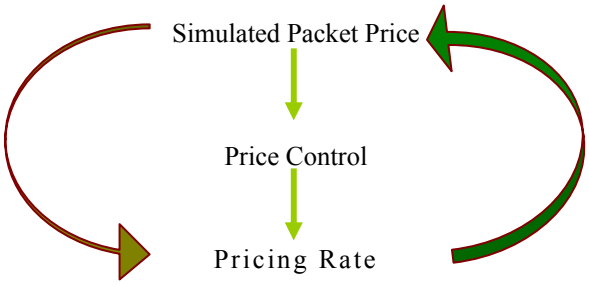


Figure 9.b, Causal Loop Diagram of Pricing Strategy

The report assumes that Price Control affects by two other constants factors, these factors are:
Min Price is the minimum packets price and Max Price is the maximum packets price.

Price Control depends on the Penetration fraction which is the fraction of the total market size that has been adopted *iMODE* services.

There are two other constants factors that are worth taking into consideration here. These factors are the Initial Price and the Price Change Delay which is the time period that *NTTDoCoMo* change their packet's price.

The whole analysis of this segment will be illustrated in the next chapters of this work.

5.4 Support & Attractiveness Causal Loop Diagram

The main idea behind this segment of the model is the analysis of the attractiveness policy according to two types of users. Those types represent current users of iMODE services which have been called *iMODE* users in the model, and new users which have been called Potential Customers in this model.

CLD of support and attractiveness shows the feedback loops that control the Attractiveness Costs. One concept in this part of the *iMODE* growth model is the assumption that the Average Usage is the amount of packets uses by customers, *iMODE* users, and Attractiveness Efforts has a direct connection to the Attractiveness Costs Rate.

There is a pool between Attractiveness Costs Rate into Attractiveness Costs, which is the costs to attract the current *iMODE* users or the potential customers. However, *iMODE* users are those who have adopted *iMODE* services, and Potential Customers are those who have not adopted the services yet.

The assumption concerning this part is that the price, i.e. Simulated Packet Price, influences both new users and current *iMODE* users, and there are a set of features appealing to those users.
Figure 9.c demonstrates this strategy.

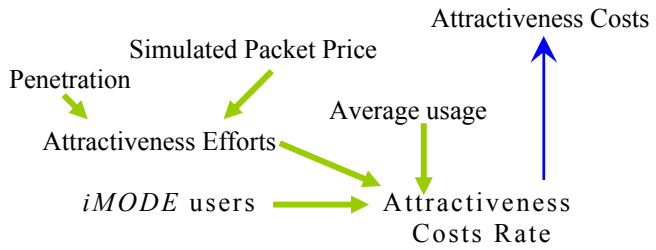


Figure 9.c, Causal Loop Diagram of Support & Attractiveness

5.5 Financial segment Causal Loop Diagram

The main factors in this part are revenue, costs and net income. Revenue is the total fees *iMODE* users pay to access *iMODE* services, and the costs are the amount of money *NTTDoCoMo* spends to serve and attract customers.

Net income can be given, in general, by the following differential equation:

Net Income = Total Revenue – Total Costs

Net Income represents the total earning *NTTDoCoMo* get from serving their services to the *iMODE* users.

Figure 9.d demonstrates the total costs policies. Attractiveness Costs Rate has a connection to three different variables. These variables are *iMODE* users, Attractiveness Efforts and Average Usage. Average Usage is the average of three

different categories of usage amount; those categories are relative with segmentation of *iMODE* users by their age.

All these variables and the whole causal loop diagram for this segment are illustrated in figure **9.d.**

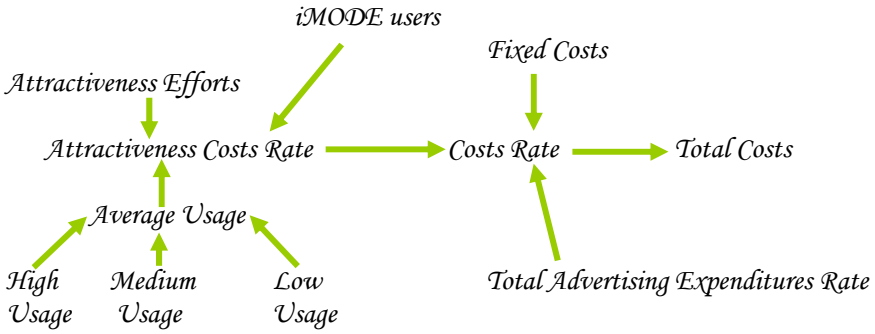


Figure 9.d Total Costs Causal Loop Diagram

These categories of amount of usage, i.e. the number of total packets *iMODE* users consumed, are:

- High Usage relatives to *iMODE* users who are under 30 years old.
- Low Usage relatives to *iMODE* users who are between 30 and 39 years old.
- Medium Usage relatives to *iMODE* users who are more than 40 years old.

Other assumptions that must be considered are:

- Fixed Costs is a constant amount of money *NTTDoCoMo* uses for their staff and maintains their network.
- Total Advertising Expenditures Effect is the amount of money *NTTDoCoMo* uses in advertising and marketing their services.

The total revenue stream is comprised of charges that *iMODE* users pay to avail of these services. These charges are can be paid by customers by means of different alternatives including:

- A monthly fixed fee an *iMODE* user pays to access services.
- Usage fee an *iMODE* user pay for the amount of packets has been used.

Therefore, a flat rate charge, and can be expressed by the following equation

Rate of Revenue = Fixed Fee Revenue + Usage Fee Revenue

The causal loop diagram of this part, and, in general, can be sketched as shown in figure 9.e.

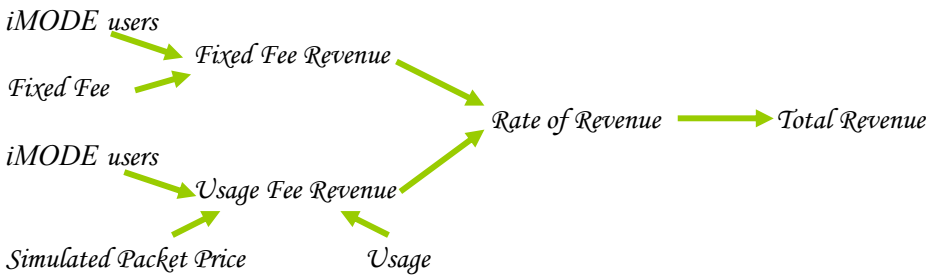


Figure 9.e Financial Segment Causal Loop Diagram

Usage represents low, high or medium usage. The usage type is dependant on the assumption it has been made, which in-turn is relative to the fractions of iMODE users. *iMODE* users might be those who are under 30, those who are 30-39 years old or those who are over 40 years old. This strategy, along with more specifications and assumptions will be described in greater detail in the next chapter.

Subsequent to demonstrating both total revenues and total costs policies, the straight ahead equation which represents Net Income of firm can be described.

Net Income = Total Revenue – Total Costs

The report assumes that there is amount of taxes must be paid by *NTTDoCoMo* to the government. Income is equal to Net Income minus the tax fraction multiplied by Net Income. This expression can be given by the following calculus.

*Income = Net Income - (Net Income * Tax Fraction)*

5.6 The influence diagram

The work starts the model by sketching the complete influence diagram, which helps it to get into the main functions and details of the problem strategy. The influence diagram gives us a good background, providing an outline of how the model should look, and makes it easier to build the conceptual model which helps us think about different issues around the growth of *iMODE* market. The work uses *Powersim Studio* to build the conceptual model.

Analysing the *iMODE*-market model provides both the positive and negative feedback as it mentioned in Causal Loop Diagram part of this paper. The positive feedback is used to generate the initial exponential growth of the success of *iMODE*-market. The negative feedback shows the factors that limit its growth.

The growth of *iMODE*-market can be viewed as an expansion by positive feedback. Here the focus is on those who have adopted it, and those who have not. Individuals who have adopted it and are satisfied recommend the service to those who have not.

The customers who have adopted the *iMODE*-services may not attempt to talk about it to Potential Customers (PC); however, the degree of attention paid to *iMODE*-services by Potential Customers and those who communicate with them depends on both the number of *iMODE users* and the number of PC. According to this analysis, the *iMODE*-services become more widely adopted, and *iMODE* users grow.

Figure 10.a, shows the complete influence diagram for iMODE potential market. Hence figure 10.b, shows the segment of the total Advertising process of the model.

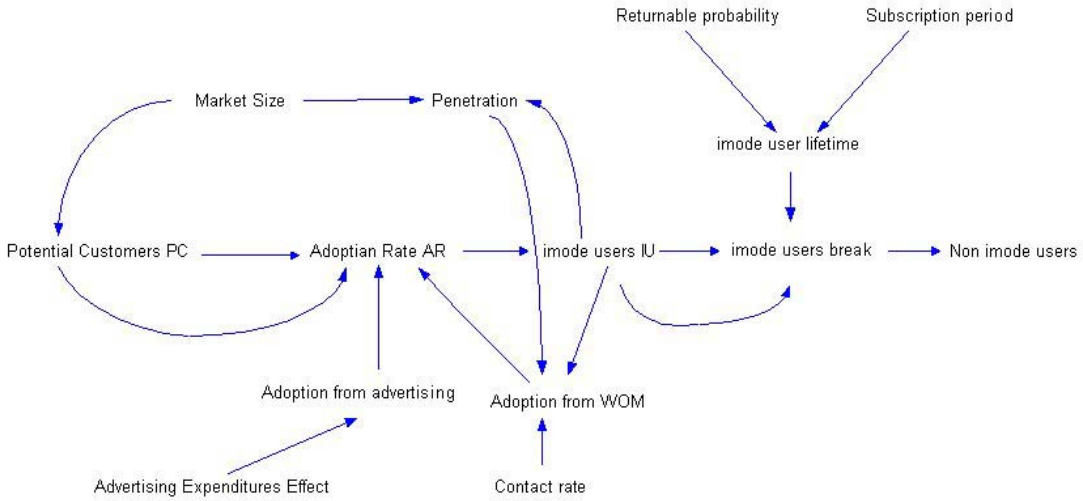


Figure 10.a Influence diagram of the potential market growth of *iMODE*-services

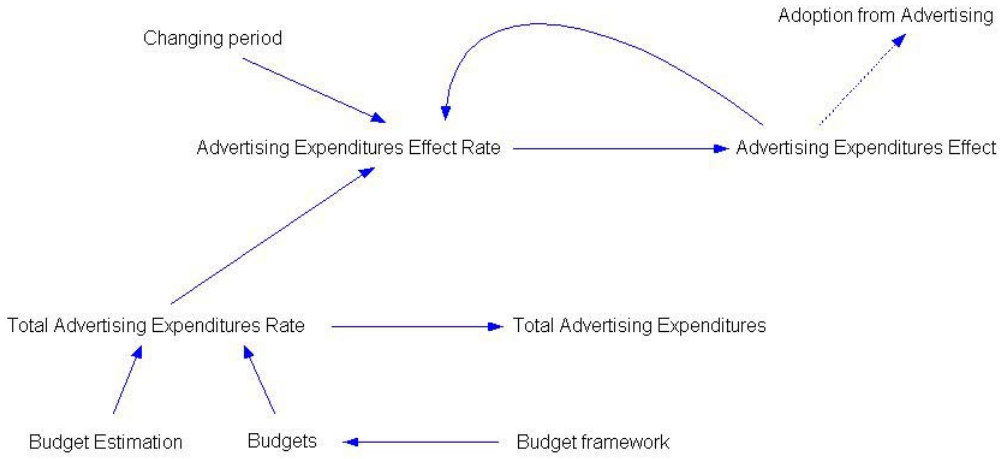


Figure 10.b Influence diagram of Advertising Expenditures Effect on the Adoption from Advertising

Influence diagram of the model, which is concerned with the potential market growth of *iMODE* services is shown in figure **10.a**. The work assumes that there are number of variables that affect the model process, thereby making it more practical and realistic.

These variables are:

- Subscription Period is the duration of time *iMODE* user sign up iMODE services
- Returnable Probability is the fraction of a NON- *iMODE* user who might return to adopt *iMODE* services, i.e. might be an *iMODE* user again.
- Market Size is the expectation of how many customers might be adopting iMODE services over period of time.
- Penetration is the fraction of market size that has adopted *iMODE* services.
- *iMODE* user lifetime is the average number of years that the user adopts *iMODE* services, and this variable relates to Subscription Period and Returnable Probability.

$$\textit{iMODE}\text{ user lifetime} = \frac{\text{Subscription Period}}{\{1 - \text{Returnable Probability}\}}$$

There are further probable factors considered for the Advertising Expenditures Effect part of the influence diagram. These variables include:

- Changing Period which is the time necessary for the information about *iMODE* services as a new product to spread between the people.
- Budget Estimation is the control enabling manual or automatic selection of the advertising expenditures budget.
- Budget Framework is the default settings for the manual advertising budget.
- Budgets emcompas two elements, the first one is the automatic budget for the advertising expenditures while the other one is the manual budget for advertising expenditures.

Another part or factor in the model is the pricing strategy, this has been mentioned previously in the report, and it provided the base to sketch the influence diagram of this segment.

The total influence representation of this part is shown in figure 10.c

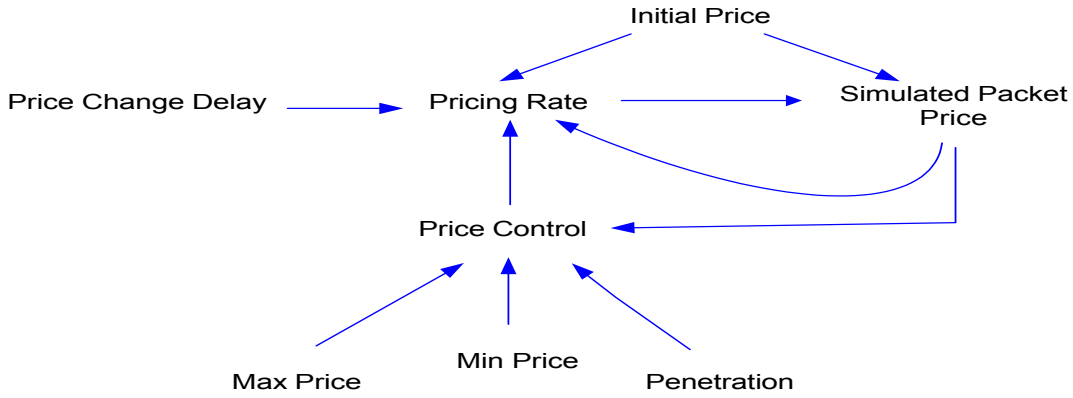


Figure 10.c Influence diagram of the Pricing Strategy

The influence diagram shows the positive feedback loop between Price Control and the Pricing Rate.

Support and Attractiveness part of the model and as it mentioned earlier in this report with some other assumptions give a base to sketch the influence diagram of this segment.

The report assumes that the Average Usage, which is the average amount of packets *iMODE* users use depends on three variables represents in Low, High and Medium usage.

This strategy depends on the analysis that *iMODE* users can be segmented in three different groups by their age. Meaning that those who are under 30 years old can be classified under the high usage group, those who are between 30 and 39 years old can

be treated as a low usage group. Finally, those who are more than 40 years old can be represented the medium usage group.

The total influence representation of this part is shown in figure 10.d

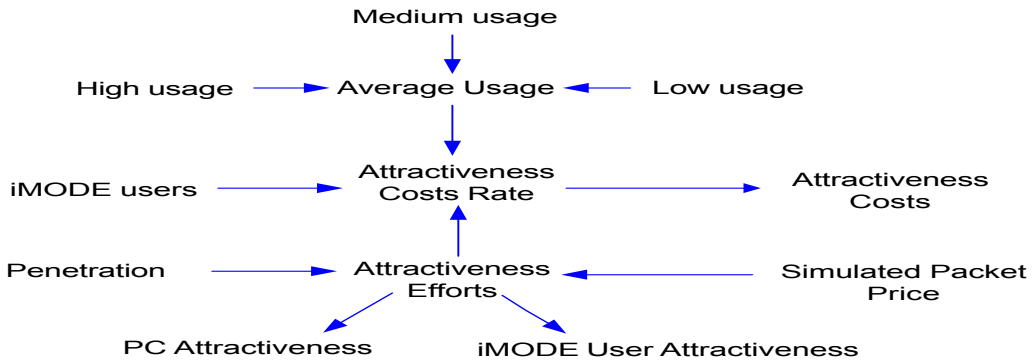


Figure 10.d Influence diagram of Support & Attractiveness

The financial causal loop diagram has been described earlier in this report. Now, it does make some assumption to make the model more realistic and according to what it has been mentioned in the reference model part of this report.

iMODE users can be classified into three groups by their age, and packets usage amount:

- High Usage is relative to those are under 30 years.
- Low Usage is relative to those are under 30-39 years.
- Medium Usage is relative to those who are more than 40 years.

By the assumption that has been made in the causal loop diagram financial part and these leads to sketch the total influence diagram of this segment as shown in figure 10.e.

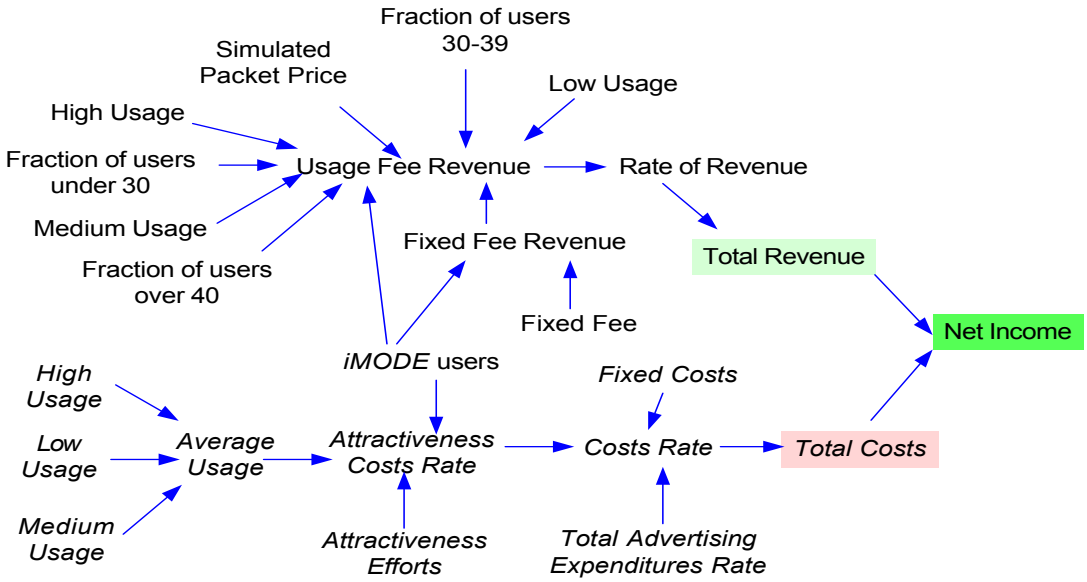


Figure 10.e Financial influence diagram

6. Stock & Flow Diagram (Levels & Rates)

Stock and flow diagram is a graphical notation of the model which consists of three different types of elements such as:

- Stock
- Flows
- Information

Stock and flow diagram shows the relationships among variables which have the potential to change over time, and it uses different type of graphical symbols to distinguish between different types of these variables.

The difference between the Causal Loop Diagram CLD and the Stock and Flow diagram is that a Stock and Flow diagram distinguishes between these different variables, which are distinguished by different graphical symbols.

The work uses the stock and flow diagram to model the *iMODE* market growth as a basis for developing the quantitative model which can be used to study the characteristics of this business process, i.e. the adoption of *iMODE* services growth in Japan.

It will see and find out later in this work the remarkable facts behind the use of stock and flow diagram, and the three mentioned elements which is a general way of graphically representing of any business process.

The first part of the stock and flow diagram represents the potential market growth of *iMODE* services as shown in figure **11.a**

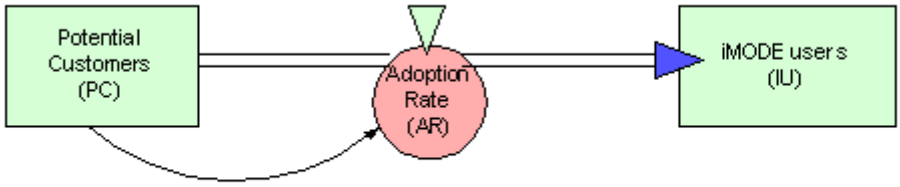


Figure 11.a iMODE Market Growth Stock & Flow Diagram

The graphical notation in figure **11.a** hints the differences between stocks/levels and flows/rates. The rectangles around Potential Customers and *iMODE* users which describe that these variables are stocks/levels, perhaps the butterfly valve around the Adoption Rate variable which describes that Adoption Rate is a flow/rate.

The idea behind the difference between a stock/level and flow/rate is that a stock/level is an accumulation of something such as *iMODE* users in this case, and a flow/rate is the movement of flow from a stock/level to another such as the flow from Potential Customers into *iMODE* users by Adoption Rate in the model.

The variables in this part of the model consist of Potential Customers (PC), *iMODE* users (IU) and Adoption Rate (AR).

The *iMODE users* are the customers have already adopted *iMODE* services. The Potential Customers are the customers who may adopt *iMODE* services.

The variables Potential Customers and *iMODE* users are shown inside rectangles, and this type of variables is called a stock or level.

The other variable, i.e. Adoption Rate variable is shown by "butterfly valve" which is called a flow or rate.

The graphical notation as shown in figure **11.a** hints at the idea that there is a flow from Potential Customers toward *iMODE* users with the rate of the flow controlled by the Adoption Rate variable.

A very interesting issue is changes in variables such as *iMODE* users over time, and the challenge is when the Adoption Rate starts to decline or increase, and to find out why this change occurred and how to address it. Hence, the challenge to sustain a successful business is the ability to address changes before it is too late.

The report will focus on investigating these changes, and how these elements and structure of the *iMODE* market growth model can be bring about such changes.

Some of the possibility to distinguish between stocks and flows is to ask what would occur if it could freeze the time and observe the *iMODE* growth. If it still observes a nonzero value for a quantity, then this quantity is a stock/level. Otherwise is a flow/rate which can only be measured over a period of time, or as a function of time.

6.1. Potential Market Stock & Flow Diagram

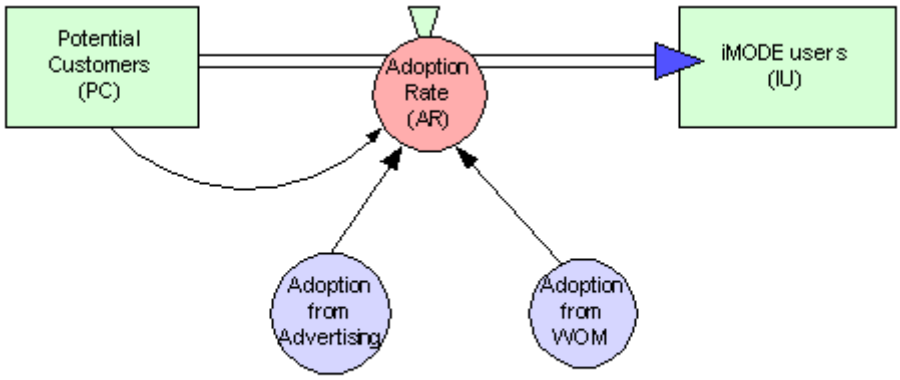


Figure 11.b WOM and Advertising Stock & Flow Diagram

The basic concept behind our *iMODE* growth model is that the potential customers catch the desire to adopt *iMODE* services from those who have already adopted these services.

The Causal Loop Diagram, and as mentioned in the previous part of our research shows the feedback loops which control the adoption rate of *iMODE* services due to both adoption from word of mouth and adoption from advertising.

The graphical notation in figure **11.b** shows this case and the relationships among them.

The total Adoption Rate is the sum of Adoption from the Word of Mouth (WOM), the Adoption from Advertising, and other external influences.

We assume that there are another factors, variables or constants which affect the model behaviour.

First is the assumption that there are a number of people who may adopt *iMODE* services, this is represented by Market Size in our model.

Second, there is another factor which affects the Adoption from Word of Mouth, and this factor we called “Contact Rate” which can be defined as the fraction of the number of people per month might be heard about *iMODE* services from those who have already adopted it.

Third, the fraction of market size that has adopted *iMODE* services which is referred to as Penetration in our stock and flow diagram.

Taking all the above elements into consideration, then the stock and flow diagram will be as shown in figure **11.c.**

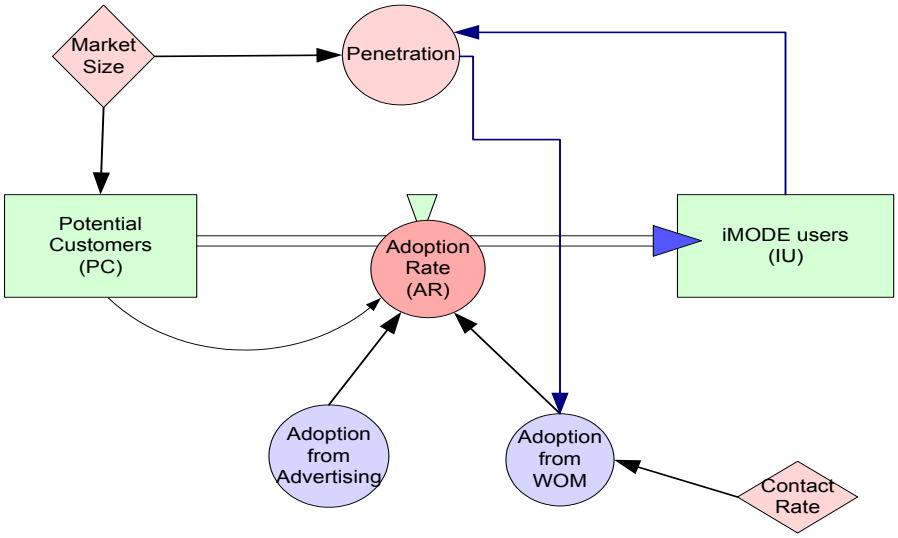


Figure 11.c WOM and Advertising Stock & Flow Diagram

To understand the relationships and the meaning of all the lines in the stock and flow diagram above, the report uses the well known mathematical expressions.

It begins with Adoption Rate AR, which can be expressed as

*AR = (Potential Customers * Adoption from advertising) + Adoption from WOM*

Where the,

Adoption from WOM = *(1 – Penetration) * iMODE users * Contact Rate*

Hence the Penetration which is the fraction of market size that has adopted *iMODE* services can be given by the following function.

$$Penetration = \frac{iMODE\ users}{Market\ Size}$$

Another segment of the model represents the *iMODE* users break part as shown in figure **11.d.**

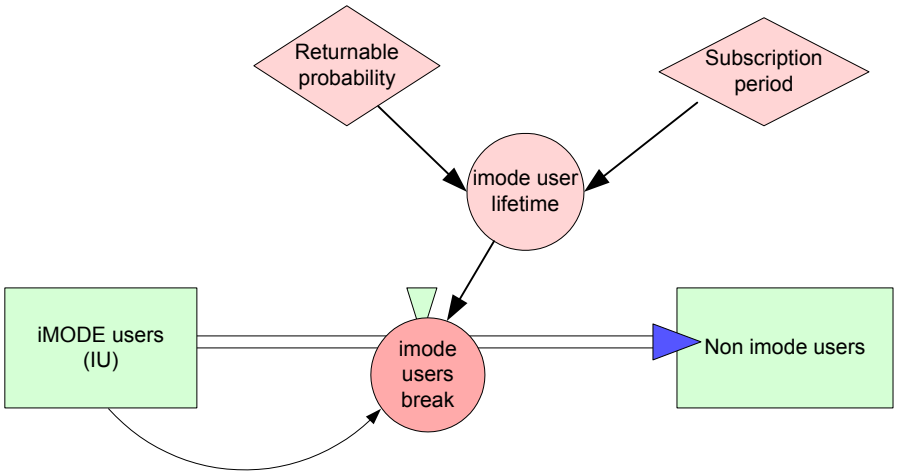


Figure 11.d iMODE users break Stock & Flow Diagram

The variables and constants that could be seen in this part are:

- *iMODE* users' break is the rate at which an *iMODE* user becomes a non *iMODE* user. This is driven by a new telecommunication service that may emerge and launch into the market. Then *iMODE* user may be persuaded to cease availing of *iMODE* and abandons the service.
- Subscription period is the duration of time which an *iMODE* user signs up for using *iMODE* services.
- Returnable probability is the factor of a non *iMODE* user who might return to use *iMODE* services, i.e. a non *iMODE* user being an *iMODE* user again.
- *iMODE* user lifetime is the average number of years, maybe months, customer subscribes to the *iMODE* services.

The mathematical expression which illustrates the relationship among *iMODE* user lifetime, Subscription period and Returnable probability can be given by the following function

$$\textit{imode user lifetime} = \frac{\textit{Subscription period}}{\textit{(1 – Returnable probability)}}$$

Hence, another relationship connects *iMODE* user break *iMODE* user lifetime and *iMODE* users which can be given by the following function

$$\textit{imode user break} = \frac{\textit{iMODE users}}{\textit{imode user lifetime}}$$

6.2. Advertising Strategy Stock & Flow Diagram

Advertising strategy is a balancing negative feedback loop which is addressed by the adoption rate due to advertising in the model.

Adoption Rate reduces the number of Potential Customers. And the reduction in Potential Customers number affects a reduction in the Adoption Rate which is given by the constant named Advertising Expenditures Effect.

Advertising Expenditures Effect depends on many other variables, and these variables had been mentioned earlier in this report, in section 5.3.

All these variables are amalgamated to sketch the stock and flow diagram for such strategy. This is shown by the diagram in figure **11.e**

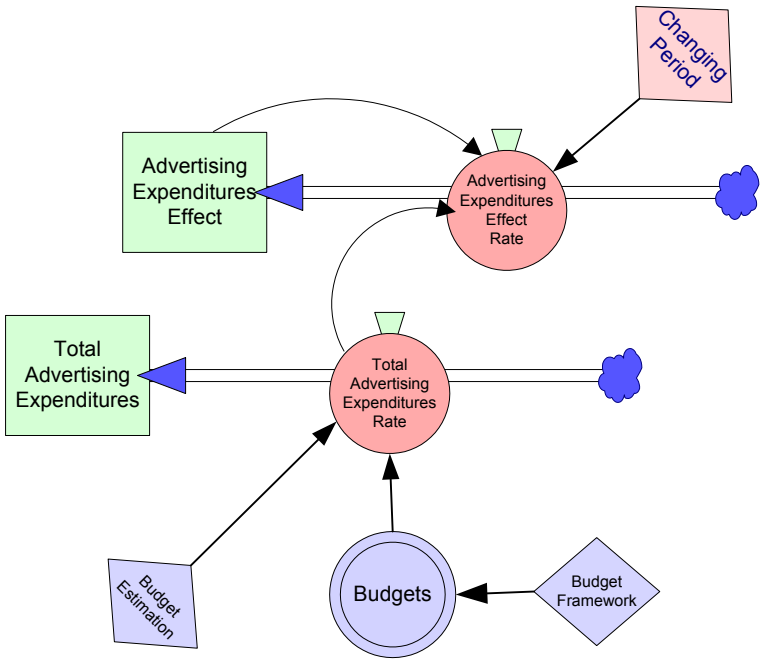


Figure 11.e Advertising Strategy Stock & Flow Diagram

6.3. Pricing strategy Stock & Flow Diagram

Pricing strategy is a positive feedback loop which is addressed by the pricing rate due price control.

Pricing Rate reduces the amount of price control. And the reduction in price control amount affects a reduction in the Pricing Rate which is given by the constant named Simulated Packet Price.

Simulated Packet Price depends on many other variables, and these variables had been mentioned earlier in this report, in section 5.3.

The work puts all these variables together to sketch the stock and flow diagram for such strategy. This is shown by the diagram in figure **11.f**

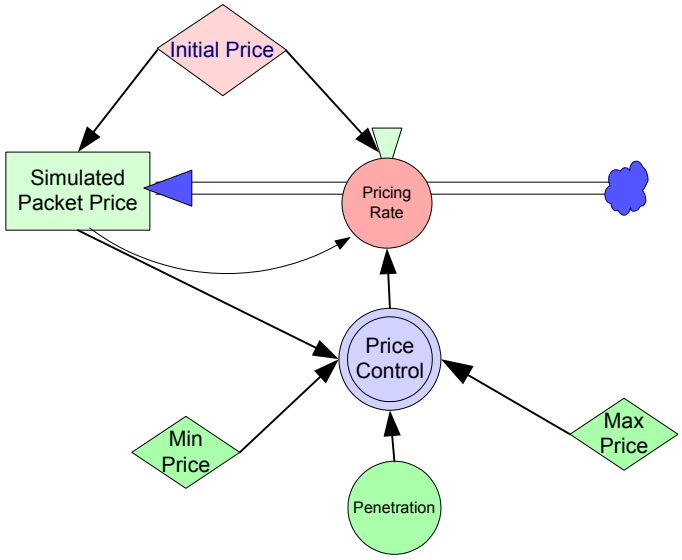


Figure 11.f Pricing Strategy Stock & Flow Diagram

6.4. Support and Attractiveness (QoS) Stock & Flow Diagram

Support and Attractiveness stock and flow diagram demonstrates a positive feedback loop which is addressed by policies we have been made in the causal loop diagram of this part, and according to two types of attractiveness.

There is a pool between Attractiveness Costs Rate into Attractiveness Costs, which is the costs to attract the current *iMODE* users or the potential customers. However, *iMODE* users are those who have adopted *iMODE* services, and Potential Customers are those who have not adopted the services yet.

The assumption concerning this part is that the price, i.e. Simulated Packet Price, influences both new users and current *iMODE* users, and there are a set of features appealing to those users.

The stock and flow diagram of this segment is shown in figure **11.g.**

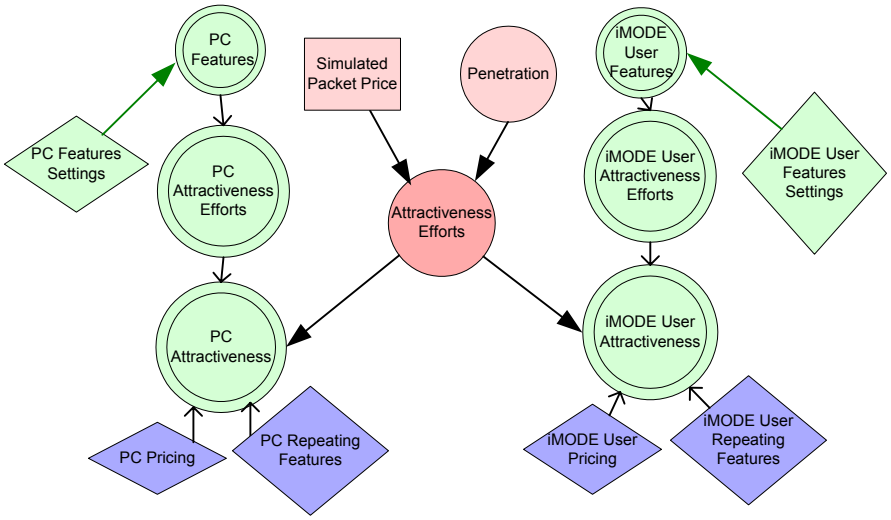


Figure 11.g Support and Attractiveness Stock & Flow Diagram

6.5. Financial Segment Stock & Flow Diagram

The report has been describes the causal loop diagrams, and the influence diagrams with the assumptions and policies that have been made earlier in this report of the financial part.

Revenues, Costs and Net Income are those who are the most important variables in this segment.

From the causal loop diagram which gives a good background to sketch stock and flow which demonstrates all the pools and how these variables connects and interacts each other on the whole process.

The report will describe this part more specific in the next chapters.

7. Modelling by Powersim Studio

After it has finished explaining and assigning the influence diagrams concerning *iMODE* services, and after it has analysed a reference models that have a clear view of the *iMODE* services growth case, and it has confidence in it. It is time to build the conceptual model using *PowerSim Studio*, which is based on that influence diagrams, and the report decided the variable names to be clear and easy to understand. It might be feeling that the variable names represent all segments of the model clearly.

iMODE services model had been implemented in *PowerSim Studio* and the report have been classified this model into many segments in order to make it clear and easy to be understood. Those different segments of this model are related to each other and interact with each other.

Those segments are:

- The Migration of *iMODE* users.
- Advertising Expenditures Effect.
- Pricing Strategy.
- Support customers and Attractiveness Efforts.
- Revenue Stream and segmentation of *iMODE* users by their age.
- Total Costs (*JPY*).
- Net Income & Income (JPY).
- Revenue, Costs and Income in Norwegian Krone (*NOK*).
- Revenue, Costs and Income in European Currency (*EURO*).

The report is explaining and analysing these segments in details.

7.1 The Migration of *iMODE* users.

This is the first part of the model, and it is the core in the model. This segment represents and illustrates the potential market growth of *iMODE* services and how the migration of *iMODE* users occurs depending on word of mouth and advertising effect.

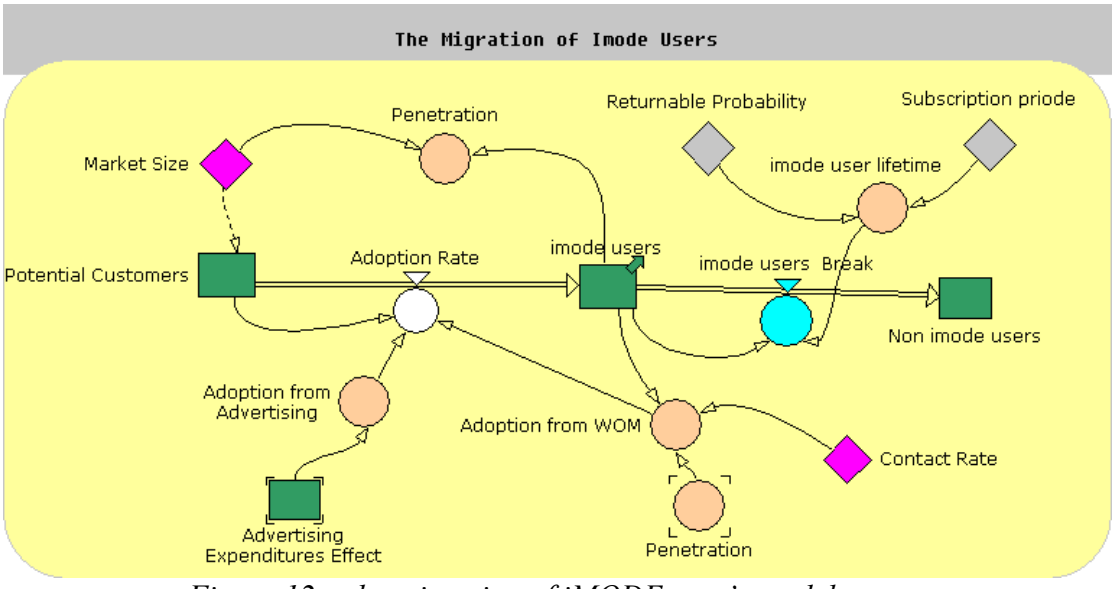


Figure 12.a the migration of iMODE user's model.

Some assumptions it has been made for *iMODE* Growth Model:

- The market size of *iMODE* services adoption is 40-60 million customers. This assumption is based on many reasons. It does know that the total numbers of population in Japan are approx. 115 million. Each population can be divided into many categories, on their age's demography or on their attitudes. By subtracting the number of people that they are children and very old that might not be adopting *iMODE* services. The work might assume that the total market size of *iMODE* services is approximately half of the total number of population.
- Each *iMODE* user tells two other potential customers about *iMODE* services each month, and this corresponds to 2 %(2CUST/CUST/MONTH).
- Subscription period of an iMODE user to adopt services is 24 months.
- Returnable probability is 75% that an iMODE user turns back to adopt services.

This part of our model illustrated in figure **12.a**.

7.2 Advertising Expenditures Effect.

The interesting features that have used to create this segment of the model can be summarised in the list below.

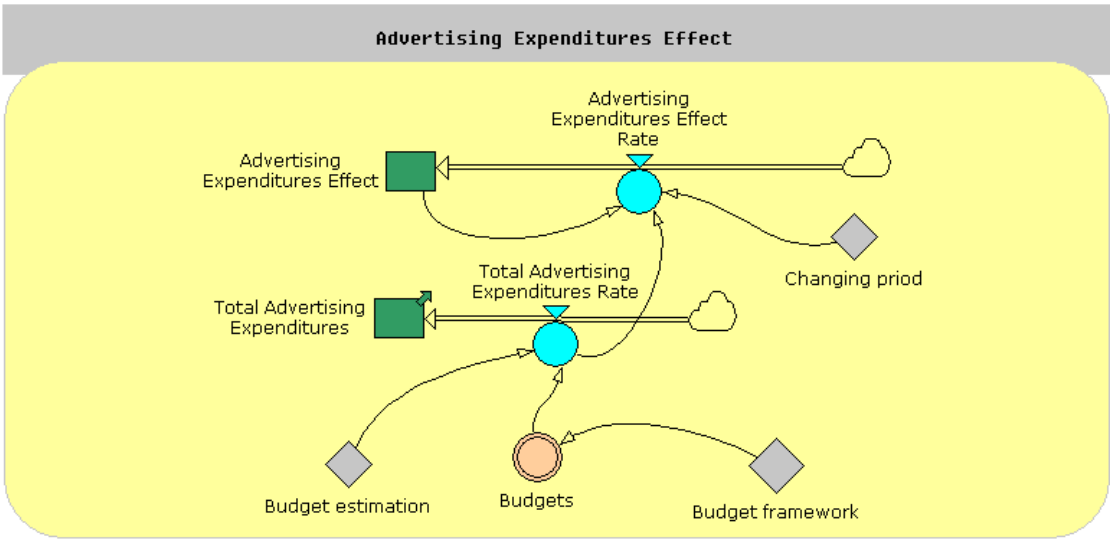


Figure 12.b advertising Expenditures Effect segment.

- The strategy that has chosen in this part of the model is the structure that allows a select between manual and automatic budget.
- The variable "Budgets" is an array contains two elements.
- There is a delay between Total Advertising Expenditures Rate and Advertising Expenditures Effect.
- The Changing Period of the Advertising Effect is one month.

All these variables and all the strategies have been made concerning this part of the model is described in figure 12.b

7.3 Pricing Strategy segment.

This part of the model demonstrates the pricing strategy that *NTTDoCoMo* might adjust their packet price on penetration which is the fraction of the Market Size that has adopted imode services.

There is a pool from pricing rate into simulated packet price which depends on many other factors such as:

- Initial Price: the initial packet price.
- Price Change Delay: is the period of time that *NTTDoCoMo* might changes their packet price.
- Price Control: an array which depends on simulated packet price, minimum packet price, maximum packet price and penetration.

The differential equation of Price Control can be shown as following:

```
Price Control = AVERAGE {Simulated Packet Price, Max Price,
                                    Min Price *(1-Penetration)}
```

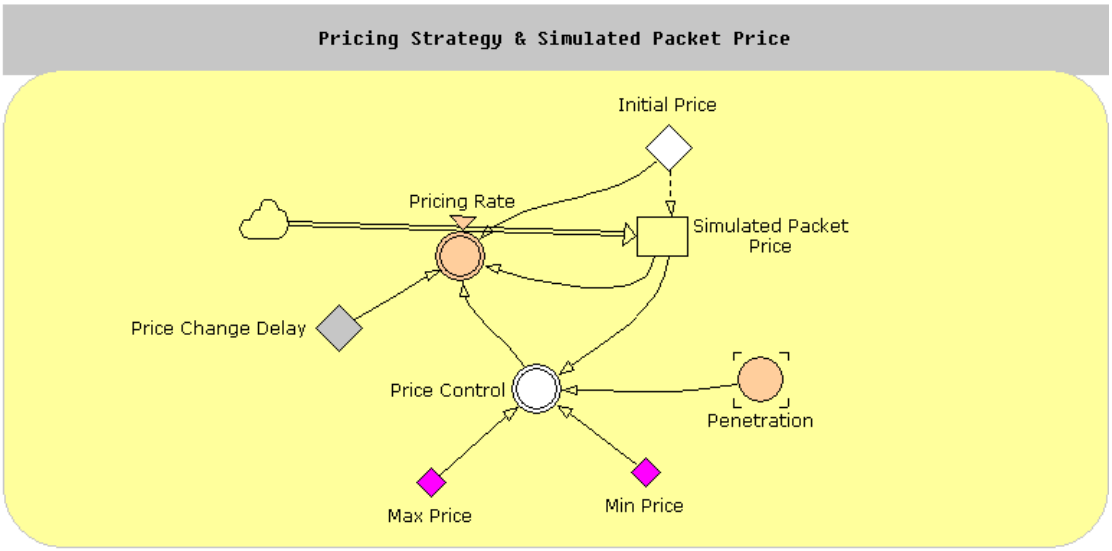


Figure 12.c Pricing Strategy segment

There is also a positive feedback loop between simulated packet price and pricing rate. This segment of the model illustrated in figure **12.c**.

7.4 Support customers & Attractiveness efforts.

This part of the model demonstrates the support and attractiveness efforts of serving both *iMODE* users and Potential Customers.
Hence, *iMODE* users are those who have adopted *iMODE* services, but Potential Customers are those who have not yet adopted *iMODE* services.

The report assumes that the simulated packet price and penetration influences both these users. As well as to this assumption, it has a choice of features settings between

appealing either Potential Customer or repeating an *iMODE* user. Then, the work uses a geometric mean to calculate the attractiveness efforts.

This segment illustrates is shown in figure **12.d.**
Some of the main differential expressions of this segment can be illustrate as following:

*iMODE user Attractiveness = {iMODE User Attractiveness Efforts * iMODE User Repeating Features * Attractiveness Efforts*iMODE User Pricing}* {1 / (iMODE User Repeating Features + iMODE User Pricing)}*

PC Attractiveness = {PC Attractiveness Efforts'PC Repeating Features * Attractiveness Efforts * PC Pricing} * {1 / (PC Repeating Features + PC Pricing)}*

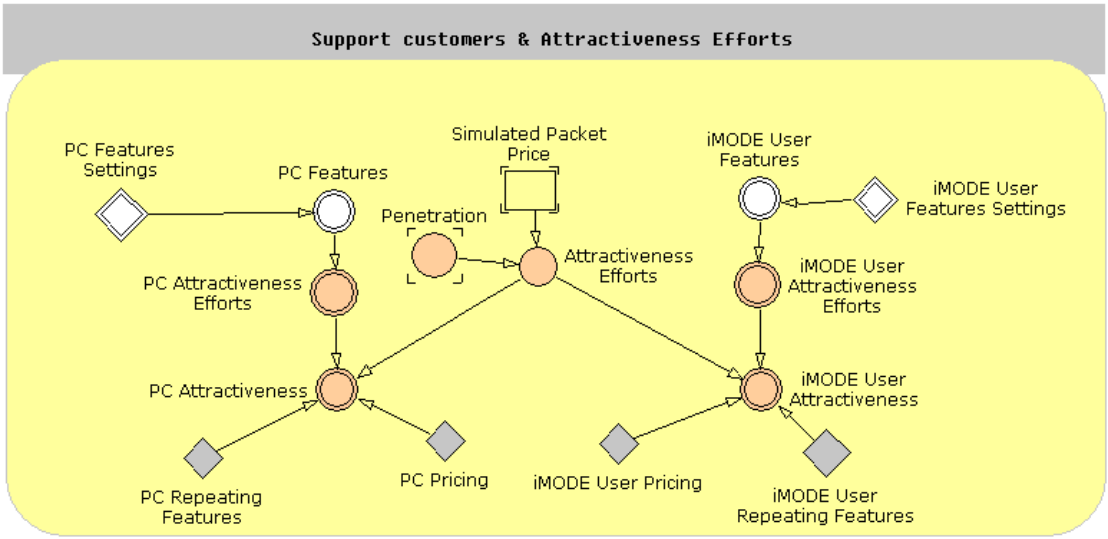


Figure 12.d Support Customers & Attractiveness Efforts

Both iMODE User Attractiveness and PC Attractiveness, Potential Customer attractiveness, are arrays.

Dimensions define the structure and number of elements in the array, and can be either sub ranges (numerical or enumeration) or named ranges (enumeration ranges, numerical or enumeration sub ranges).

These arrays have two elements, and the differential expression of both *iMODE* User Attractiveness and PC Attractiveness can be addressed as:

*iMODE User Attractiveness = {iMODE User Attractiveness Efforts * iMODE User Repeating Features * Attractiveness Efforts * iMODE User Pricing} *{1/ [iMODE User Repeating Features + iMODE User Pricing]}*

*PC Attractiveness = {PC Attractiveness Efforts * PC Repeating Features * Attractiveness Efforts * PC Pricing} *{1/ [PC Repeating Features + PC Pricing]}*

All other variables that could seen in this segment of the model are also arrays of two elements, these variables includes *iMODE* User Attractiveness Efforts, *iMODE* User Features and *iMODE* User Features Settings, the same definition and expressions can be represented for the Potential Customer part.

The last variable which more interesting here is the Attractiveness Efforts can be expressed differentially as following:
Attractiveness Efforts = Simulated Packet Price(1-Penetration)*

7.5 Revenue Stream and segmentation of *iMODE* users by their age

This part of the model demonstrates the stream of revenues occurred when *iMODE* users start pay to *NTTDoCoMo* for the *iMODE* services they adopt.

The underlying concept that has been adhered to has been illustrated in the causal loop diagram concerning this segment of this model. The revenues have been classified depending on the segmentation of *iMODE* users by their age and usage amount of packets. Additionally, there is a fixed fee which must be paid by users to access *iMODE* services. This segment is shown in figure **12.e**

Some of the differential expression for this part of the model can be given by the following equations:

*Fixed Fee Revenue = imode users * Fixed Fee*

Where, Fixed Fee is a monthly fee an imode user must pay to access services.

*Usage Fee Revenue = (imode users under 30 * High Usage * Simulated Packet Price)*
*+ (imode users over 40 * Medium Usage * Simulated Packet Price)*
*+ (imode users 30-39 * Low Usage * Simulated Packet Price)*

Where, High Usage are the number of packets consumes by imode users who are under 30 years old, and Low Usage are the number of packets consumes by *iMODE* users who are between 30 and 39 years old. Finally, Medium Usage is the number of packets consumes by those who are over 40 years old. While simulated packet price is the price of one packet, where one packet is equal to 128 byte.
There are connections between those two revenues stream into Rate of Revenue, and there is a pool into Total Revenues.

Rate of Revenue = Usage Fee Revenue + Fixed Fee Revenue

The simulation model integrate the differential equation over time to calculate the total revenues every date during period of time interval.

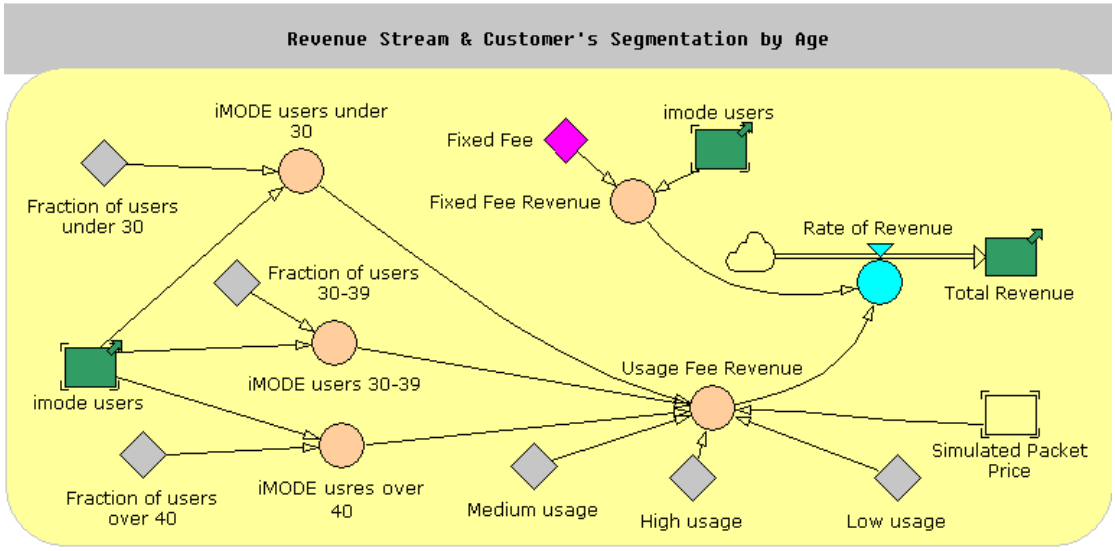


Figure 12.e Support Customers & Attractiveness Efforts

7.6 Total Costs (*JPY*)

This part of the model demonstrates the total costs NTTDoCoMo might used to support their customers, and maintain their network devices.
There are many variables that have been taken into consideration such as:

- Attractiveness Costs Rate.
- Fixed Costs.
- Advertising Expenditures Costs.

The total costs, and straight ahead is equal to the accumulation of those variables as shown in figure 12.f.

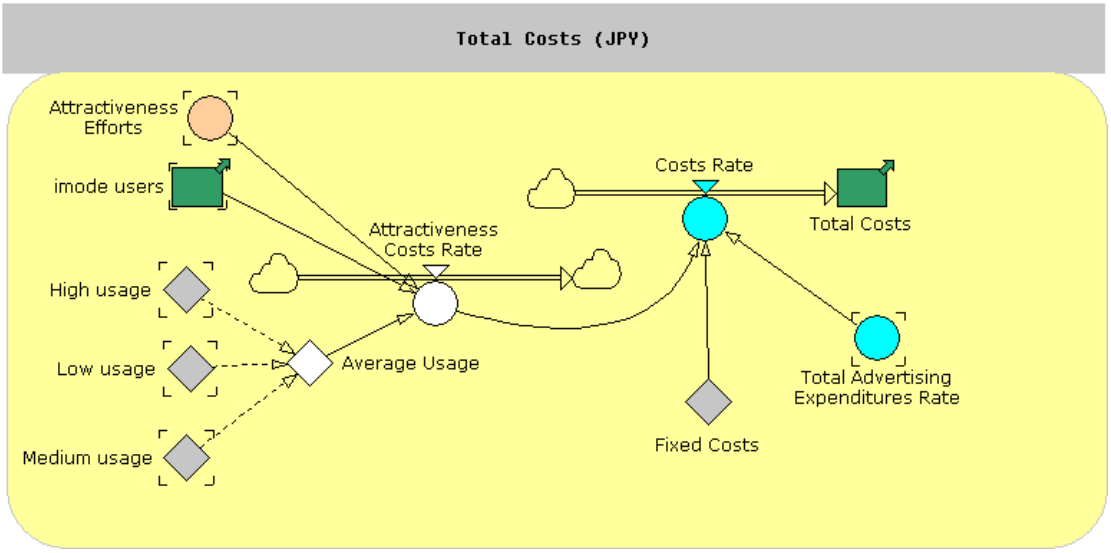


Figure 12.f Support Customers & Attractiveness Efforts

Attractiveness Costs Rate depends on the strategy that has been assumed. This strategy establishes that the Attractiveness Costs Rate depends on the total *iMODE* users and the amount of data traffic, i.e. the amount of usage which is illustrates by three categories as High, Low and Medium usage. These categories connected to a

variable which is called Average Usage. And Average Usage connected directly to the Attractiveness Costs Rate

7.7 Net Income & Income (JPY).

This segment of the model assumed that there is amount of money that *NTTDoCoMo* might paid to the government, and this is represented by the variable called Tax Fraction as it shown in figure 12.g.
The mathematical forms for this part can be expressed as following

Net Income = Total Revenue – Total Costs
*Income = Net Income – (Net Income * Tax Fraction)*

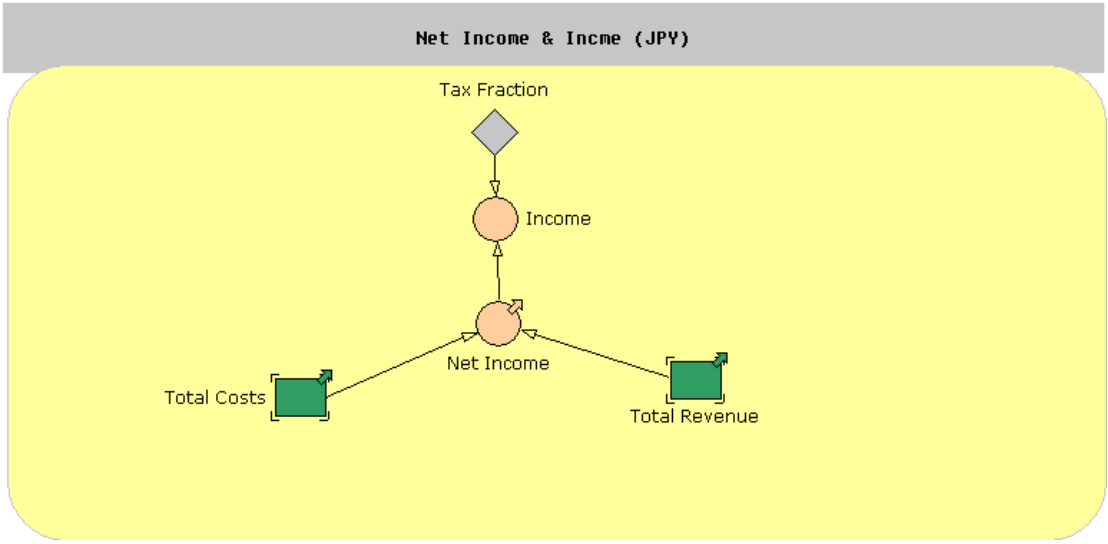


Figure 12.g Support Customers & Attractiveness Efforts

7.8 Revenue, Costs and Income in (*NOK*) & (EURO).

The next two figures illustrate the Revenue, Costs and Income converted from the Japanese yen to Norwegian Krone and European Currency relatively.

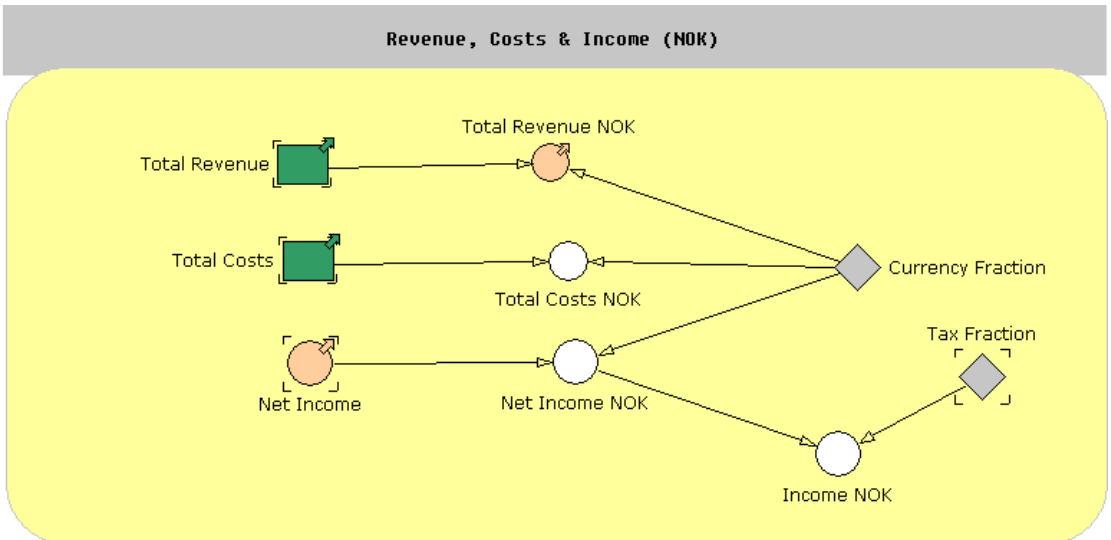


Figure 12.h Support Customers & Attractiveness Efforts

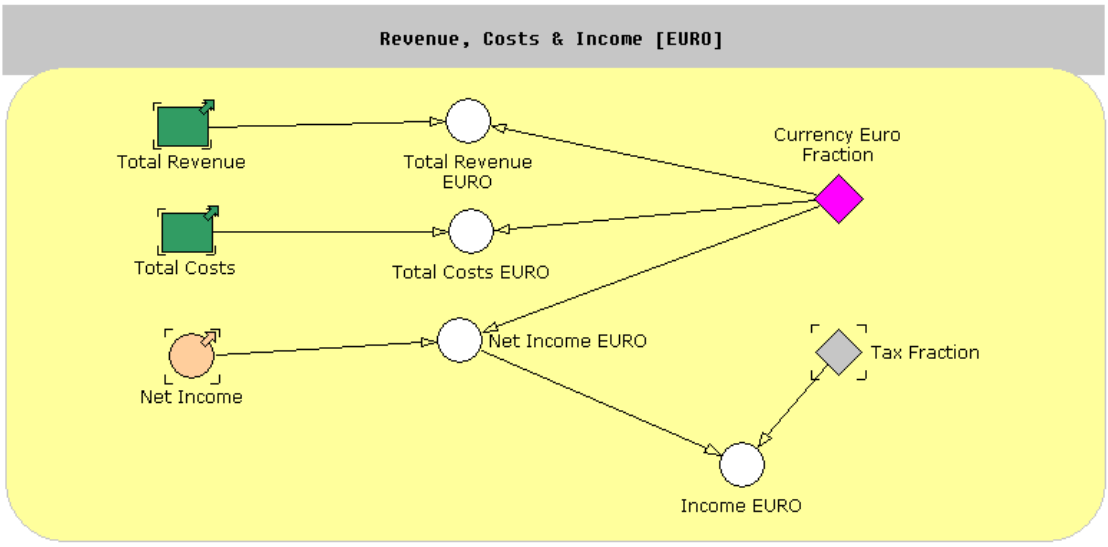


Figure 12.i Support Customers & Attractiveness Efforts

7.9 Simulation the model

It is time for the simulation stage of the system dynamics modelling. After the model has been built with *PowerSim Studio* as computational tools and this model has been put into the computer. As well as all the variables and equations are well defined, the work starts to simulate the model to explore the results and view the behaviour over time.

The benefits of the simulation of the model are that it shall realise whether the actual behaviour differs from the expectations.

The simulation process has been starting by setting up appropriate simulation settings for the model. The two most important are the *time horizon* and *time step.*

The *time horizon* is the time interval which wants the model to simulate. In this case, it has chosen that the time interval is from February, 1999 until the end of year 2006.

While, the *time step* which is the second setting variable is chosen by the computational tools that has been used, i.e. *PowerSim Studio*.

After determining the *time horizon* and the *time step* of simulation of the model. The work has the ability to simulate the model under different conditions, and observed the results by graphical curves which have been showing the behaviour of the growth market of the *iMODE* services. [*See appendix C!*]

7.10 Presentation Mode in PowerSim Studio

PowerSim Studio has many features to run the model. The design mode and the presentation mode are some of these features. The design mode allows user to edit and create the model, controls, objects, etc, and this has been achieved by converting all the causal loop diagrams and stock and flow diagrams into a conceptual model.
The presentation mode is like a browser, and a graphical user interface, where the user can run, simulates the model, assign input values and study the results.

We have been built a presentation mode as shown in figure **12.j**

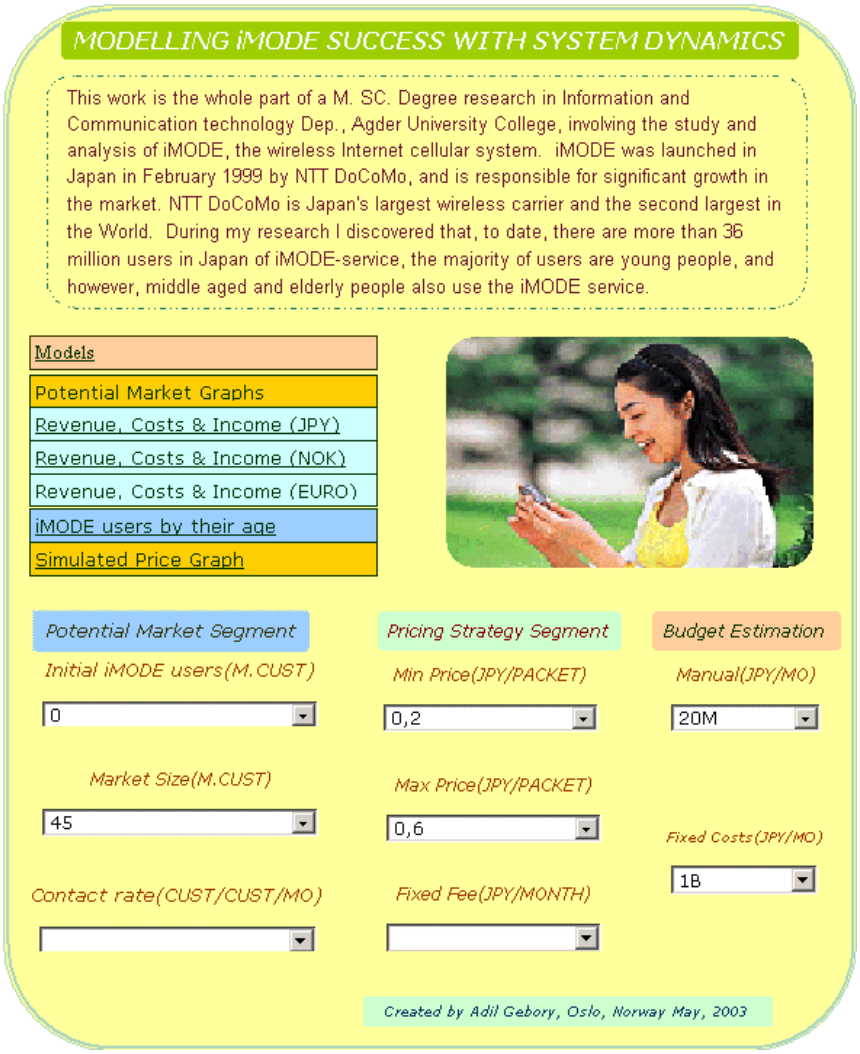


Figure 12.j Support Customers & Attractiveness Efforts

The user has the ability to choose a several input values for the model, and simulate the model under these choices.

The combo boxes give the user capability to assign the following input data for the model before the simulation process

- Initial imode users {0, 1, 5, 10, 15, 20, 25} Millions customers.
- Total Market size {40, 45, 50, 55, 60} Millions customers.
- Contact Rate concerning *word of mouth* {1, 1.5, 3} Customer per Customer per Month.
- Minimum Packet price (Min Price) {0.1, 0.2, 0.3} JPY per Packet.
- Maximum Packet price (Max Price) {0.4, 0.5, 0.6} JPY per Packet.
- Fixed Fee {100, 200, 300} JPY per Customer per Month.
- Budgets {20, 40, 60} Millions JPY.
- Fixed Costs {1, 5, 10} Billions JPY.

There are several links to different stages that user can browse to another parts or segments of the model.

When all the variables which are represented by combo boxes in the graphical user interface have been registered, then the simulation process can start and many graphs may be created which explore the relationship between some variables with time. [*See appendix C!]*

The units that had been used in the model and clearly obvious from the presentation mode are:

- "JPY" is the Japanese currency.
- "NOK" is the Norwegian Krone.
- "EURO" is the European currency.
- "CUST" is the unit that defines the number of subscribers of imode services.
- "PACKET" is the unit that defined the amount of usage of data traffic.

8. Graphical User Interface (GUI) by JavaSwing

This part of the work is an application built with *Java Swing* programming language. The work had tried to make the graphical user interface which had been explored in the previous chapter of this report into a modern shape. A prototype had been built which explore a *Java Swing Applet* that can be fetching on browser like the most known one called *Internet Explorer*.
Java Swing Applet has almost the same capabilities as the presentation mode in *PowerSim Studio* and is shown in figure 13.

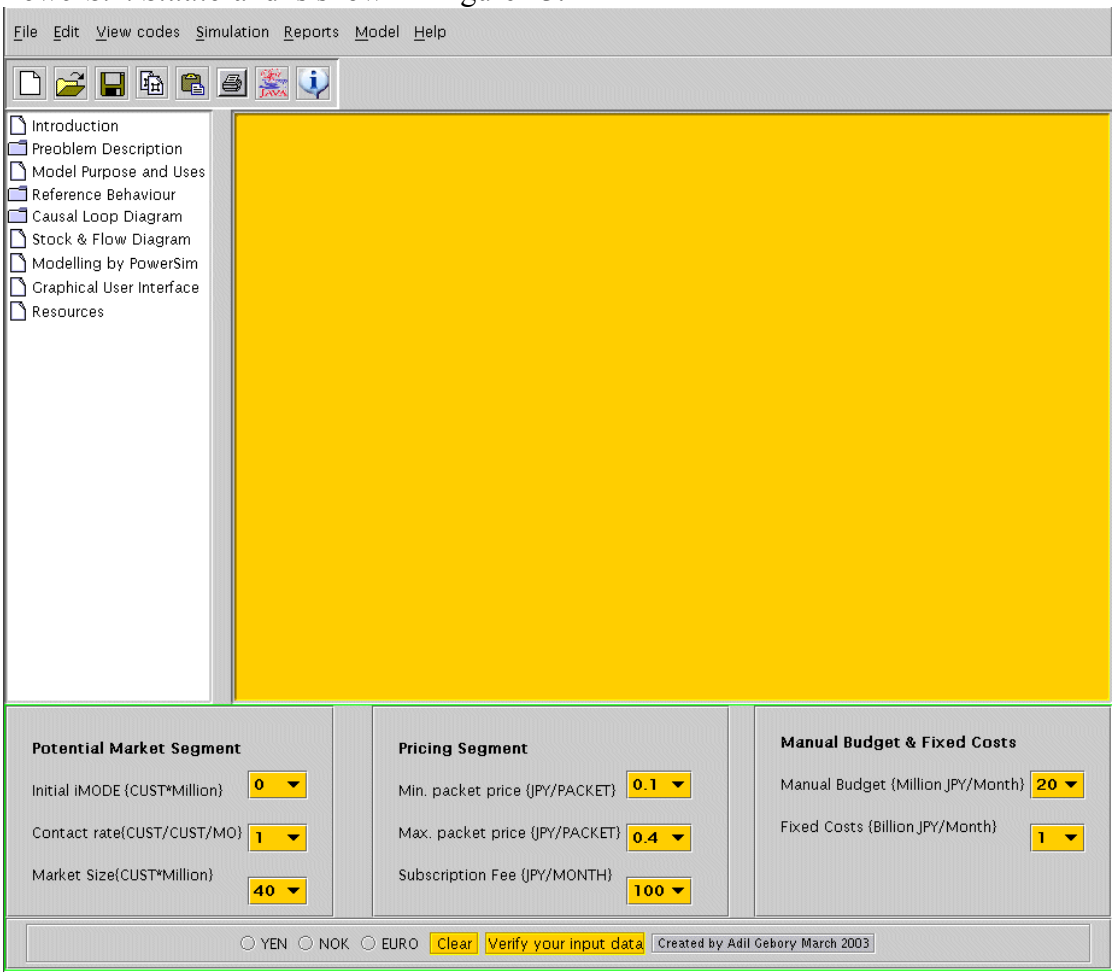


Figure 13 the graphical user interface as a *Java Swing Applet*

There are more features the user can do with this *applet*. Such as:

- The normal file menu with new, open, save and exit.
- The edit menu with sub choices like undo, copy, cut and paste.
- The view codes menu with some sub choices to view the java codes back this applet.
- Simulation menu with sub choices like run, forward, backward and pause.
- Reports menu with sub choices which give the user ability to view tables explore the number of *IMODE* users in term of time, and revenue reports.
- Model menu with sub choices to explore curves and segments of the model.
- Help menu with sub choice shows the creator and date that applet had been built.

There are no dynamic connections between model in *PowerSim Studio* and the *Java Swing Applet* because of the time limited and the needs to many other computational tools that could not be provided.

Some of these computational tools are:

- Metro Server.
- PowerSim Exporter.
- Java Metro API packages.

System requirements:

To run all features in this project, the Personal Computer must have the following software:

- MS Win32 (Windows 98, ME, NT or higher version).
- Microsoft Internet Explorer version 5.5 or higher.
- Java Plug-in 1.4(get it free from http://java.sun.com).
- PowerSim Studio 2001.
- Microsoft Word 2000 or higher.
- Adobe Acrobat Reader 4.0 or higher
  (Get it free from http://www.adobe.com).

9. Results

The model may provide a new basis for understanding reality and acting on that understanding. It has been demonstrated how the market growth occurs when the model has been simulated, and the graphical curves that were produced showed the occurrence of a dynamic process of the whole work.

Simulation showed the significant growth of the total number of *iMODE* subscribers as an exponential function versus time, and the accumulated revenue from the amount of usage which was obvious from the financial segment of the model.

While, after a period of time applied the exponential growth could not continue, i.e. there was a peak value where the graph started to decrease in terms of the decrease in iMODE subscribers for various reasons as is shown in figure 14.a.

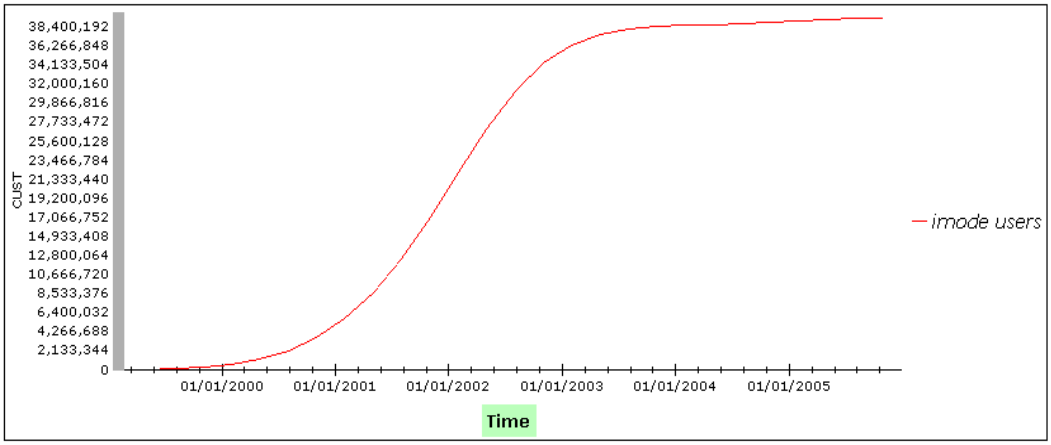


Figure 14.a Total number of iMODE subscribers versus time

The accumulated revenue from the amount of usage which has been mentioned above can be demonstrated as in figure 14.b.

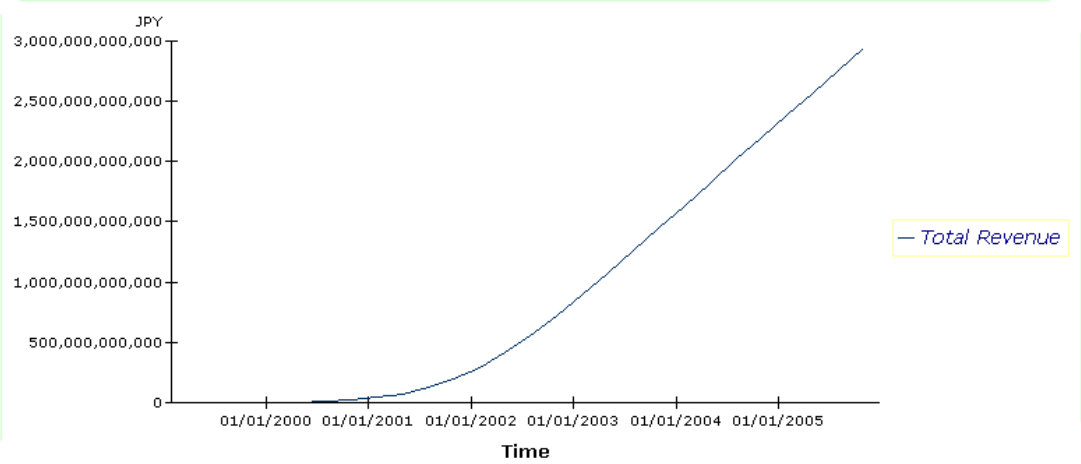


Figure 14.b Total Revenue from amount of usage versus time

10. Conclusions

This report discussed the behaviour of a system which explores the growth of *iMODE* services in Japan by use of *System Dynamics* approach as time passes and new decisions as a significant type of management problem, and it have been tackled the issues of how this system reacts to dynamic forces and how those reactions shape the behaviour as it moves into the future.

The work has started from a number of written hypothesis, policies and developed a model which represents the problem, i.e. the market growth of *iMODE* services in Japan.

It has explored the problem by sketching the causal loop diagrams, and created stock and flow diagrams which have been helping to get the model to be more obvious and easy to build it by using *PowerSim Studio.*

The model has helped to explore some of these hypothesis and policies that have been made. Some of the problems in implementing model results are explaining those results in terms that the people implementing them can relate to.

A simple conceptual model has been built by using *PowerSim Studio* to help it thinks about different issues around the market growth of *iMODE* services in Japan with the features of *System Dynamics*. Many assumptions have been assigned to make the model being near the actual state.

It has been fortunate that the model produces the results which are close aligned to reality and close to the results envisaged prior to the models development and operation. Had the model failed to produce satisfactorily accurate results, the developer would have been compelled to reassess alternatives and chose the one which most reflected the real world scenario.

An important aspect of a good project is to feed results back as they arise. One of the attributes of *System Dynamics* techniques is the rapid initial result, whether from the models or diagrams.

It thinks that the work have developed a solid knowledge and understanding of how *System Dynamics* can be used to explore and solve intricate problems.

11. Resources

11.1. References:

- R. Geoff Coyle: System Dynamics Modelling - A Practical Approach, Chapman & Hall, London, Glasgow. 1996. ISBN 0-412-61710-2. 413 pages.

- Jose J. Gonzalez: Lectures in Agder College, Dep. Of Communication & Information Technology, 2000. [http://fag.grm.hia.no/ikt1200]

- Maier, F. H.: New Product Diffusion Models in Innovation Management- A System Dynamics Perspective. System Dynamics Review Vol. 14, No. 4, (Winter 1998): 285-308

11.2. Web sites:

- The UMTS Forum [http://www.umts-forum.org/]
- Eurotechnology [http://www.eurotechnology.com/iMODE/faq-gen.html]
- *NTT DoCoMo*, 2001 [http://www.nttdocomo.com/new/contents/01/whatnew0305.html]
- Arthur D. Little [http://www.arthurdlittle.com/]
- Strategy Dynamics web site [http://www.strategydynamics.com/]
- NTT DoCoMo, INC [http://www.nttdocomo.co.jp/english/
- Sun Microsystems [http://java.sun.com]
- Java Community web site [http://community.borland.com/java/]

11.3. Applications:

- Microsoft Windows XP home edition
- Powersim Studio Student Express 2001
- Powersim Solver 2.1
- Microsoft Word 2002
- Microsoft Excel 2002
- Microsoft PowerPoint 2002
- Adobe Acrobat version 5.0
- Adobe PhotoShop version 7.0
- Microsoft FrontPage 2002
- Microsoft Visio Professional
- Paint Shop Pro 7.1
- Java Development Kit (JDK 1.4.1)

12. Appendixes

A. The model printed from *PowerSim Studio*.
B. Equations printed directly from *PowerSim Studio*.
C. Graphical curves.
D. Java Codes.
E. HTML codes.

13. What's on the *CD-ROM?*

The compact disc is an auto runs compact disc starts the "index.html" by use of the Internet Explorer as a browser.

"BROWSERCALL.EXE" is an executable program which gives the ability to run html file from compact disc.
"autorun.inf" had been coded to call the "BROWSERCALL.EXE" and start index.html by use of Internet Explorer.

The main page has many links which it able to switch to model in *PowerSim Studio*, report as word document, applet in java swing, report as portable document format, and some other links such web site of the Agder University College and the web site of the Information and Communication department in Agder University College.

The contents of the compact disc are illustrated in figure 15, but the homepage of the web site is illustrated in figure 16.

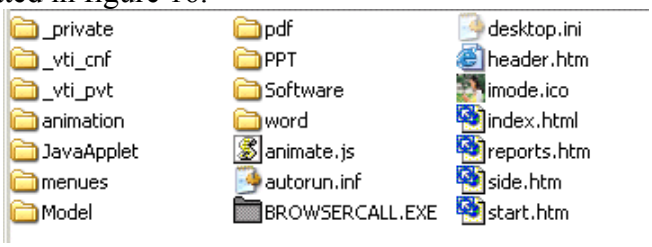


Figure 15 Compact disc contents

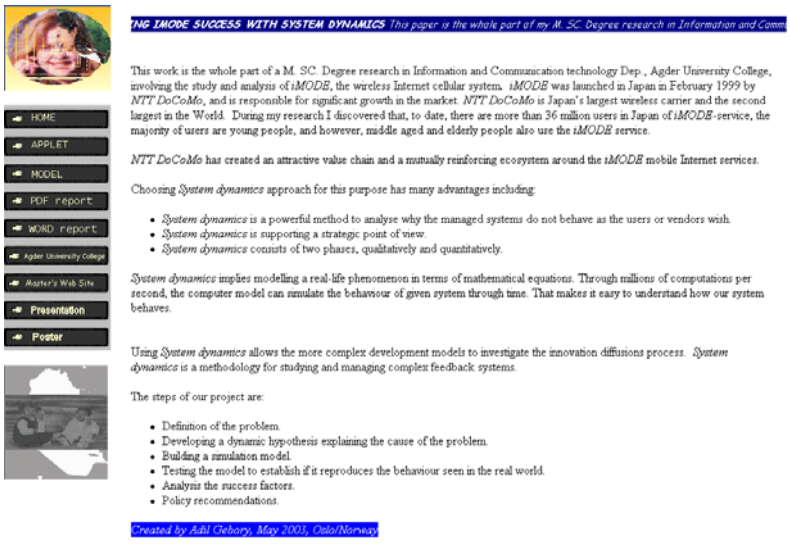


Figure 16 Home page of the web site

APPENDIX A

MODELLING iMODE SUCCESS WITH SYSTEM DYNAMICS

This work is the whole part of a M. SC. Degree research in Information and Communication technology Dep., Agder University College, involving the study and analysis of iMODE, the wireless Internet cellular system. iMODE was launched in Japan in February 1999 by NTT DoCoMo, and is responsible for significant growth ir the market. NTT DoCoMo is Japan's largest wireless carrier and the second largest i the World. During my research I discovered that, to date, there are more than 36 million users in Japan of iMODE-service, the majority of users are young people, anc however, middle aged and elderly people also use the iMODE service.

Models

- Potential Market Graphs
- Revenue, Costs & Income (JPY)
- Revenue, Costs & Income (NOK)
- Revenue, Costs & Income (EURO)
- iMODE users by their age
- Simulated Price Graph

Potential Market Segment

Initial iMODE users(M.CUST

Market Size(M.CUST

60

Contact rate(CUST/CUST/MO)

1,5

Pricing Strategy Segment

Min Price(JPY/PACKET

0,2

Max Price(JPY/PACKET

0,6

Fixed Fee(JPY/MONTH

300

Budget Estimation

Manual(JPY/MO

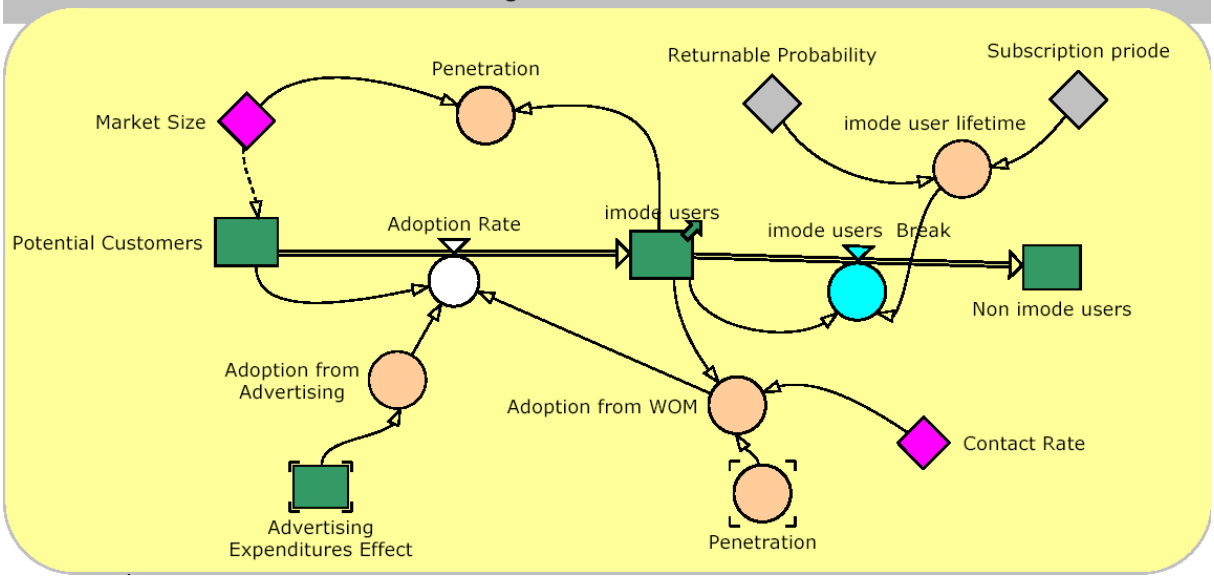
50M

Fixed Costs(JPY/MO

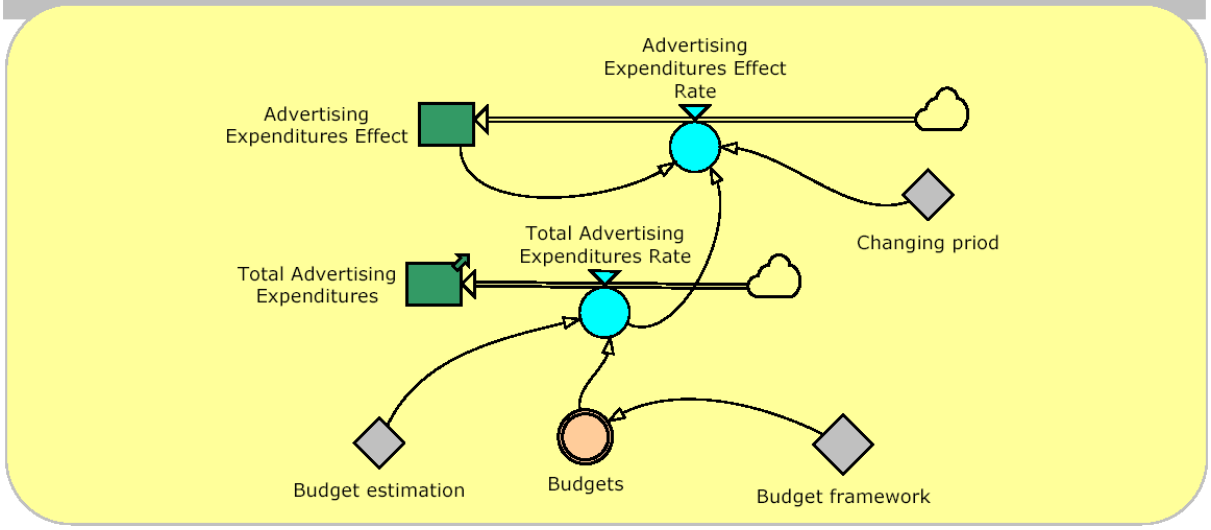
1B

Created by Adil Gebory, Oslo, Norway May, 2003

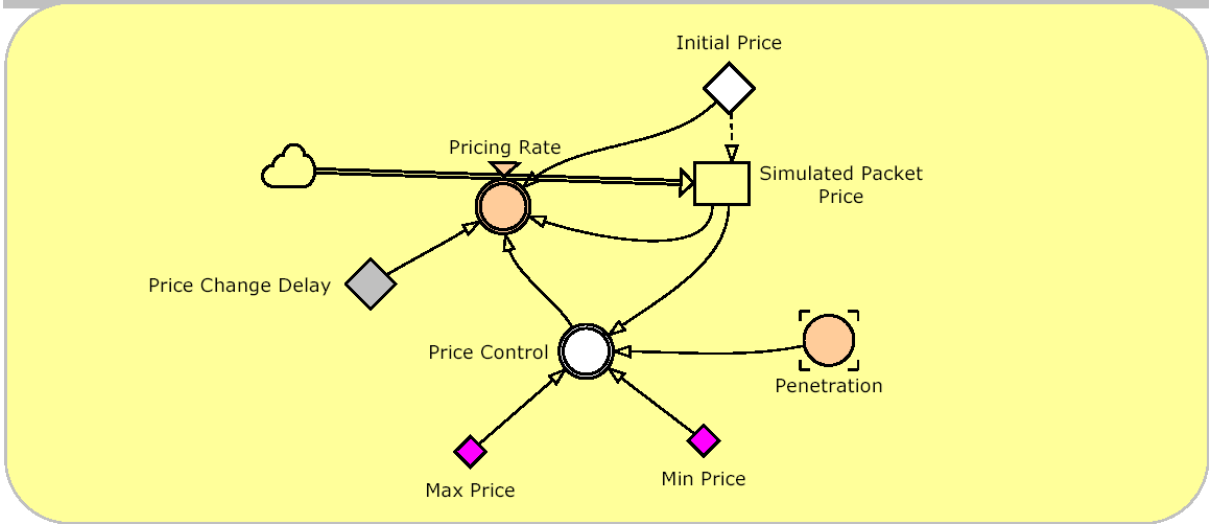
The Migration of Imode Users



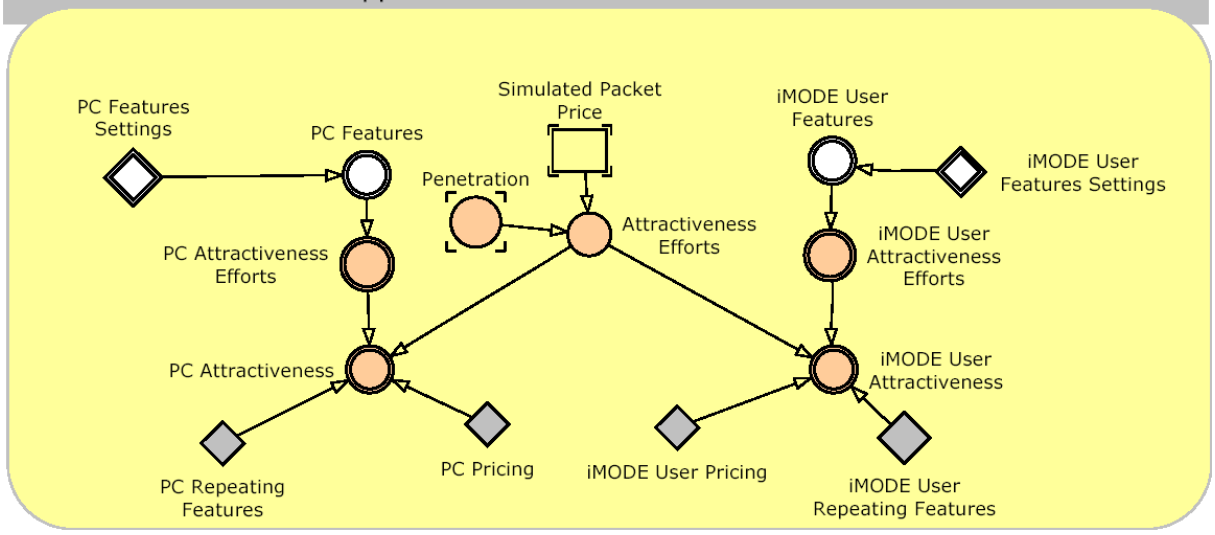
Advertising Expenditures Effect



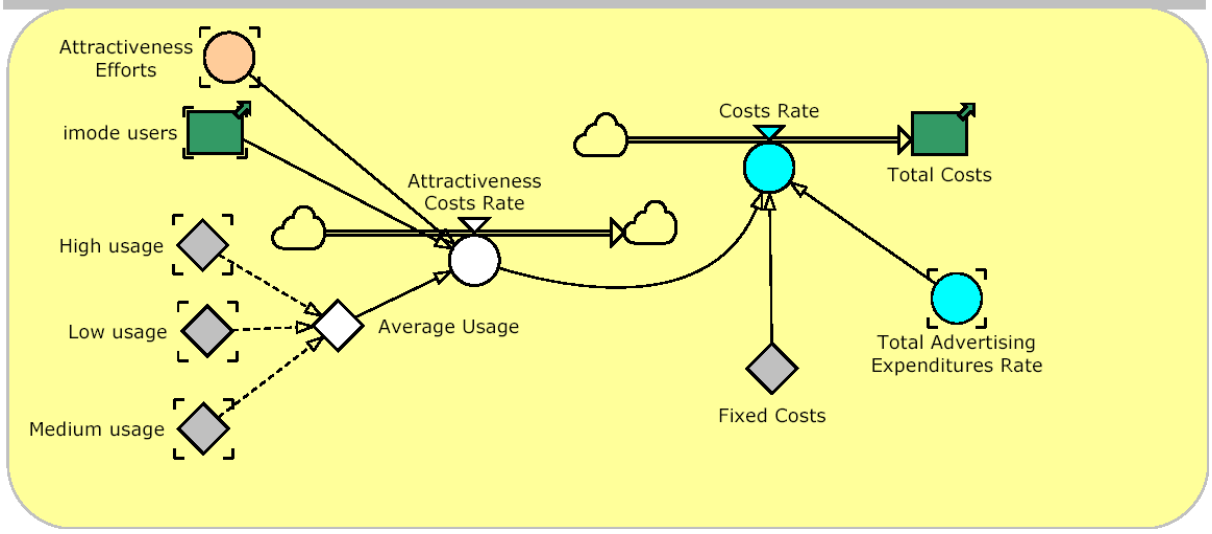
Pricing Strategy & Simulated Packet Price



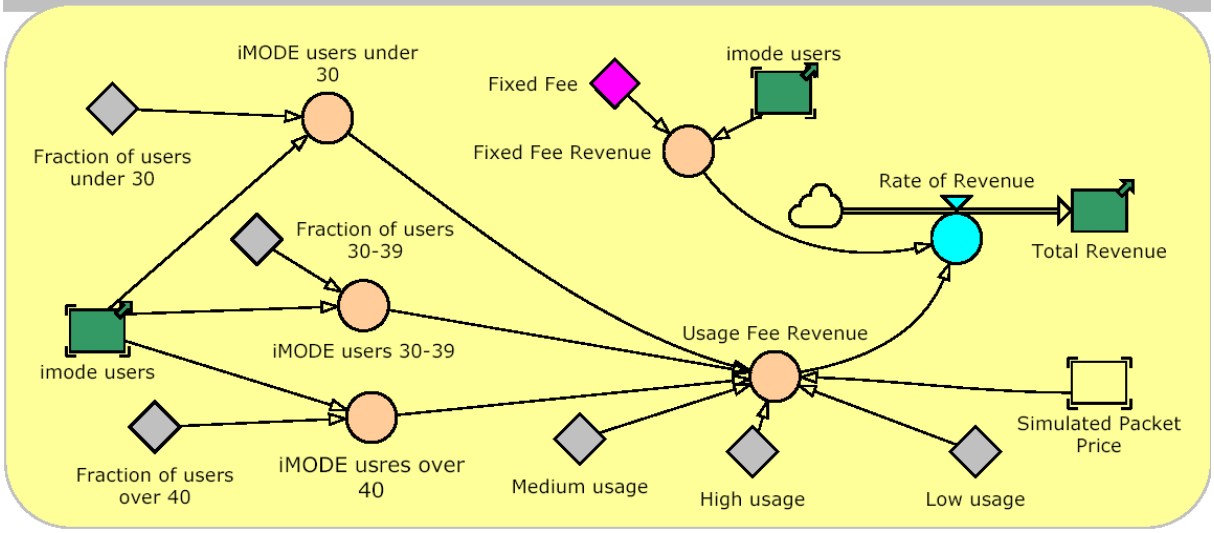
Support customers & Attractiveness Efforts



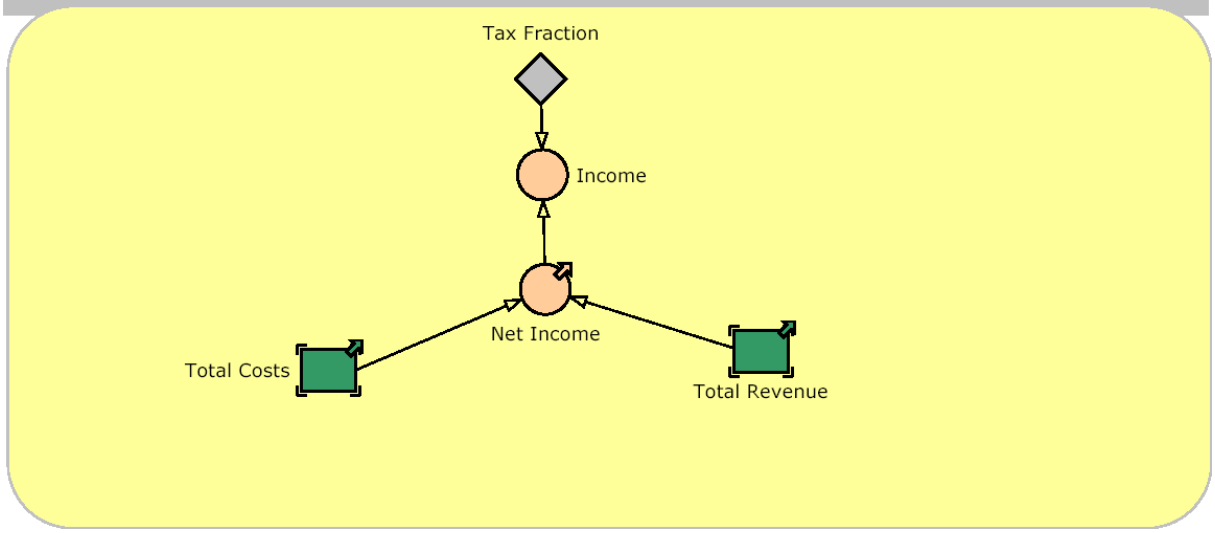
Total Costs (JPY)



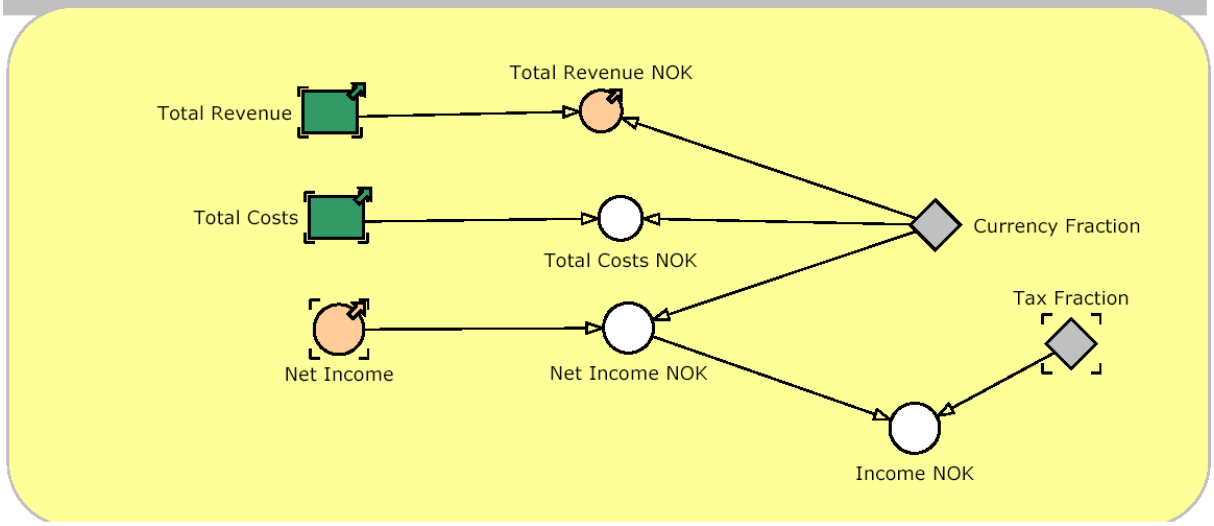
Revenue Stream & Customer's Segmentation by Age



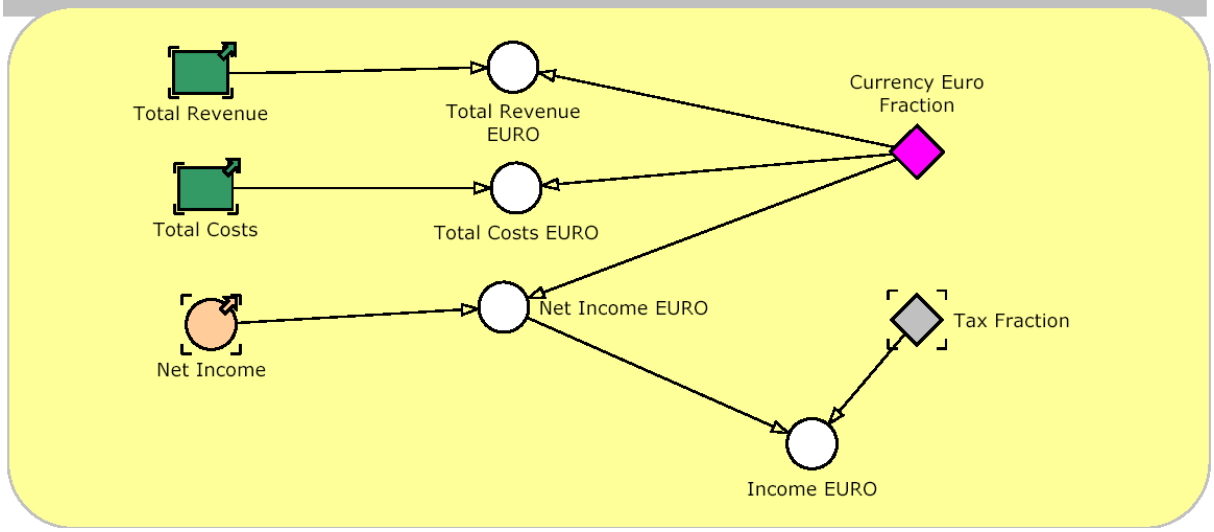
Net Income & Incme (JPY)



Revenue, Costs & Income (NOK)



Revenue, Costs & Income [EURO]



APPENDIX B

```
mainmodel Imode success modell {
    aux Adoption from Advertising {
        autotype Real
        autounit MO^-1
        def GRAPH('Advertising Expenditures Effect', 0<<JPY/MO>>, 40000<<JPY/MO>>, {0, 0.007, 0.021, 0.
            039, 0.066, 0.103, 0.138, 0.165, 0.186, 0.192, 0.195}<<3/MO>>)
        doc The nonlinear effect of advertising on adoption rate. Input is Advertising Expenditures Effect, while
            the output is the percentage of a potential customers who will adopt an imode product per month.
    }
    aux Adoption from WOM {
        autotype Real
        unit CUST/MO
        def (1 - Penetration) * 'imode users'*'Contact Rate'
        doc The number of customers adopt imode by word of mouth [WOM].
    }
    aux Adoption Rate {
        autotype Real
        autounit CUST/MO
        def 'Adoption from WOM'+'Adoption from Advertising'*'Potential Customers'
        doc The rate of adoption imode services.
    }
    level Advertising Expenditures Effect {
        autotype Real
        unit JPY/MO
        init 0
        inflow { autodef  'Advertising Expenditures Effect Rate' }
        doc The actual effect of advertising expeditures which accounts for the delay in spreading of information.
    }
    aux Advertising Expenditures Effect Rate {
        autotype Real
        unit JPY/MO²
        def ('Total Advertising Expenditures Rate' - 'Advertising Expenditures Effect') / 'Changing priod'
        doc The rate of advertising expeditures effect which accounts for the delay in spreading of information.
    }
    aux Attractiveness Costs Rate {
        autotype Real
        autounit JPY/MO
        def 'Attractiveness Efforts'*'Average Usage'*'imode users'
        doc The rate of costs due by attractiveness.
    }
    aux Attractiveness Efforts {
        autotype Real
        autounit JPY/PACKET
        def 'Simulated Packet Price'*(1-Penetration)
        doc The impact of the price on attractiveness of imode services.
    }
    const Average Usage {
        autotype Real
        unit PACKET/(CUST*MO)
        init AVERAGE('High usage','Low usage','Medium usage')
        doc The average usage by imode users a month.
    }
    const Budget estimation {
```

```
    type Integer
    init 1
    doc The control enabling manual or automatic selection of the advertising expenditures.
}
const Budget framework {
    autotype Real
    unit JPY/MO
    init 20e6
    doc The default settings for the manual advertising expenditures budget.
}
aux Budgets {
    autotype Real
    unit JPY/MO
    autodim 1..2
    def { GRAPH((TIME-STARTTIME) / TIMESTEP, 0, 2, {800, 3300, 6800, 10500, 15300, 20000, 21900,
        22500, 20800, 17500, 11500, 0, 0, 0, 0, 0, 0, 0, 0, 0}<<JPY/MO>>), 'Budget framework' }
    doc The first element is the automatic budget for advertising expenditures, while the second element is
        the manual budget (controlled by the user) for advertising expenditures.
}
const Changing priod {
    autotype Real
    unit MO
    init 3
    doc The time necessary for the information of advertising to spread.
}
const Contact Rate {
    autotype Real
    unit CUST/CUST/MO
    init 0.167
    doc Number of people per month might be heard about imode from an imode user.
}
aux Costs Rate {
    autotype Real
    autounit JPY/MO
    def 'Fixed Costs'+'Attractiveness Costs Rate'+'Total Advertising Expenditures Rate'
    doc The rate of total costs a month.
}
const Currency Euro Fraction {
    autotype Real
    unit JPY/EUR
    init 120
    doc The fraction to change the currency from JPY to NOK, norwegian krone.
}
const Currency Fraction {
    autotype Real
    unit NOK/JPY
    init 0.06
    doc The fraction to change the currency from JPY to NOK, norwegian krone.
}
const Fixed Costs {
    autotype Real
    unit JPY/MO
    init 1e9
```

```
    doc Other fixed costs a month which have been spent by NTTDoCoMo to serve their users.
}
const Fixed Fee {
    autotype Real
    unit JPY/(CUST*MO)
    init 250
    doc The subscription fee which are fixed every month an imode user must pay to access imode services.
}
aux Fixed Fee Revenue {
    autotype Real
    unit JPY/MO
    def 'Fixed Fee'*'imode users'
    doc The accumulated revenue from the fixed fee.
}
const Fraction of users 30-39 {
    autotype Real
    init 0.2
    doc The fraction of imode users who are between 30 and 39 years that has adopted imode services.
}
const Fraction of users over 40 {
    autotype Real
    init 0.3
    doc The fraction of imode users who are over 40 years that has adopted imode services.
}
const Fraction of users under 30 {
    autotype Real
    init 0.5
    doc The fraction of imode users who are under 30 years that has adopted imode services.
}
const High usage {
    autotype Real
    unit PACKET/(CUST*MO)
    init 6000
    doc The number of packets which have been used by imode users who are under 30 years a month.
}
aux iMODE User Attractiveness {
    autotype Real
    autounit JPY/PACKET
    autodim 1..2
    def ('iMODE User Attractiveness Efforts'*'iMODE User Repeating Features' *
        'Attractiveness Efforts'*'iMODE User Pricing')*(1/('iMODE User Repeating Features'+'iMODE User
        Pricing'))
    doc Attractiveness of imode user who have already adopted imode services
}
aux iMODE User Attractiveness Efforts {
    autotype Real
    dim 1..2
    def 'iMODE User Features'
    doc The impact of features on imode user who have already adopted imode services and uses them.
}
aux iMODE User Features {
    autotype Real
    dim 1..2
```

```
    def 'iMODE User Features Settings'
    doc Features of imode user who have already using imode services.
}
const iMODE User Features Settings {
    type Real
    dim 1..2
    init {1,3}
    doc Features range that might apeal to imode users.
}
aux imode user lifetime {
    autotype Real
    unit MO
    def 'Subscription priode'/(1 -'Returnable Probability')
    doc Average number of years customer subscribes to the imode service.
}
const iMODE User Pricing {
    autotype Real
    init 1
    doc The weight of imode user's price.
}
const iMODE User Repeating Features {
    autotype Real
    init 1
    doc Features of imode user's weight.
}
level imode users {
    autotype Real
    unit CUST
    init 0
    outflow { autodef  'imode users  Break' }
    inflow { autodef  'Adoption Rate' }
    doc The actual number of customers who have adopted imode services.
}
aux imode users  Break {
    autotype Real
    unit CUST/MO
    def 'imode users'/'imode user lifetime'
    doc The rate at which an imode users become a non imode users.
        This is driven by a new telecom.services might be showing up or launching into the market, i.e
        customers might be stopped using imode services.
}
aux iMODE users 30-39 {
    autotype Real
    autounit CUST
    def 'Fraction of users 30-39'*'imode users'
    doc The number of imode users who are between 30 and 39 years that has adopted imode services.
}
aux iMODE users under 30 {
    autotype Real
    autounit CUST
    def 'Fraction of users under 30'*'imode users'
    doc The number of imode users who are under 30 years that has adopted imode services.
}
```

```
aux iMODE usres over 40 {
    autotype Real
    autounit CUST
    def 'Fraction of users over 40'*'imode users'
    doc The number of imode users who are over 40 years that has adopted imode services.
}
aux Income {
    autotype Real
    autounit JPY
    def 'Net Income'-('Net Income'*'Tax Fraction')
    doc The total income without taxes.(JPY)
}
aux Income EURO {
    autotype Real
    autounit EUR
    def 'Net Income EURO'-('Net Income EURO'*'Tax Fraction')
    doc The total income without taxes given by european currency (EURO).
}
aux Income NOK {
    autotype Real
    autounit NOK
    def 'Net Income NOK'-('Net Income NOK'*'Tax Fraction')
    doc The total income without taxes given by norwegean Krone.
}
const Initial Price {
    autotype Real
    unit JPY/PACKET
    init 0.01
    doc The initial packet's price.
}
const Low usage {
    autotype Real
    unit PACKET/(CUST*MO)
    init 1000
    doc The number of packets which have been used by imode users who are between  30 and 39 years a
        month.
}
const Market Size {
    autotype Real
    unit CUST
    init 45e6
    doc The total market size, i.e. the expectation of how many customers might be using imode over period
        of time.
}
const Max Price {
    autotype Real
    unit JPY/PACKET
    init 0.6
    doc The maximum packet's price.
}
const Medium usage {
    autotype Real
    unit PACKET/(CUST*MO)
```

```
    init 3000
    doc The number of packets which have been used by imode users who are over 40 years a month.
}
const Min Price {
    autotype Real
    unit JPY/PACKET
    init 0.2
    doc The minumum packet's price.
}
aux Net Income {
    autotype Real
    autounit JPY
    def 'Total Revenue'-'Total Costs'
    doc The total income accumulated by serving imode users, including taxes.(JPY)
}
aux Net Income EURO {
    autotype Real
    unit EUR
    def 'Net Income'/'Currency Euro Fraction'
    doc The total income accumulated by serving imode users, including taxes given by european currency
        (EURO).
}
aux Net Income NOK {
    autotype Real
    autounit NOK
    def 'Net Income'*'Currency Fraction'
    doc The total income accumulated by serving imode users, including taxes given by norwegean Krone.
}
level Non imode users {
    autotype Real
    unit CUST
    init 0
    inflow { autodef  'imode users  Break' }
    doc Customers leave imode to another interactive product, as an example a new telecom. services which
        might be more interesting for their needs.
}
aux PC Attractiveness {
    autotype Real
    autounit JPY/PACKET
    autodim 1..2
    def ('PC Attractiveness Efforts'*'PC Repeating Features' *
        'Attractiveness Efforts'*'PC Pricing')*(1/('PC Repeating Features'+'PC Pricing'))
    doc Attractiveness of Potential Customer who have not yet adopted imode services.
}
aux PC Attractiveness Efforts {
    autotype Real
    dim 1..2
    def 'PC Features'
    doc The impact of features on Potential Customer who have not yet adoped imode services.
}
aux PC Features {
    autotype Real
    autodim 1..2
```

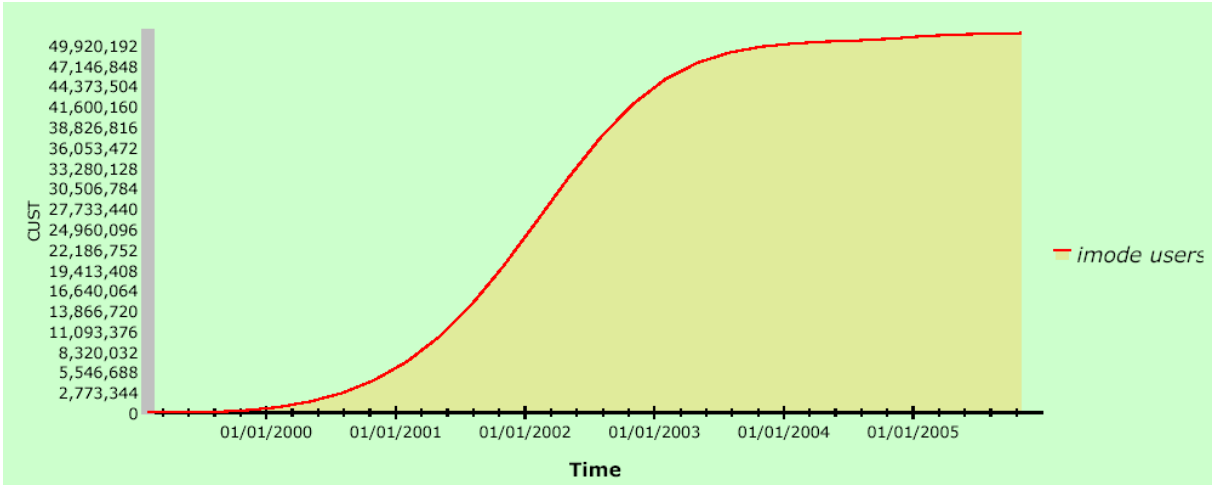
```
    def 'PC Features Settings'
    doc Features of Potential Customer who have not yet adopted imode services.
}
const PC Features Settings {
    autotype Real
    dim 1..2
    init {1,2}
    doc Features range that might apeal to Potential Customer.
}
const PC Pricing {
    autotype Real
    init 1
    doc The weight of Potential Customer's price
}
const PC Repeating Features {
    autotype Real
    init 1
    doc Features of the potential customer's weight.
}
aux Penetration {
    autotype Real
    def 'imode users'/'Market Size'
    doc The fraction of the Market Size that has adopted imode services.
}
level Potential Customers {
    autotype Real
    unit CUST
    init 'Market Size'
    outflow { autodef  'Adoption Rate' }
    doc The number of potential customers is determined by the Market size and the imode users.
}
const Price Change Delay {
    autotype Real
    unit MO
    init 12
    doc The time delay to change the price..
}
aux Price Control {
    autotype Real
    unit JPY/PACKET
    dim 1..3
    def AVERAGE('Simulated Packet Price','Max Price','Min Price'*(1-Penetration))
    doc Packet's price control which is the average of simulated packet price,max price and the multiplication
        of min price with (1- the penetration).
}
aux Pricing Rate {
    autotype Real
    autounit JPY/(MO*PACKET)
    autodim 1..3
    def ('Initial Price'+'Price Control'-'Simulated Packet Price')/'Price Change Delay'
    doc The rate of simulated packet's price.
}
aux Rate of Revenue {
```

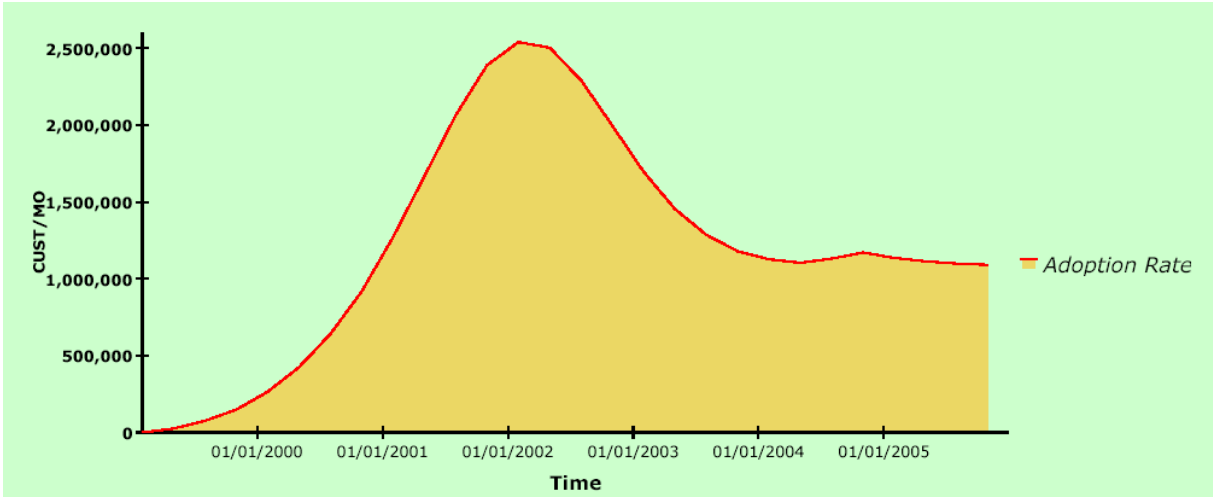
```
    autotype Real
    autounit JPY/MO
    def 'Fixed Fee Revenue'+'Usage Fee Revenue'
    doc The rate of total fees from both accumulated usage fees and accumulated fixed fees. The stream of
        revenue to NTTDoCoMo every month
}
const Returnable Probability {
    autotype Real
    init 0.5
    doc The probability of a non imode user who might return to use imode services, i.e. might being an
        imode user again.
}
level Simulated Packet Price {
    autotype Real
    unit JPY/PACKET
    init 'Initial Price'
    inflow { autodef  COLLECT('Pricing Rate') }
    doc The simulated price of one packet.
}
const Subscription priode {
    autotype Real
    unit MO
    init 24
    doc Duration of time imode users sign up for services.
}
const Tax Fraction {
    autotype Real
    init 0.24
    doc The tax fraction vendor pays to the Government.
}
level Total Advertising Expenditures {
    autotype Real
    unit JPY
    init 0
    inflow { autodef  'Total Advertising Expenditures Rate' }
    doc The total advertising expeditures.
}
aux Total Advertising Expenditures Rate {
    autotype Real
    unit JPY/MO
    def LOOKUP( Budgets, 'Budget estimation')
    doc The amount of advertising expeditures per year(i.e. the advertising expenditures rate).
}
level Total Costs {
    autotype Real
    unit JPY
    init 0
    inflow { autodef  'Costs Rate' }
    doc The total costs accumulated to support imode users given by japaneese yen.
}
aux Total Costs EURO {
    autotype Real
    autounit EUR
```

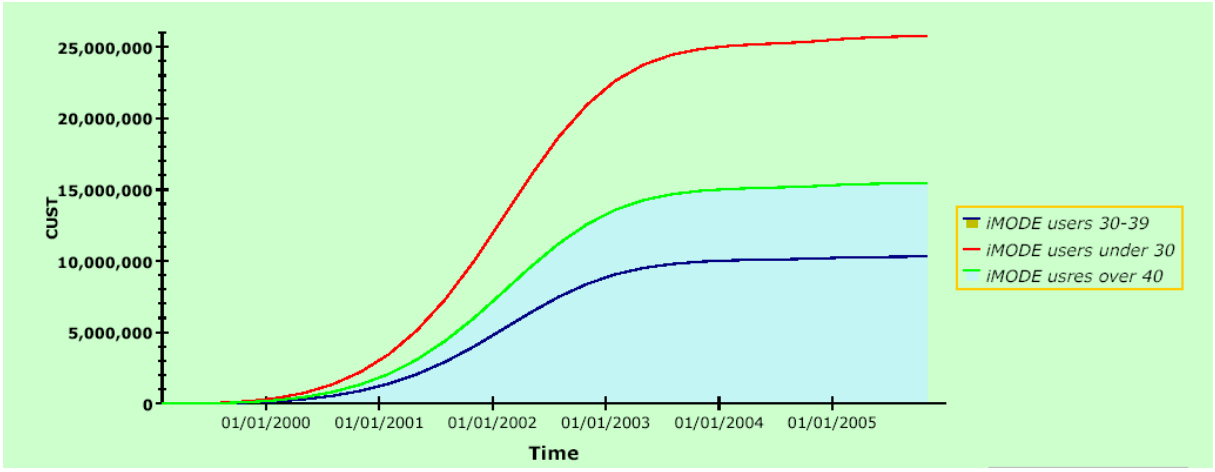
```
        def 'Total Costs'/'Currency Euro Fraction'
        doc Total costs accumulated to support imode users given by european currency (EURO).
    }
    aux Total Costs NOK {
        autotype Real
        autounit NOK
        def 'Total Costs'*'Currency Fraction'
        doc Total costs accumulated to support imode users given by norwegean Krone.
    }
    level Total Revenue {
        autotype Real
        unit JPY
        init 0
        inflow { autodef  'Rate of Revenue' }
        doc Total revenue accumulated by usage and fixed fee from imode users(JPY).
    }
    aux Total Revenue EURO {
        autotype Real
        autounit EUR
        def 'Total Revenue'/'Currency Euro Fraction'
        doc Total revenue accumulated by usage and fixed fee from imode users given by european currency
            (EURO).
    }
    aux Total Revenue NOK {
        autotype Real
        unit NOK
        def 'Total Revenue'*'Currency Fraction'
        doc Total revenue accumulated by usage and fixed fee from imode users given by norwegean Krone.
    }
    aux Usage Fee Revenue {
        autotype Real
        unit JPY/MO
        def ('Simulated Packet Price'*'High usage'*'iMODE users under 30')+('Simulated Packet Price'*'Low
            usage'*'iMODE users 30-39')+('Simulated Packet Price'*'Medium usage'*'iMODE usres over 40')
        doc The accumulated revenue from usage fees.
    }
}
range Strategy {
    def 1..3
}
unit % {
    def ATOMIC
    doc Percent
}
unit CUST {
    def ATOMIC
    doc Subscribers of the imode-services
}
unit EUR {
    def __CURRENCY("EUR")
    doc European Currency
}
unit JPY {
```

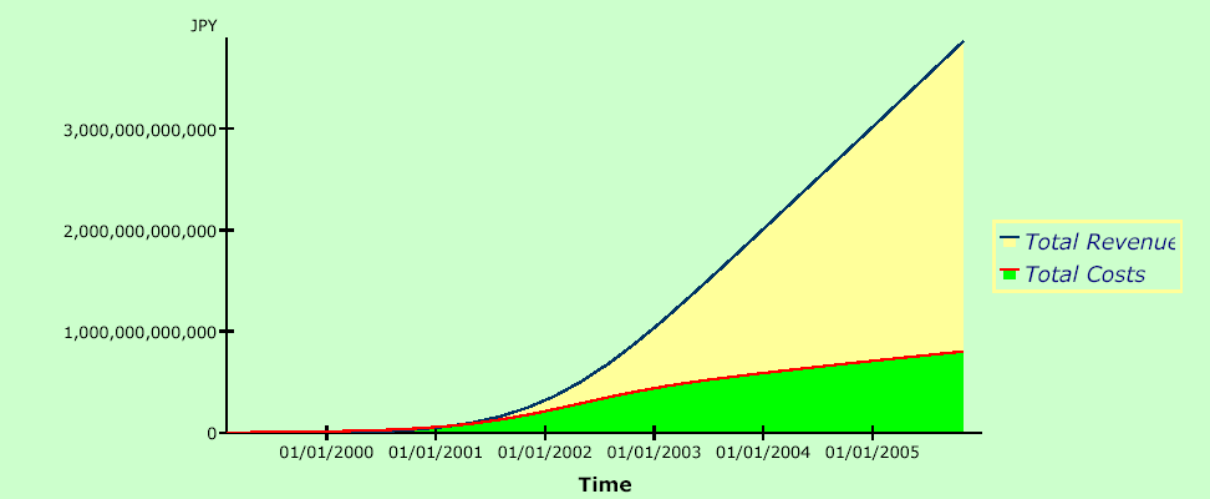
```
    def __CURRENCY("JPY")
    doc Japanese Yen
}
unit NOK {
    def __CURRENCY("NOK")
    doc Norwegian Kroner
}
unit PACKET {
    def ATOMIC
    doc The unit of data (1 packet = 128 byte)
}
```

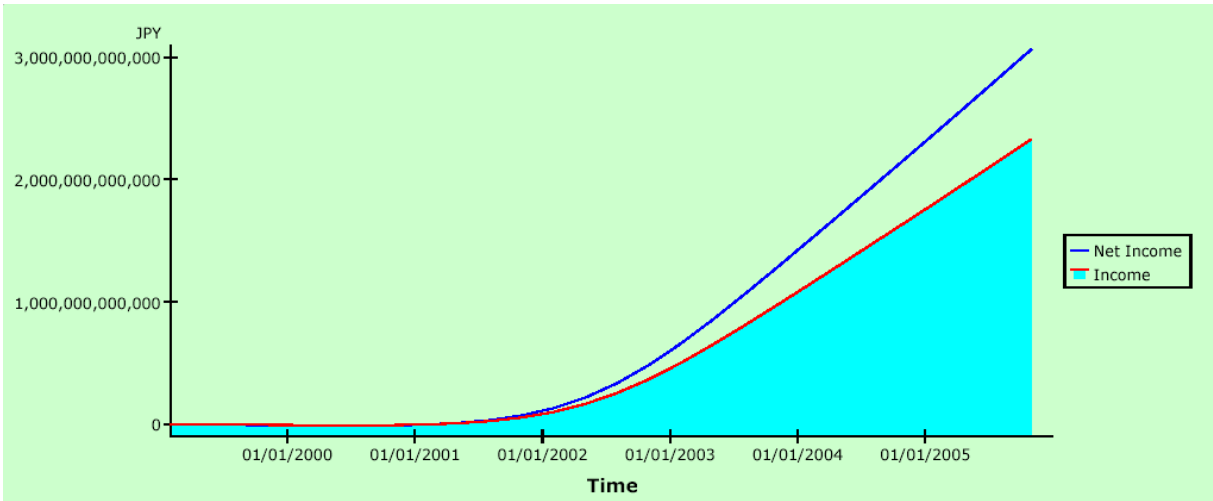
APPENDIX C

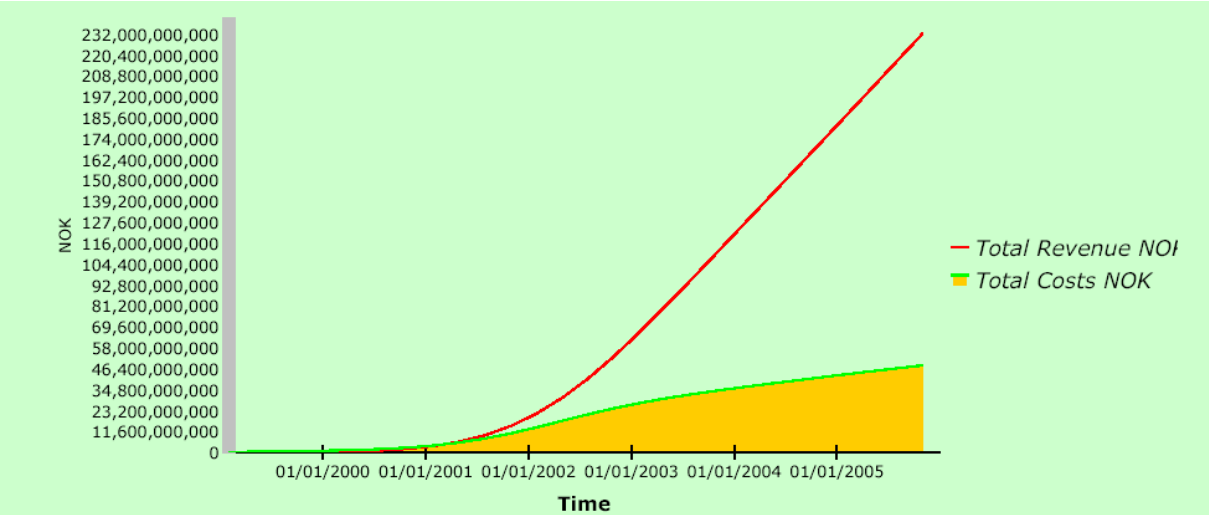

49,920,192
47,146,848
44,373,504
41,600,160
38,826,816
36,053,472
33,280,128
30,506,784
27,733,440
24,960,096
22,186,752
19,413,408
16,640,064
13,866,720
11,093,376
8,320,032
5,546,688
2,773,344
0
CUST
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
imode users

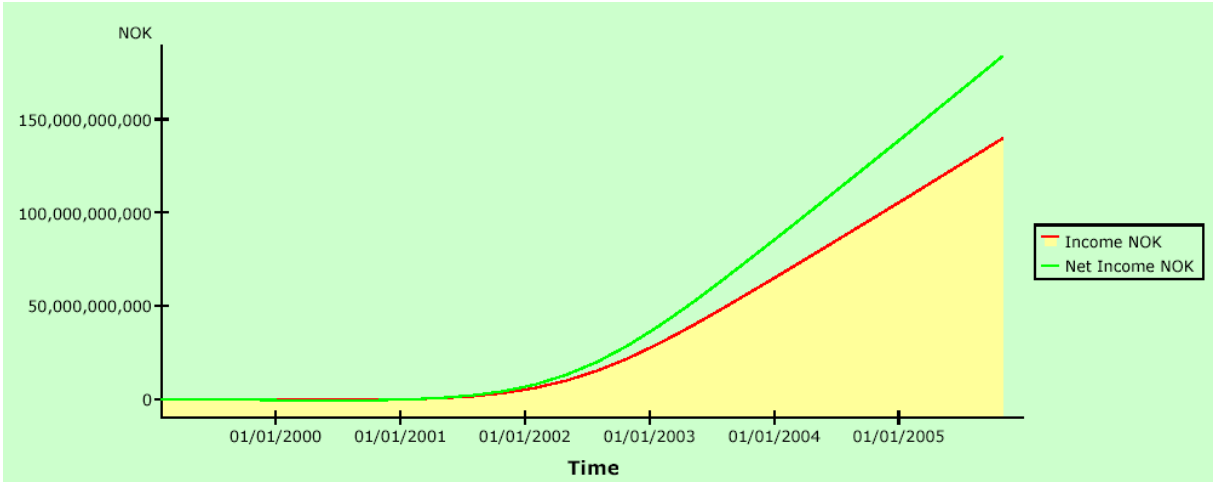

2,500,000
2,000,000
1,500,000
1,000,000
500,000
0
CUST/MO
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
Adoption Rate

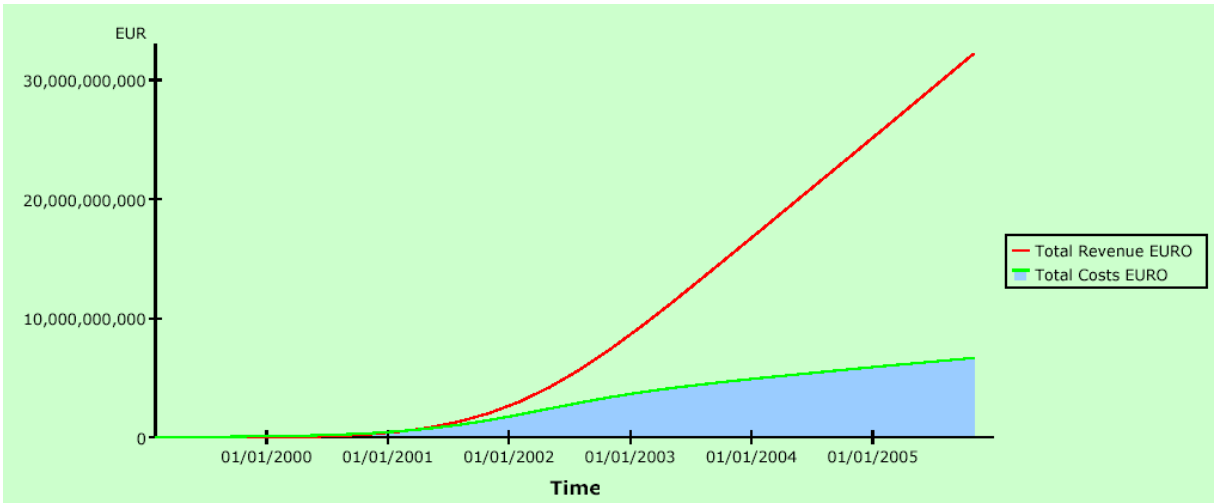

25,000,000
20,000,000
15,000,000
10,000,000
5,000,000
0
CUST
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
iMODE users 30-39
iMODE users under 30
iMODE usres over 40

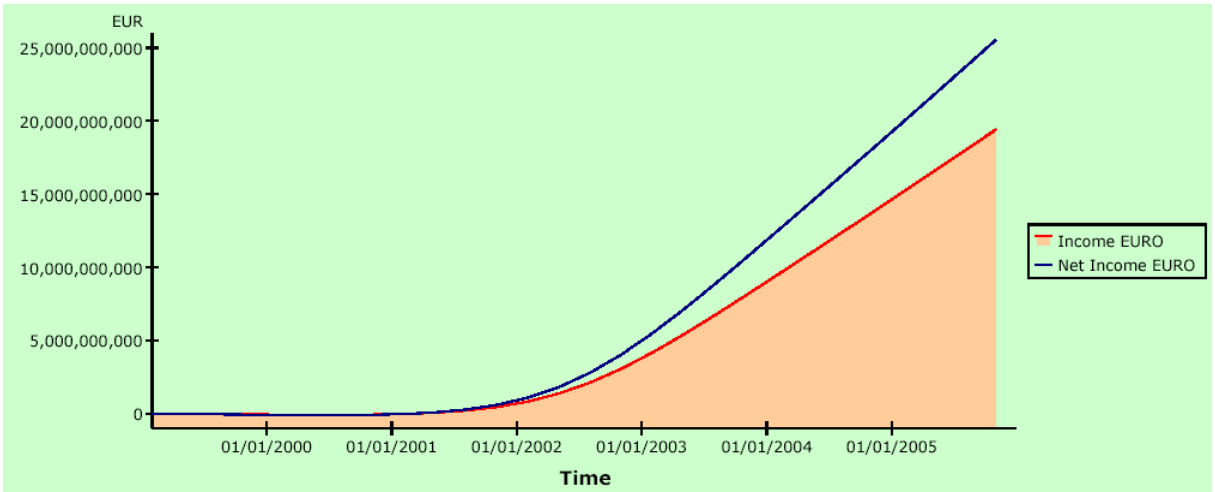

JPY
3,000,000,000,000
2,000,000,000,000
1,000,000,000,000
0
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
Total Revenue
Total Costs

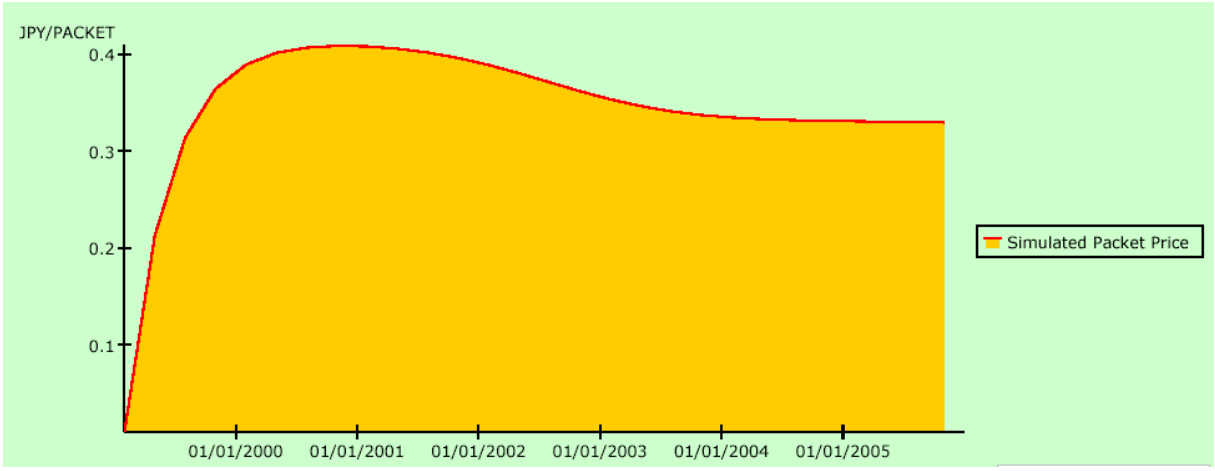

JPY
3,000,000,000,000
2,000,000,000,000
1,000,000,000,000
0
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
Net Income
Income

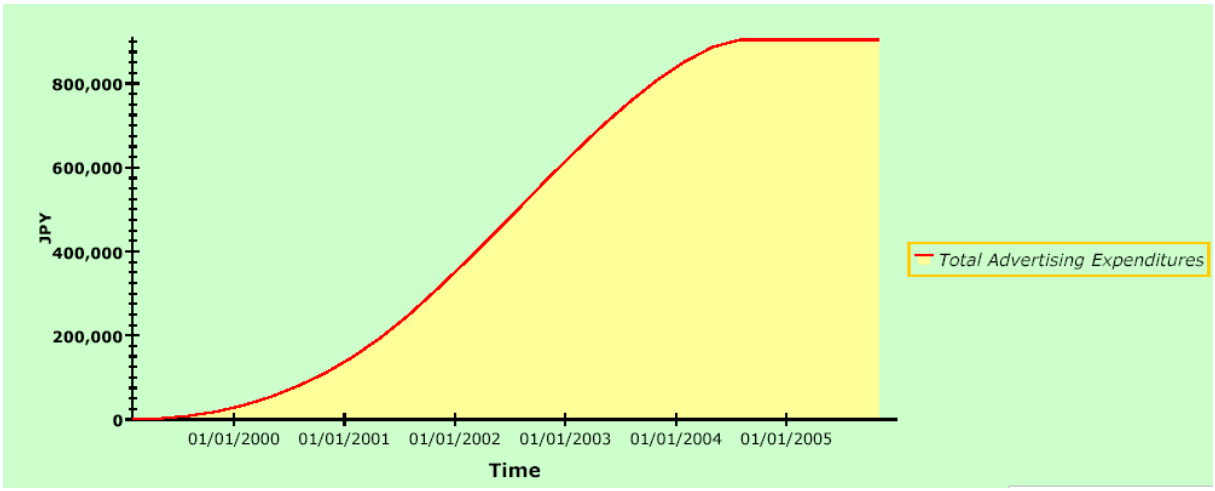

NOK
232,000,000,000
220,400,000,000
208,800,000,000
197,200,000,000
185,600,000,000
174,000,000,000
162,400,000,000
150,800,000,000
139,200,000,000
127,600,000,000
116,000,000,000
104,400,000,000
92,800,000,000
81,200,000,000
69,600,000,000
58,000,000,000
46,400,000,000
34,800,000,000
23,200,000,000
11,600,000,000
0
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
Total Revenue NOK
Total Costs NOK


NOK
150,000,000,000
100,000,000,000
50,000,000,000
0
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
Income NOK
Net Income NOK


EUR
30,000,000,000
20,000,000,000
10,000,000,000
0
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
Total Revenue EURO
Total Costs EURO


EUR
25,000,000,000
20,000,000,000
15,000,000,000
10,000,000,000
5,000,000,000
0
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
Income EURO
Net Income EURO


JPY/PACKET
0.4
0.3
0.2
0.1
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Simulated Packet Price


800,000
600,000
400,000
200,000
0
JPY
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
Time
Total Advertising Expenditures

APPENDIX D

// Imode.java

```
1 import javax.swing.text.html.*;
2 import java.awt.*;
3 import java.awt.print.PrinterJob;
4 import java.awt.event.*;
5 import java.awt.Image;
6 import java.awt.FileDialog;
7 import java.net.*;
8 import java.net.URL;
9 import java.applet.*;
10 import javax.swing.event.*;
11 import javax.swing.*;
12 import javax.swing.tree.DefaultMutableTreeNode;
13 import javax.swing.tree.DefaultTreeModel;
14 import javax.swing.tree.TreePath;
15 import javax.swing.border.*;
16 import java.io.*;
17 import javax.print.*;
18 import javax.print.attribute.*;
19 import javax.print.attribute.standard.*;
20 import javax.swing.colorchooser.*;
21 import javax.swing.filechooser.*;
22 import javax.accessibility.*;
23 import java.util.*;
24 import javax.swing.Timer;
25 import javax.swing.text.html.*;
26
27 public class Imode extends JApplet implements ActionListener,MouseListener,ItemListener,HyperlinkListener
28 {
29   private JEditorPane html,codeEditor;
30   Timer timer;
31   Action loadAction,stopAction;
32   String txt = "How are You";
33   int textLocation =90;
34   private TabbedPaneModel modelTabbedPane;
35   private TabbedPaneGraph graphTabbedPane;
36   static private final String newline = "\n";
37   private JFileChooser fc;
38   private FileInputStream textStream;
39   private JMenuBar mb;
40   private JMenu file,edit,view,simulation,reports,modelPane,help;
41   private JMenuItem new_,open,saveItem,print,exit;
42   private JMenuItem undo,cut,copy,paste;
43   private JMenuItem run;
44   private JMenuItem revenueR,imodeUsers;
45   private JMenuItem about;
46   private JMenuItem modelItem;
47   private JMenuItem GUIItem;
48   private JMenuItem graphItem;
49   private JMenuItem showBlackBoard;
50   private JMenuItem iMODEJava,messageJava;
51   private JRadioButton euro,nok,yen;
52   private ButtonGroup g_roup;
53   private JButton vbutton,clear,start;
54   private JButton open_icon_button,save_icon_button,cut_icon_button,paste_icon_button;
55   private JButton help_icon_button,sun_icon_button,java_icon_button,ps_icon_button;
56   private JButton copy_icon_button,print_icon_button,myLabelButton,tableButton,new_icon_button;
57   public  JTextField init_sub,wom,loss;
58   private JLabel introduction,init_subLabel,womLabel,lossLabel;
59   private JPanel introd_panel,cb_panel,data,lossPanel,womPanel;
60   private JPanel icons_panel,tablePanel;
61   private JPanel textAreaPanel;
62   private Icon icon;
63   //public  JTextArea moroTextArea;
64   public MyTextArea text;
65   private ImageIcon new_icon,save_icon,open_icon,paste_icon,close_icon,panel_icon;
66   private Message msg;
67   private Warning warning;
68   private VerifyingMessage verifyingMessage;
69   private Revenue revenue;
70   private ImodeUsers imodUsersTable;
71   boolean badURL;
72   private DefaultMutableTreeNode home,rootNode;
73   private JTree tree;
74   private JScrollPane paneTree,displayPane,buttonPane,pane,moroPane;
75   protected Graphics g;
76   private PrinterJob printerJob;
```

```
77  private Image myImage;
78  private URL leftButtonURL,sun_URL;
79  private URL wordURL,pdfURL;
80  protected String leftButtonFilename;
81  private JTable table;
82  TreePath selectedPath;
83  private DocFlavor flavor;
84  private Doc doc;
85  private Component panel5,panel6;
86  public final String jpg = "jpg";
87  public final String gif = "gif";
88  JButton xButton;
89  private TabbedPaneGUI tabbedPaneGUI;
90  JFrame frame;
91  JViewport vp ;
92  private CodeViewer myViewer;
93
94  public void init()
95  {
96    myViewer = new CodeViewer();
97    frame = new JFrame("IMODE GRAPHS PICTURES");
98    myImage = getImage(getDocumentBase(),"photos/01.gif");
99    modelTabbedPane = new TabbedPaneModel();
100   graphTabbedPane = new TabbedPaneGraph();
101   tabbedPaneGUI = new         TabbedPaneGUI();
102   Color c = Color.magenta;
103   Color cc = Color.green;
104   Color ccc = Color.orange;
105   Color cccc = Color.lightGray;
106   Color ccccc = Color.GRAY;
107   Font treeFont = new             Font("JTree",Font.PLAIN,12);
108   Font menuFont = new             Font("JMenu",Font.PLAIN,12);
109   Font textFont = new             Font("JScrollPane",Font.PLAIN,14);
110   text = new MyTextArea();
111   text.getAccessibleContext().setAccessibleName("What        is your      name?");
112   text.getAccessibleContext().setAccessibleName("How old are you");
113   text.setBackground(ccc);
114   text.setBorder(new SoftBevelBorder(BevelBorder.LOWERED));
115   text.setEditable(false);
116   rootNode = new DefaultMutableTreeNode("Home");
117   DefaultMutableTreeNode introduction = new DefaultMutableTreeNode("Introduction");
118   DefaultMutableTreeNode problem = new DefaultMutableTreeNode("Preoblem Description");
119   DefaultMutableTreeNode iMODE_services = new DefaultMutableTreeNode("iMODE Services");
120   DefaultMutableTreeNode technical = new DefaultMutableTreeNode("Technical Perspective");
121   DefaultMutableTreeNode imode_wap = new DefaultMutableTreeNode("iMODE versus WAP");
122   DefaultMutableTreeNode imode_success = new DefaultMutableTreeNode("iMODE Success");
123   DefaultMutableTreeNode success_factors = new DefaultMutableTreeNode("Success factors of iMODE");
124   problem.add(iMODE_services);
125   problem.add(technical);
126   problem.add(imode_wap);
127   problem.add(imode_success);
128   problem.add(success_factors);
129   DefaultMutableTreeNode model_purpose = new DefaultMutableTreeNode("Model Purpose and Uses");
130   DefaultMutableTreeNode ref_behaviour = new DefaultMutableTreeNode("Reference Behaviour");
131   DefaultMutableTreeNode imodeU_rb = new DefaultMutableTreeNode("iMODE users Reference Behaviour");
132   DefaultMutableTreeNode usageRB = new DefaultMutableTreeNode("Usage Reference Behaviour");
133   ref_behaviour.add(imodeU_rb);
134   ref_behaviour.add(usageRB);
135   DefaultMutableTreeNode CLD = new DefaultMutableTreeNode("Causal Loop Diagram");
136   DefaultMutableTreeNode WOM_CLD = new DefaultMutableTreeNode("CLD of Adoption from WOM");
137   DefaultMutableTreeNode ADV_CLD = new DefaultMutableTreeNode("CLD of Adoption from Advertising");
138   DefaultMutableTreeNode Influence_Diagram = new DefaultMutableTreeNode("Influence Diagram");
139   CLD.add(WOM_CLD);
140   CLD.add(ADV_CLD);
141   CLD.add(Influence_Diagram);
142   DefaultMutableTreeNode SFD = new DefaultMutableTreeNode("Stock & Flow Diagram");
143   DefaultMutableTreeNode ps_model = new DefaultMutableTreeNode("Modelling by PowerSim");
144   DefaultMutableTreeNode GUI = new DefaultMutableTreeNode("Graphical User Interface");
145   DefaultMutableTreeNode resources = new DefaultMutableTreeNode("Resources");
146   rootNode.add(introduction);
147   rootNode.add(problem);
148   rootNode.add(model_purpose);
149   rootNode.add(ref_behaviour);
150   rootNode.add(CLD);
151   rootNode.add(SFD);
152   rootNode.add(ps_model);
```

```
153   rootNode.add(GUI);
154   rootNode.add(resources);
155   tree = new JTree(rootNode);
156   tree.setFont(treeFont);
157   tree.addMouseListener (this);
158   tree.setRootVisible(false);
159   paneTree = new JScrollPane();
160   paneTree.setViewportView(tree);
161   paneTree.setWheelScrollingEnabled(true);
162   paneTree.setBorder(new SoftBevelBorder(BevelBorder.LOWERED));
163   displayPane         = new JScrollPane(text);
164   displayPane.setFont(textFont);
165   vp = displayPane.getViewport();
166   displayPane.setBorder(new SoftBevelBorder(BevelBorder.LOWERED));
167   mb = new JMenuBar();
168   mb.setBorder(BorderFactory.createLoweredBevelBorder());
169   mb.setBorder(BorderFactory.createEtchedBorder());
170   file = mb.add(new JMenu("File"));
171   file.setMnemonic(KeyEvent.VK_F);
172   file.setFont(menuFont);
173   edit = mb.add(new JMenu("Edit"));
174   edit.setMnemonic(KeyEvent.VK_E);
175   edit.setFont(menuFont);
176   view = mb.add(new JMenu("View codes"));
177   view.setMnemonic(KeyEvent.VK_V);
178   view.setFont(menuFont);
179   simulation = mb.add(new JMenu("Simulation"));
180   simulation.setMnemonic(KeyEvent.VK_S);
181   simulation.setFont(menuFont);
182   reports = mb.add(new JMenu("Reports"));
183   reports.setMnemonic(KeyEvent.VK_R);
184   reports.setFont(menuFont);
185   modelPane = mb.add(new JMenu("Model"));
186   modelPane.setMnemonic(KeyEvent.VK_M);
187   modelPane.setFont(menuFont);
188   help = mb.add(new JMenu("Help"));
189   help.setMnemonic(KeyEvent.VK_H);
190   help.setFont(menuFont);
191
192   new_ = new JMenuItem ("New");
193   new_.setFont(menuFont);
194   new_.addActionListener(this);
195   file.add(new_);
196   open = new JMenuItem ("Open");
197   open.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O, ActionEvent.CTRL_MASK));
198   open.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
199   open.setFont(menuFont);
200   open.setActionCommand("http//www.vg.no");
201   open.addActionListener(this);
202   file.add(open);
203
204   saveItem = new JMenuItem("Save");
205   saveItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_E, ActionEvent.CTRL_MASK));
206   saveItem.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
207   saveItem.setFont(menuFont);
208   saveItem.addActionListener(this);
209   file.add(saveItem);
210
211   print = new JMenuItem("Print");
212   print.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_P, ActionEvent.CTRL_MASK));
213   print.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
214   print.setFont(menuFont);
215   print.addActionListener(this);
216   file.add(print);
217
218   exit = new JMenuItem ("Exit");
219   exit.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_F4, ActionEvent.ALT_MASK));
220   exit.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
221   exit.setFont(menuFont);
222   exit.addActionListener(this);
223   file.add(exit);
224
225   undo = new JMenuItem ("Undo");
226   undo.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_Z, ActionEvent.CTRL_MASK));
227   undo.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
228   undo.setFont(menuFont);
```

```
229   undo.addActionListener(this);
230   edit.add(undo);
231
232   copy = new JMenuItem ("Copy");
233   copy.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C, ActionEvent.CTRL_MASK));
234   copy.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
235   copy.setFont(menuFont);
236   copy.addActionListener(this);
237   edit.add(copy);
238
239   cut = new JMenuItem  ("Cut");
240   cut.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X, ActionEvent.CTRL_MASK));
241   cut.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
242   cut.setFont(menuFont);
243   cut.addActionListener(this);
244   edit.add(cut);
245
246   paste = new JMenuItem("Paste");
247   paste.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_V, ActionEvent.CTRL_MASK));
248   paste.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
249   paste.setFont(menuFont);
250   paste.addActionListener(this);
251   edit.add(paste);
252
253   showBlackBoard = new JMenuItem("Black Board");
254   showBlackBoard.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_B, ActionEvent.CTRL_MASK));
255   showBlackBoard.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
256   showBlackBoard.setFont(menuFont);
257   showBlackBoard.addActionListener(this);
258
259   messageJava = new JMenuItem("Message.java");
260   messageJava.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_M, ActionEvent.CTRL_MASK));
261   messageJava.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
262   messageJava.setFont(menuFont);
263   messageJava.addActionListener(this);
264
265   iMODEJava = new JMenuItem("iMODE.java");
266   iMODEJava.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_I, ActionEvent.CTRL_MASK));
267   iMODEJava.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
268   iMODEJava.setFont(menuFont);
269   iMODEJava.addActionListener(this);
270
271   view.add(showBlackBoard);
272   view.add(iMODEJava);
273   view.add(messageJava);
274
275   run = new JMenuItem  ("Run");
276   run.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_R, ActionEvent.CTRL_MASK));
277   run.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
278   run.setFont(menuFont);
279   run.addActionListener(this);
280
281   JMenuItem forwardStep = new JMenuItem  ("Forward  >>");
282   //forwardStep.setAccelerator(KeyStroke.getKeyStroke();
283   forwardStep.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
284   forwardStep.setFont(menuFont);
285   forwardStep.addActionListener(this);
286
287   JMenuItem backwardStep = new JMenuItem  ("Backward  <<");
288   //backwardStep.setAccelerator(KeyStroke.getKeyStroke();
289   backwardStep.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
290   backwardStep.setFont(menuFont);
291   backwardStep.addActionListener(this);
292
293   JMenuItem pause = new JMenuItem  ("Pause");
294   //backwardStep.setAccelerator(KeyStroke.getKeyStroke();
295   pause.getAccessibleContext().setAccessibleDescription("This doesn't really do anything");
296   pause.setFont(menuFont);
297   pause.addActionListener(this);
298
299   simulation.add(run);
300   simulation.add(forwardStep);
301   simulation.add(backwardStep);
302   simulation.add(pause);
303
304   imodeUsers = new JMenuItem("iMODE users report");
```

```
305   imodeUsers.setFont(menuFont);
306   imodeUsers.addActionListener(this);
307   reports.add(imodeUsers);
308
309   revenueR = new JMenuItem("Revenue report");
310   revenueR.setFont(menuFont);
311   revenueR.addActionListener(this);
312   reports.add(revenueR);
313
314   modelItem = new JMenuItem("Model in PowerSim");
315   modelItem.setFont(menuFont);
316   modelItem.addActionListener(this);
317
318   GUIItem = new JMenuItem("Report Images ");
319   GUIItem.setFont(menuFont);
320   GUIItem.addActionListener(this);
321
322   graphItem = new JMenuItem("Graphs ");
323   graphItem.setFont(menuFont);
324   graphItem.addActionListener(this);
325
326
327   modelPane.add(modelItem);
328   modelPane.add(GUIItem);
329   modelPane.add(graphItem);
330
331   about = new JMenuItem("About IMODE success in Japan");
332   about.setFont(menuFont);
333   about.addActionListener(this);
334   help.add(about);
335
336   new_icon = createImageIcon("icons/new.gif");
337   new_icon_button = new JButton(new_icon);
338   new_icon_button.setBorder(BorderFactory.createEtchedBorder());
339   new_icon_button.addActionListener(this);
340
341   open_icon = createImageIcon("icons/open.gif");
342   open_icon_button = new JButton(open_icon);
343   open_icon_button.addActionListener(this);
344   open_icon_button.setAlignmentX(0);
345   open_icon_button.setBorder(BorderFactory.createEtchedBorder());
346
347   save_icon = createImageIcon("icons/save.gif");
348   save_icon_button = new JButton(save_icon);
349   save_icon_button.addActionListener(this);
350   save_icon_button.setBorder(BorderFactory.createEtchedBorder());
351
352   ImageIcon copy_icon =createImageIcon("icons/copy.gif");
353   copy_icon_button = new JButton(copy_icon);
354   copy_icon_button.addActionListener(this);
355   copy_icon_button.setBorder(BorderFactory.createEtchedBorder());
356
357   paste_icon = createImageIcon("icons/paste.gif");
358   paste_icon_button = new JButton(paste_icon);
359   paste_icon_button.addActionListener(this);
360   paste_icon_button.setBorder(BorderFactory.createEtchedBorder());
361
362   ImageIcon print_icon = createImageIcon("icons/print.gif");
363   print_icon_button = new JButton(print_icon);
364   print_icon_button.addActionListener(this);
365   print_icon_button.addMouseListener (this);
366   print_icon_button.setEnabled(true);
367   print_icon_button.setBorder(BorderFactory.createEtchedBorder());
368
369   ImageIcon java_icon = createImageIcon("icons/java.gif");
370   java_icon_button = new JButton(java_icon);
371   java_icon_button.addActionListener(this);
372   java_icon_button.setBorder(BorderFactory.createEtchedBorder());
373
374   ImageIcon help_icon = createImageIcon("icons/info.gif");
375   help_icon_button = new JButton(help_icon);
376   help_icon_button.addActionListener(this);
377   help_icon_button.setBorder(BorderFactory.createEtchedBorder());
378
379   JPanel icons_panel_1 = new JPanel(new FlowLayout());
380   icons_panel_1.add(new_icon_button);
```

```
381   icons_panel_1.add(open_icon_button);
382   icons_panel_1.add(save_icon_button);
383   icons_panel_1.add(copy_icon_button);
384   icons_panel_1.add(paste_icon_button);
385   icons_panel_1.add(print_icon_button);
386   icons_panel_1.add(java_icon_button);
387   icons_panel_1.add(help_icon_button);
388
389   icons_panel_1.setBorder(BorderFactory.createEtchedBorder());
390   icons_panel_1.setBorder(BorderFactory.createLoweredBevelBorder());
391
392   JPanel icons_panel_2 = new JPanel(new FlowLayout());
393
394
395   icons_panel = new JPanel(new BorderLayout());
396   icons_panel.add(icons_panel_1,BorderLayout.WEST);
397   icons_panel.add(icons_panel_2,BorderLayout.EAST);
398
399   start = new JButton("Verify your input data");
400   start.setFont(menuFont);
401   start.setBackground(ccc);
402   start.addActionListener(this);
403   start.setBorder(BorderFactory.createEtchedBorder());
404
405
406   clear = new JButton(" Clear ");
407   clear.setFont(menuFont);
408   clear.setBackground(ccc);
409   clear.addActionListener(this);
410   clear.setBorder(BorderFactory.createEtchedBorder());
411
412   g_roup = new ButtonGroup();
413
414   euro = new JRadioButton("EURO",false);
415   euro.setFont(menuFont);
416   euro.setBorder(BorderFactory.createEtchedBorder());
417   euro.addItemListener(this);
418
419   nok = new JRadioButton("NOK",false);
420   nok.setFont(menuFont);
421   nok.setBorder(BorderFactory.createEtchedBorder());
422   nok.addItemListener(this);
423
424   yen = new JRadioButton("YEN",false);
425   yen.setFont(menuFont);
426   yen.setBorder(BorderFactory.createEtchedBorder());
427   yen.addItemListener(this);
428
429   g_roup.add(yen);
430   g_roup.add(nok);
431   g_roup.add(euro);
432
433   JLabel mylabel = new JLabel(" Created by Adil Gebory March 2003 ");
434
435   Font myFont = new Font("mylabel",Font.PLAIN,10);
436   mylabel.setFont(myFont);
437
438   myLabelButton = new JButton();
439   myLabelButton.addActionListener(this);
440   myLabelButton.add(mylabel);
441   myLabelButton.setBackground(cccc);
442   myLabelButton.setBorder(BorderFactory.createEtchedBorder());
443
444   cb_panel = new JPanel(new FlowLayout());
445
446   cb_panel.setBorder(BorderFactory.createLineBorder(Color.green));
447   cb_panel.setBorder(BorderFactory.createEtchedBorder());
448
449   cb_panel.add(yen);
450   cb_panel.add(nok);
451   cb_panel.add(euro);
452   cb_panel.add(clear);
453   cb_panel.add(start);
454   cb_panel.add(myLabelButton);
455
456   JButton cb_button = new JButton();
```

```
457   cb_button.add(cb_panel);
458
459   JLabel marketInfoLabel = new JLabel("Potential Market Segment");
460   JLabel xLabel = new JLabel(" ");
461
462   init_subLabel= new JLabel("Initial iMODE {CUST*Million}");
463   init_subLabel.setFont(menuFont);
464
465   JComboBox init_sub = new JComboBox();
466   init_sub.addActionListener(this);
467   init_sub.addItem("0");
468   init_sub.addItem("1");
469   init_sub.addItem("5");
470   init_sub.addItem("10");
471   init_sub.addItem("15");
472   init_sub.addItem("20");
473   init_sub.setBackground(ccc);
474
475   womLabel= new JLabel("Contact rate{CUST/CUST/MO}");
476   womLabel.setFont(menuFont);
477
478   JComboBox wom = new JComboBox();
479   wom.addActionListener(this);
480   wom.addItem("1");
481   wom.addItem("1.5");
482   wom.addItem("3");
483   wom.setBackground(ccc);
484
485   lossLabel= new JLabel("Market Size{CUST*Million}");
486   lossLabel.setFont(menuFont);
487
488   JComboBox loss = new JComboBox();
489   loss.addActionListener(this);
490   loss.addItem("40");
491   loss.addItem("45");
492   loss.addItem("50");
493   loss.addItem("55");
494   loss.addItem("60");
495   loss.setBackground(ccc);
496
497
498   lossPanel = new JPanel(new GridLayout(4,1,0,20));
499   lossPanel.add(marketInfoLabel);
500   lossPanel.add(init_subLabel);
501   lossPanel.add(womLabel);
502   lossPanel.add(lossLabel);
503
504
505   womPanel = new JPanel(new GridLayout(4,1,0,20));
506   womPanel.add(xLabel);
507   womPanel.add(init_sub);
508   womPanel.add(wom);
509   womPanel.add(loss);
510
511
512   data = new JPanel (new FlowLayout(20));
513   data.add(lossPanel);
514   data.add(womPanel);
515  //************************************
516
517   JLabel pricingInfoLabel= new JLabel("Pricing Segment");
518   //minPriceLabel.setFont(menuFont);
519   JLabel minPriceLabel= new JLabel("Min. packet price {JPY/PACKET}");
520   minPriceLabel.setFont(menuFont);
521   JLabel maxPriceLabel= new JLabel("Max. packet price {JPY/PACKET}");
522   maxPriceLabel.setFont(menuFont);
523   JLabel fixedPriceLabel= new JLabel("Subscription Fee {JPY/MONTH}");
524   fixedPriceLabel.setFont(menuFont);
525
526   JComboBox minPrice = new JComboBox();
527   minPrice.addActionListener(this);
528   minPrice.addItem("0.1");
529   minPrice.addItem("0.2");
530   minPrice.addItem("0.3");
531   minPrice.setBackground(ccc);
532
```

```
533   JComboBox maxPrice = new JComboBox();
534   maxPrice.addActionListener(this);
535   maxPrice.addItem("0.4");
536   maxPrice.addItem("0.5");
537   maxPrice.addItem("0.6");
538   maxPrice.setBackground(ccc);
539
540   JComboBox fixedPrice = new JComboBox();
541   fixedPrice .addActionListener(this);
542   fixedPrice .addItem("100");
543   fixedPrice .addItem("200");
544   fixedPrice .addItem("300");
545   fixedPrice.setBackground(ccc);
546
547
548   JPanel pricingLabelPanel = new JPanel(new GridLayout(4,1,0,20));
549   pricingLabelPanel.add(pricingInfoLabel);
550   pricingLabelPanel.add(minPriceLabel);
551   pricingLabelPanel.add(maxPriceLabel);
552   pricingLabelPanel.add(fixedPriceLabel);
553
554           JLabel xxLabel = new JLabel(" ");
555   JPanel pricingFieldPanel = new JPanel(new GridLayout(4,1,0,20));
556   pricingFieldPanel.add(xxLabel);
557   pricingFieldPanel.add(minPrice);
558   pricingFieldPanel.add(maxPrice);
559   pricingFieldPanel.add(fixedPrice);
560
561   JPanel price_data = new JPanel (new FlowLayout(20));
562   price_data.add(pricingLabelPanel);
563   price_data.add(pricingFieldPanel);
564
565   JButton data_price_button = new JButton("");
566   data_price_button.add(price_data);
567   //*****************************************
568
569   JLabel budgetCostsLabel= new JLabel("Manual Budget & Fixed Costs ");
570
571   JLabel budgetLabel= new JLabel("Manual Budget {Million JPY/Month}");
572   budgetLabel.setFont(menuFont);
573
574   JLabel costsLabel= new JLabel("Fixed Costs {Billion JPY/Month}");
575   costsLabel.setFont(menuFont);
576
577   JComboBox manual = new JComboBox();
578   manual .addActionListener(this);
579   manual .addItem("20");
580   manual .addItem("40");
581   manual .addItem("60");
582   manual.setBackground(ccc);
583
584   JComboBox costs = new JComboBox();
585   costs .addActionListener(this);
586   costs .addItem("1");
587   costs .addItem("5");
588   costs .addItem("10");
589   costs.setBackground(ccc);
590
591   JPanel fractionLabelPanel = new JPanel(new GridLayout(3,1,0,20));
592   fractionLabelPanel.add(budgetCostsLabel);
593   fractionLabelPanel.add(budgetLabel);
594   fractionLabelPanel.add(costsLabel);
595
596    JLabel xxxLabel = new JLabel(" ");
597   JPanel fractionFieldPanel = new JPanel(new GridLayout(3,1,0,20));
598   fractionFieldPanel.add(xxxLabel);
599   fractionFieldPanel.add(manual);
600   fractionFieldPanel.add(costs);
601
602   JPanel fraction_data = new JPanel (new FlowLayout(20));
603   fraction_data.add(fractionLabelPanel);
604   fraction_data.add(fractionFieldPanel);
605
606   JButton data_fraction_button = new JButton("");
607   data_fraction_button.add(fraction_data);
608
```

```
609
610
611   //*****************************************
612   JButton data_button = new JButton("");
613   data_button.add(data);
614
615   //moroTextArea = new JTextArea(8,20);
616   //moroTextArea.setBackground(Color.lightGray);
617   //moroTextArea.setEditable(false);
618
619   //moroPane = new JScrollPane(moroTextArea);
620   //moroPane.setBorder(BorderFactory.createEtchedBorder());
621
622   JPanel commonPanel = new JPanel (new GridLayout(1,2,20,20));
623   commonPanel.add(data_price_button);
624   commonPanel.add(data_fraction_button);
625
626   JPanel newP = new JPanel(new BorderLayout());
627   newP.setBorder(BorderFactory.createLineBorder(Color.green));
628
629   newP.add(data_button,BorderLayout.WEST);
630   newP.add(commonPanel,BorderLayout.EAST);
631   newP.add(cb_button,BorderLayout.SOUTH);
632
633   introd_panel = new JPanel(new BorderLayout());
634   introd_panel.add(displayPane,BorderLayout.EAST);
635   introd_panel.add(paneTree,BorderLayout.WEST);
636   introd_panel.add(newP,BorderLayout.SOUTH);
637
638   JPanel menuPanel = new JPanel(new GridLayout(2,1));
639   menuPanel.add(mb);
640   menuPanel.add(icons_panel);
641   getContentPane().add(menuPanel,BorderLayout.NORTH);
642   getContentPane().add(introd_panel,BorderLayout.CENTER);
643   msg = new Message();
644   warning = new Warning();
645   verifyingMessage = new VerifyingMessage(this);
646
647   imodUsersTable = new ImodeUsers();
648   revenue = new Revenue();
649   badURL = false;
650
651   JDialog.setDefaultLookAndFeelDecorated(true);
652   fc = new JFileChooser();
653  }
654 //*****************************************************
655   protected ImageIcon createImageIcon(String path)
656   {
657      java.net.URL imgURL = TabbedPaneModel.class.getResource(path);
658      if (imgURL != null)
659      {
660         return new ImageIcon(imgURL);
661      }
662      else
663      {
664         text.setText("Couldn't find file: " + path);
665         return null;
666
667             }
668  }
669 //*****************************************************
670  public void actionPerformed(ActionEvent ae)
671  {
672   if(ae.getSource() == start)
673   {
674      try
675      {
676            getInputData(text);
677            verifyingMessage.setVisible(true);
678            //showVerifyingMessage();
679      }
680      catch(Exception e)
681      {
682            text.setText("Something wrong, check and try again!");
683            //showVerifyingMessage();
684      }
```

```
685  }
686
687  else if(ae.getSource() == myLabelButton)
688  {
689     ownerInfo(text);
690  }
691
692  else if(ae.getSource() == exit)
693  {
694    setVisible(false);
695  }
696
697  else if(ae.getSource() == about)
698  {
699     showHelpMessage();
700  }
701  else if(ae.getSource() == clear)
702  {
703     init_sub.setText("");
704     wom.setText("");
705     loss.setText("");
706     text.setText("");
707     text.setText("");
708  }
709  else if(ae.getSource() == print)
710  {
711  }
712
713  else if(ae.getSource() == revenueR)
714  {
715     showRevenueTable();
716  }
717
718  else if(ae.getSource() == imodeUsers)
719  {
720     showImodeUsersTable();
721  }
722
723  else if(ae.getSource() == help_icon_button)
724  {
725        showHelpMessage();
726  }
727
728  else if(ae.getSource() == java_icon_button)
729  {
730        try
731        {
732           URL sunURL = new URL("http://java.sun.com");
733        }
734        catch(MalformedURLException e)
735        {
736        }
737  }
738
739  else if(ae.getSource() == new_icon_button )
740  {
741        openNewFile();
742  }
743    else if(ae.getSource() == new_ )
744  {
745        openNewFile();
746  }
747
748  else if(ae.getSource() == open)
749  {
750     openFile();
751  }
752
753  else if(ae.getSource() == open_icon_button)
754  {
755     openFile();
756  }
757
758  else if(ae.getSource() == saveItem)
759  {
760     saveFile();
```

```
761   }
762
763   else if(ae.getSource() == save_icon_button)
764   {
765      saveFile();
766   }
767
768   else if(ae.getSource() == showBlackBoard)
769   {
770      newBlackBoard();
771   }
772
773   else if(ae.getSource() == copy)
774   {
775   }
776
777   else if(ae.getSource() == modelItem)
778   {
779            showModelSlides();
780   }
781
782   else if(ae.getSource() == iMODEJava)
783   {
784            String filename = "codes/iMODE.java";
785            JTextArea mytextArea = new JTextArea();
786      myViewer.loadSourceCode(myViewer,filename,mytextArea);
787      vp.add(mytextArea);
788      text.setText("File: "+ filename);
789   }
790
791   else if(ae.getSource() == messageJava)
792   {
793            String filename = "codes/Message.java";
794            JTextArea mytextArea = new JTextArea();
795      myViewer.loadSourceCode(myViewer,filename,mytextArea);
796      vp.add(mytextArea);
797      text.setText("File: "+ filename);
798   }
799
800   else if(ae.getSource() == GUIItem)
801   {
802            showGUIScreenPictures();
803   }
804
805   else if(ae.getSource() == graphItem)
806   {
807            showGraphTabbed();
808   }
809
810
811 }// end of actionPerformed method
812
813 //*************************************************************
814      public void itemStateChanged(ItemEvent e)
815      {
816            if(e.getItem()== yen)
817                  text.setText("Your currency is YEN");
818            else if(e.getItem()== euro)
819                  text.setText("Your currencu is EURO");
820            else
821                  text.setText("Your currencu is NOK");
822      }
823
824
825 //*************************************************************
826
827      public void selectedNode()
828      {
829
830            selectedPath = tree.getSelectionPath();
831            DefaultMutableTreeNode selectedNode = (DefaultMutableTreeNode)
832                        selectedPath.getLastPathComponent();
833
834      }
835
836 //*************************************************
```

```
837
838     public void fillInnTextArea()
839     {
840     }
841
842 //*****************************************************
843     public void homeSite()
844     {
845     }
846 //*****************************************************
847
848     public void mouseClicked(MouseEvent me)
849     {
850
851         int selRow =tree.getClosestRowForLocation(me.getX(),me.getY());
852         int selNextRow =tree.getLeadSelectionRow();
853         TreePath selPath = tree.getPathForLocation(me.getX(),me.getY());
854         tree.fireTreeExpanded(selPath);
855
856
857
858         if(selRow == 0)
859         {
860          String p = "/html/introduction.html";
861          HtmlDemo(p);
862         }
863
864         else if(selRow == 1)
865         {
866          String p = "/html/problem_description.html";
867          HtmlDemo(p);
868         }
869
870         else if(selRow == 2)
871         {
872          String p = "/html/imode_services.html";
873          HtmlDemo(p);
874         }
875         else if(selRow == 3)
876         {
877          String p = "/html/tech_perspective.html";
878          HtmlDemo(p);
879         }
880         else if(selRow == 4)
881         {
882          String p = "/html/imode_versus_wap.html";
883          HtmlDemo(p);
884         }
885         else if(selRow == 5)
886         {
887                     String p = "/html/imode_success.html";
888          HtmlDemo(p);
889         }
890         else if(selRow == 6)
891         {
892          String p = "/html/success_factors.html";
893          HtmlDemo(p);
894         }
895         else if(selRow == 7)
896         {
897          String p = "/html/purpose.html";
898          HtmlDemo(p);
899         }
900         else if(selRow == 8)
901         {
902          String p = "/html/reference_behaviour.html";
903          HtmlDemo(p);
904         }
905         else if(selRow == 9)
906         {
907          String p = "/html/iu_ref_behaviour.html";
908          HtmlDemo(p);
909         }
910         else if(selRow == 10)
911         {
912          String p = "/html/usage_ref_behaviour.html";
```

```
913             HtmlDemo(p);
914            }
915            else if(selRow == 18)
916            {
917             String p = "/html/resources.html";
918             HtmlDemo(p);
919            }
920      }
921      public void mousePressed(MouseEvent mp)
922      {
923
924      }
925      public void mouseReleased(MouseEvent mr)
926      {
927
928      }
929      public void mouseEntered(MouseEvent ment)
930      {
931            if(ment.getSource() == print_icon_button)
932            {
933                  print_icon_button.setEnabled(false);
934            }
935
936      }
937      public void mouseExited(MouseEvent mexit)
938      {
939            if(mexit.getSource() == print_icon_button)
940            {
941                  print_icon_button.setEnabled(true);
942            }
943
944      }
945
946 //*************************************************
947
948      protected URL getURL(String filename)
949      {
950            URL codeBase = getCodeBase();
951            URL url = null;
952
953            try
954            {
955                  url = new URL(codeBase, filename);
956            }
957            catch (java.net.MalformedURLException e)
958            {
959                  text.setText("Couldn't create connection: " + "badly specified URL");
960                  return null;
961      }
962
963      return url;
964   }
965 //*****************************************************************************
966
967      public void getInputData(JTextArea txt)
968      {
969
970        try
971        {
972            int pc = Integer.parseInt(init_sub.getText());
973            Double contact_rate = Double.valueOf(wom.getText());
974            int market_size = Integer.parseInt(loss.getText());
975
976            if(String.valueOf(init_sub.getText()) == null)
977            {
978                  txt.setText("");
979                  txt.append("Do it better!");
980            }
981            else
982            {
983                  txt.setText("");
984                  txt.append("Your input are:\n");
985                  txt.append("----------------\n");
986                  txt.append("Initial Potential Customers = \t"+ pc + "M customers\n");
987                  txt.append("WOM Contact Rate = \t"+ contact_rate + " %\n");
988                  txt.append("Market Size =\t\t"+ market_size+"M customers");
```

```
            }
        }
        catch(Exception e)
        {
            txt.setText("");
            txt.setText("Somthing wrong, do it again!");
        }

    }
//*************************************************************************
    public void ownerInfo(JTextArea txt)
    {
            txt.setText("");
            txt.append("Name:\tAdil Gebory\n");
            txt.append("Yrke;\tSenior Excutive Officer\n");
            txt.append("Email:\taalgeb99@siving.hia.no\n");
            txt.append("Address:\tMaria Dehlis vei 59\n");
            txt.append("\t1084 Oslo\n");
            txt.append("Private:\t+47 221 61 944\n");
            txt.append("Mobile:\t+47 957 60 187 Oslo");
    }
//*************************************************************************

  public void modelSite()
  {
  }

  public void printDocument()
  {

  }
  //************************************************

    protected Component makeTextPanel(String text)
   {
      JPanel panel = new JPanel(false);

      JLabel filler = new JLabel(text);
      filler.setHorizontalAlignment(JLabel.CENTER);
      panel.setLayout(new GridLayout(1, 2));
      panel.add(filler);

      return panel;
   }

   //*********************************************

    protected Component makeImagePanel()
    {
      JPanel panel = new JPanel(false);

      JButton b = new JButton("Helo me");
      panel.setLayout(new GridLayout(1, 1));
      panel.add(b);

      return panel;
   }
   //*********************************************

    public void openFile()
    {
        try
        {
            int returnVal = fc.showOpenDialog(this);

            if (returnVal == JFileChooser.APPROVE_OPTION)
            {
                File file = fc.getSelectedFile();
            }
            else
            {
                //text.append("Open command cancelled by user." + newline);
            }
        }
```

```
1065            catch(Exception e)
1066            {
1067                  text.setText("Feil: " + e);
1068            }
1069
1070      }
1071  //*************************************************************
1072
1073
1074      public void saveFile()
1075      {
1076            try
1077            {
1078
1079                  int returnVal = fc.showSaveDialog(this);
1080                  if (returnVal == JFileChooser.APPROVE_OPTION)
1081                  {
1082                        File file = fc.getSelectedFile();
1083
1084                  }
1085                  else
1086                  {
1087                        //text.append("Save command cancelled by user." + newline);
1088                  }
1089            }
1090            catch(Exception e)
1091            {
1092                  text.setText("Feil: " + e);
1093            }
1094      }
1095
1096  //*************************************************
1097             protected void showHelpMessage()
1098             {
1099                        JFrame.setDefaultLookAndFeelDecorated(true);
1100                        JFrame frame = new JFrame("IMODE SUCCESS IN JAPAN BY SD");
1101             frame.getContentPane().add(msg,BorderLayout.CENTER);
1102             frame.setSize(300, 300);
1103             frame.setVisible(true);
1104             frame.pack();
1105             }
1106
1107             protected void showVerifyingMessage()
1108             {
1109                        JFrame.setDefaultLookAndFeelDecorated(true);
1110                        JFrame frame = new JFrame("Verifying Message");
1111             frame.getContentPane().add(verifyingMessage,BorderLayout.CENTER);
1112             frame.setSize(300, 300);
1113
1114             frame.setVisible(true);
1115             frame.pack();
1116
1117             }
1118
1119             protected void showWarningMessage()
1120             {
1121                        JFrame.setDefaultLookAndFeelDecorated(true);
1122                        JFrame frame = new JFrame("Warning");
1123             frame.getContentPane().add(warning,BorderLayout.CENTER);
1124             frame.setSize(300, 300);
1125             frame.setVisible(true);
1126             frame.pack();
1127             }
1128
1129      protected void openNewFile()
1130      {
1131
1132            JFileChooser chooser = new JFileChooser();
1133            chooser.setApproveButtonText("Save");
1134            chooser.showDialog(this, null);
1135            chooser.setVisible(true);
1136             }
1137
1138             protected void newBlackBoard()
1139             {
1140                        JFrame.setDefaultLookAndFeelDecorated(true);
```

```
1141                    JTextArea mytextArea = new JTextArea();
1142            JFrame frame = new JFrame("Write your comments here, It is just for your own use");
1143            frame.getContentPane().add(mytextArea,BorderLayout.CENTER);
1144            frame.setSize(500,500);
1145            frame.setVisible(true);
1146    }
1147
1148            protected void showGraphTabbed()
1149            {
1150                    JFrame.setDefaultLookAndFeelDecorated(true);
1151                    JFrame frame = new JFrame("iMODE GRAPHS");
1152            frame.getContentPane().add(graphTabbedPane,BorderLayout.CENTER);
1153            frame.setSize(800,500);
1154            frame.setVisible(true);
1155            frame.pack();
1156            }
1157
1158            protected void showGUIScreenPictures()
1159            {
1160                    JFrame.setDefaultLookAndFeelDecorated(true);
1161                    JFrame frame = new JFrame("Graphical User Interface Screen pictures");
1162            frame.getContentPane().add(tabbedPaneGUI,BorderLayout.CENTER);
1163            frame.setSize(800,500);
1164            frame.setVisible(true);
1165            frame.pack();
1166            }
1167
1168            protected void showModelSlides()
1169            {
1170                    JFrame.setDefaultLookAndFeelDecorated(true);
1171                    JFrame frame = new JFrame("Pictures from the Model in PowerSim Studio");
1172            frame.getContentPane().add(modelTabbedPane,BorderLayout.CENTER);
1173            frame.setSize(800,500);
1174            frame.setVisible(true);
1175            frame.pack();
1176            }
1177
1178
1179            protected void showImodeUsersTable()
1180            {
1181                    JFrame.setDefaultLookAndFeelDecorated(true);
1182                    JFrame frame = new JFrame("IMODE USERS TABLE");
1183            frame.getContentPane().add(imodUsersTable,BorderLayout.CENTER);
1184            frame.setSize(350, 550);
1185            frame.setVisible(true);
1186            frame.pack();
1187            }
1188
1189            protected void showRevenueTable()
1190            {
1191                    JFrame.setDefaultLookAndFeelDecorated(true);
1192                    JFrame frame = new JFrame("REVENUE TABLE");
1193            frame.getContentPane().add(revenue,BorderLayout.CENTER);
1194            frame.setSize(350, 550);
1195            frame.setVisible(true);
1196            frame.pack();
1197            }
1198
1199            protected void showAnimationPanel()
1200            {
1201
1202            }
1203
1204  protected void HtmlDemo(String path)
1205  {
1206    try {
1207                URL url = null;
1208                try
1209                {
1210                        url = getClass().getResource(path);
1211      }
1212      catch (Exception e)
1213      {
1214                        text.setText("Failed to open " + path);
1215                        url = null;
1216      }
```

```
1217
1218        if(url != null)
1219        {
1220           html = new JEditorPane(url);
1221           html.setEditable(false);
1222           html.addHyperlinkListener(createHyperLinkListener());
1223                       vp.add(html);
1224        }
1225     }
1226     catch (MalformedURLException e)
1227     {
1228        text.setText("Malformed URL: " + e);
1229     }
1230     catch (IOException e)
1231     {
1232        text.setText("IOException: " + e);
1233     }
1234  }
1235
1236
1237  public HyperlinkListener createHyperLinkListener()
1238  {
1239           return new HyperlinkListener()
1240           {
1241              public void hyperlinkUpdate(HyperlinkEvent e)
1242              {
1243                       if (e.getEventType() == HyperlinkEvent.EventType.ACTIVATED)
1244                       {
1245                          if (e instanceof HTMLFrameHyperlinkEvent)
1246                          {
1247
     ((HTMLDocument)html.getDocument()).processHTMLFrameHyperlinkEvent((HTMLFrameHyperlinkEvent)e);
1248
1249                          }
1250                          else
1251                          {
1252                                   try
1253                                   {
1254                                      html.setPage(e.getURL());
1255                                   } catch (IOException ioe)
1256                                   {
1257                                      text.setText("IOE: " + ioe);
1258                                   }
1259                          }
1260                       }
1261              }
1262           };
1263  }
1264
1265  public void hyperlinkUpdate(HyperlinkEvent e)
1266  {}
1267
1268
1269
1270 }//end of class Imode
1271 //*************************************************************
1272
1273  class MyTextArea extends JTextArea
1274  {
1275     public MyTextArea()
1276     {
1277        super(null, 15, 66);
1278        setEditable(false);
1279        setText("");
1280     }
1281
1282     public float getAlignmentX ()
1283     {
1284        return LEFT_ALIGNMENT;
1285     }
1286
1287     public float getAlignmentY ()
1288     {
1289        return TOP_ALIGNMENT;
1290     }
1291  }
```

// TabbedPaneModel.java

```
1 import javax.swing.JTabbedPane;
2 import javax.swing.ImageIcon;
3 import javax.swing.JLabel;
4 import javax.swing.JPanel;
5 import javax.swing.JFrame;
6 import javax.swing.*;
7 import java.awt.*;
8 import java.awt.event.*;
9
10 public class TabbedPaneModel extends JPanel implements ActionListener
11 {
12    public TabbedPaneModel()
13    {
14       ImageIcon icon = createImageIcon("images/open.gif");
15       Font myFont = new Font("tabbedPane",Font.PLAIN,11);
16       JTabbedPane tabbedPane = new JTabbedPane();
17       tabbedPane.setFont(myFont);
18
19       ImageIcon iconMarket = createImageIcon("images_model/model_00.gif");
20       ImageIcon iconAdv = createImageIcon("images_model/model_01.gif");
21       ImageIcon iconAttract = createImageIcon("images_model/model_02.gif");
22       ImageIcon iconRevenue = createImageIcon("images_model/model_03.gif");
23       ImageIcon iconCosts = createImageIcon("images_model/model_04.gif");
24       ImageIcon iconPrice = createImageIcon("images_model/model_05.gif");
25       ImageIcon iconAtt_2 = createImageIcon("images_model/model_06.gif");
26
27       JButton panel1 = new JButton(iconMarket);
28       JButton panel2 = new JButton(iconAdv);
29       JButton panel3 = new JButton(iconAttract);
30       JButton panel4 = new JButton(iconRevenue);
31       JButton panel5 = new JButton(iconCosts);
32       JButton panel6 = new JButton(iconPrice);
33       JButton panel7 = new JButton(iconAtt_2);
34
35       tabbedPane.addTab("Market Growth",icon,panel1);
36       tabbedPane.setSelectedIndex(0);
37       tabbedPane.addTab("Advertising",icon,panel2);
38       tabbedPane.addTab("Attarctiveness Strategy",icon,panel3);
39       tabbedPane.addTab("Revenue",icon,panel4);
40       tabbedPane.addTab("Costs & Income",icon,panel5);
41       tabbedPane.addTab("Pricing",icon,panel6);
42       tabbedPane.addTab("Attarctiveness Costs",icon,panel7);
43
44       setLayout(new GridLayout(1,1));
45       add(tabbedPane);
46    }
47
48    protected Component makeTextPanel(String text)
49    {
50       ImageIcon iconIntro = createImageIcon("images/home.gif");
51       JButton introButton = new JButton(iconIntro);
52       JLabel filler = new JLabel(text);
53       JPanel panel = new JPanel();
54       panel.setLayout(new GridLayout(1, 1));
55       panel.add(introButton);
56       return panel;
57    }
58
59     protected Component makeImagePanel(ImageIcon imgicon)
60    {
61       JPanel panel = new JPanel(false);
62       JButton b = new JButton(imgicon);
63       panel.setLayout(new GridLayout(1, 1));
64       panel.add(b);
65       return panel;
66    }
67
68    protected static ImageIcon createImageIcon(String path)
69    {
70       java.net.URL imgURL = TabbedPaneModel.class.getResource(path);
71       if (imgURL != null)
72       {
73          return new ImageIcon(imgURL);
74       }
75       else
76       {
```

```
77          System.err.println("Couldn't find file: " + path);
78          return null;
79        }
80    }
81
82    public void actionPerformed(ActionEvent ae)
83   {
84   }
85 }
```

// TabbedPaneGUI.java

```
1 import javax.swing.JTabbedPane;
2 import javax.swing.ImageIcon;
3 import javax.swing.JLabel;
4 import javax.swing.JPanel;
5 import javax.swing.JFrame;
6 import javax.swing.*;
7 import java.awt.*;
8 import java.awt.event.*;
9
10 public class TabbedPaneGUI extends JPanel implements ActionListener
11 {
12    public TabbedPaneGUI()
13    {
14       ImageIcon icon = createImageIcon("images/open.gif");
15       JTabbedPane tabbedPane = new JTabbedPane(1);
16       Font myFont = new Font("tabbedPane",Font.PLAIN,11);
17       JPanel p = new JPanel();
18       p.setLayout(new BoxLayout(p,BoxLayout.Y_AXIS));
19       p.setAlignmentY(TOP_ALIGNMENT);
20       p.setAlignmentX(LEFT_ALIGNMENT);
21       ImageIcon icon00 = createImageIcon("images_report/rep_00.gif");
22       ImageIcon icon01 = createImageIcon("images_report/rep_01.gif");
23       ImageIcon icon02 = createImageIcon("images_report/rep_02.gif");
24       ImageIcon icon03 = createImageIcon("images_report/rep_03.gif");
25       ImageIcon icon04 = createImageIcon("images_report/rep_04.gif");
26       ImageIcon icon05 = createImageIcon("images_report/rep_05.gif");
27       JButton panel0 = new JButton(icon00);
28       JButton panel1 = new JButton(icon01);
29       JButton panel2 = new JButton(icon02);
30       JButton panel3 = new JButton(icon03);
31       JButton panel4 = new JButton(icon04);
32       JButton panel5 = new JButton(icon05);
33       tabbedPane.addTab("WAP&IMODE",icon,panel0,"Nothing to do");
34       tabbedPane.setSelectedIndex(0);
35       tabbedPane.setFont(myFont);
36       tabbedPane.addTab("Referance",icon,panel1);
37       tabbedPane.addTab("Potential Customers",icon,panel2);
38       tabbedPane.addTab("Advertising)",icon,panel3);
39       tabbedPane.addTab("WOM&Advertising",icon,panel4);
40       tabbedPane.addTab("Non IMODE users",icon,panel5);
41       setLayout(new GridLayout(1,1));
42       add(tabbedPane);
43    }
44
45    protected Component makeTextPanel(String text)
46    {
47       ImageIcon iconIntro = createImageIcon("images/home.gif");
48       JButton introButton = new JButton(iconIntro);
49       JLabel filler = new JLabel(text);
50       JPanel panel = new JPanel();
51       panel.setLayout(new GridLayout(1, 1));
52       panel.add(introButton);
53       return panel;
54    }
55
56     protected Component makeImagePanel(ImageIcon imgicon)
57    {
58       JPanel panel = new JPanel(false);
59       JButton b = new JButton(imgicon);
60       panel.setLayout(new GridLayout(1, 1));
61       panel.add(b);
62       return panel;
63    }
64
65    protected static ImageIcon createImageIcon(String path)
66    {
67       java.net.URL imgURL = TabbedPaneGUI.class.getResource(path);
68       if (imgURL != null)
69       {
70          return new ImageIcon(imgURL);
71       }
72       else
73       {
74          System.err.println("Couldn't find file: " + path);
75          return null;
76       }
77    }
78
79    public void actionPerformed(ActionEvent ae)
80   {
81   }
82 }
```

// TabbedPaneGraph.java

```
1 import javax.swing.JTabbedPane;
2 import javax.swing.ImageIcon;
3 import javax.swing.JLabel;
4 import javax.swing.JPanel;
5 import javax.swing.JFrame;
6 import javax.swing.*;
7 import java.awt.*;
8 import java.awt.event.*;
9
10 public class TabbedPaneGraph extends JPanel implements ActionListener
11 {
12    public TabbedPaneGraph()
13    {
14       ImageIcon icon = createImageIcon("images/open.gif");
15       Font myFont = new Font("tabbedPane",Font.PLAIN,11);
16       JTabbedPane tabbedPane = new JTabbedPane();
17       tabbedPane.setFont(myFont);
18       ImageIcon iconMarket = createImageIcon("images_graphs/graph_00.gif");
19       ImageIcon iconAR = createImageIcon("images_graphs/graph_01.gif");
20       ImageIcon iconCostRevenue = createImageIcon("images_graphs/graph_02.gif");
21       ImageIcon iconCostRevenueN = createImageIcon("images_graphs/graph_03.gif");
22       ImageIcon iconIncome = createImageIcon("images_graphs/graph_04.gif");
23       ImageIcon iconIncomeN = createImageIcon("images_graphs/graph_05.gif");
24       ImageIcon iconByAge = createImageIcon("images_graphs/graph_06.gif");
25       ImageIcon iconSimPrice = createImageIcon("images_graphs/graph_07.gif");
26
27       JButton panel1 = new JButton(iconMarket);
28       JButton panel2 = new JButton(iconAR);
29       JButton panel3 = new JButton(iconCostRevenue);
30       JButton panel4 = new JButton(iconCostRevenueN);
31       JButton panel5 = new JButton(iconIncome);
32       JButton panel6 = new JButton(iconIncomeN);
33       JButton panel7 = new JButton(iconByAge);
34       JButton panel8 = new JButton(iconSimPrice);
35
36       tabbedPane.addTab("Market Growth",icon,panel1);
37       tabbedPane.setSelectedIndex(0);
38       tabbedPane.addTab("Adoption Rate",icon,panel2);
39       tabbedPane.addTab("Finantial JPY",icon,panel3);
40       tabbedPane.addTab("Finantial NOK",icon,panel4);
41       tabbedPane.addTab("Income JPY",icon,panel5);
42       tabbedPane.addTab("Income NOK",icon,panel6);
43       tabbedPane.addTab("Users By Age",icon,panel7);
44       tabbedPane.addTab("Simulated Price",icon,panel8);
45
46       setLayout(new GridLayout(1,1));
47       add(tabbedPane);
48    }
49
50    protected Component makeTextPanel(String text)
51    {
52       ImageIcon iconIntro = createImageIcon("images/home.gif");
53       JButton introButton = new JButton(iconIntro);
54       JLabel filler = new JLabel(text);
55       JPanel panel = new JPanel();
56       panel.setLayout(new GridLayout(1, 1));
57       panel.add(introButton);
58       return panel;
59    }
60
61     protected Component makeImagePanel(ImageIcon imgicon)
62    {
63       JPanel panel = new JPanel(false);
64       JButton b = new JButton(imgicon);
65       panel.setLayout(new GridLayout(1, 1));
66       panel.add(b);
67       return panel;
68    }
69
70    protected static ImageIcon createImageIcon(String path)
71    {
72       java.net.URL imgURL = TabbedPaneGraph.class.getResource(path);
73       if (imgURL != null)
74       {
75          return new ImageIcon(imgURL);
76       }
```

```
77      else
78      {
79          System.err.println("Couldn't find file: " + path);
80          return null;
81      }
82    }
83
84    public void actionPerformed(ActionEvent ae)
85  {
86  }
87
88 }
```

// Revenue.java

```
1 import java.awt.event.*;
2 import java.applet.*;
3 import java.io.*;
4 import java.awt.*;
5 import java.text.*;
6 import java.util.*;
7 import java.net.*;
8 import javax.swing.*;
9 import javax.swing.border.*;
10 import java.awt.print.PrinterJob;
11
12
13 class Revenue extends JPanel
14 {
15         public Revenue()
16         {
17
18
19                 Font myFont = new Font("JLabel",Font.PLAIN,12);
20
21                 String data[][] = {{"22/09/1999","Services launched"},
22                                    {"22/09/2000","50000000"},
23                                    {"22/09/2001","60000000"},
24                                    {"22/09/2002","70000000"},
25                                    {"22/09/2003","80000000"},
26                                    {"22/09/2004","90000000"},
27                                    {"22/09/2005","100000000"},
28                                    {"22/09/2006","110000000"},
29                                    {"22/09/2008","120000000"}};
30                 String columnNames[] = {"Date","Rvenue * 1000"};
31
32
33                 JTable table = new JTable(data,columnNames);
34                 JScrollPane pane = JTable.createScrollPaneForTable(table);
35                 pane.setBorder(BorderFactory.createLoweredBevelBorder());
36                 setLayout(new GridLayout(1,1));
37                 add(pane);
38
39         }
40
41 }
42
```

// Message.java

```
1 import java.awt.event.*;
2 import java.applet.*;
3 import java.io.*;
4 import java.awt.*;
5 import java.text.*;
6 import java.util.*;
7 import java.net.*;
8 import javax.swing.*;
9
10 class Message extends JPanel implements ActionListener
11 {
12          JLabel l1,l2,l3,l4,l5,l6,l7;
13          JButton ok;
14          JPanel p1,p2,p;
15
16          public Message()
17          {
18                    Color c = Color.orange;
19                    l1 = new JLabel(" Imode success in Japan");
20                    l2 = new JLabel("\tCreated by");
21                    l3 = new JLabel("\tAdil Gebory         ");
22                    l4 = new JLabel("Maria Dehlis vei 59");
23                    l5 = new JLabel("1084 Oslo,Norway");
24                    l6 = new JLabel("Private:+47 221 61 944");
25                    l7 = new JLabel("Email:\taalgeb99@siving.hia.no");
26
27                    Font myFont = new Font("JLabel",Font.PLAIN,11);
28                    l1.setFont(myFont);
29                    l2.setFont(myFont);
30                    l3.setFont(myFont);
31                    l4.setFont(myFont);
32                    l5.setFont(myFont);
33                    l6.setFont(myFont);
34                    l7.setFont(myFont);
35
36                    ok = new JButton("Close");
37                    ok.setFont(myFont);
38                    ok.addActionListener(this);
39                    ok.setBorder(BorderFactory.createEtchedBorder());
40                    ok.setBackground(c);
41
42                    p = new JPanel(new FlowLayout());
43                    p.add(ok);
44
45                    JButton info_button = new JButton("");
46                    p1 = new JPanel(new GridLayout(7,1));
47                    p1.add(l1);
48                    p1.add(l2);
49                    p1.add(l3);
50                    p1.add(l4);
51                    p1.add(l5);
52                    p1.add(l6);
53                    p1.add(l7);
54
55                    info_button.add(p1);
56                    info_button.setBorder(BorderFactory.createEtchedBorder());
57
58                    String path = "images/02.gif";
59                    ImageIcon myIcon = createImageIcon(path);
60                    JButton myButton = new JButton(myIcon);
61
62                    p2 = new JPanel(new GridLayout(1,2));
63                    p2.add(myButton);
64                    p2.add(info_button);
65
66
67                    setLayout(new GridLayout(1,1));
68          add(p2);
69          }
70
71          protected ImageIcon createImageIcon(String path)
72          {
73          java.net.URL imgURL = TabbedPaneModel.class.getResource(path);
74          if (imgURL != null)
75          {
76                    return new ImageIcon(imgURL);
```

```
77          }
78          else
79          {
80                  return null;
81                  }
82          }
83
84          public void actionPerformed(ActionEvent ae)
85          {
86                  if(ae.getSource() == ok)
87                          setVisible(false);
88          }
89 }//End of class Message
90
```

// ImodeUsers.java

```
import java.awt.event.*;
import java.applet.*;
import java.io.*;
import java.awt.*;
import java.text.*;
import java.util.*;
import java.net.*;
import javax.swing.*;
import javax.swing.border.*;


class ImodeUsers extends JPanel
{

	public ImodeUsers()
	{

		Font myFont = new Font("JLabel",Font.PLAIN,12);
		String data[][] = {{"22/09/1999","Services launched"},
					{"28/06/1999","500000"},
					{"08/08/1999","1000000"},
					{"15/03/2000","5000000"},
					{"06/08/2000","10000000"},
					{"22/11/2000","15000000"},
					{"11/02/2001","19115000"},
					{"18/02/2001","19398000"},
					{"25/02/2001","19675000"},
					{"04/03/2001","20015000"},
					{"25/03/2001","21356000"},
					{"01/04/2001","21753000"},
					{"08/04/2001","22086000"},
					{"15/04/2001","22412000"},
					{"22/04/2001","22681000"},
					{"09/09/2001","27139000"},
					{"16/09/2001","27239000"},
					{"23/09/2001","27553000"},
					{"30/09/2001","27768000"},
					{"07/10/2001","27925000"},
					{"01/02/2002","31300000"},
					{"01/03/2002","32200000"},
					{"01/04/2002","32600000"},
					{"01/05/2002","33000000"},
					{"01/06/2002","33500000"},
					{"01/07/2002","34100000"},
					{"25/08/2002","34385000"},
					{"01/09/2002","34449000"},
					{"08/09/2002","34530000"}};
		String columnNames[] = {"Date","iMODE users {customers}"};


		JTable table = new JTable(data,columnNames);

		JScrollPane pane = JTable.createScrollPaneForTable(table);
		pane.setBorder(BorderFactory.createLineBorder(Color.green));
	pane.setBorder(BorderFactory.createLoweredBevelBorder());
		setLayout(new GridLayout(1,1));
		add(pane);
	}

}

```

// ImageDialog.java

```
1 import java.awt.*;
2 import java.awt.print.PrinterJob;
3 import java.awt.event.*;
4 import java.awt.Image;
5 import java.awt.FileDialog;
6 import java.net.*;
7 import java.net.URL;
8 import java.applet.*;
9 import javax.swing.*;
10 import javax.swing.tree.DefaultMutableTreeNode;
11 import javax.swing.tree.DefaultTreeModel;
12 import javax.swing.tree.TreePath;
13 import javax.swing.border.*;
14 import java.io.*;
15 import javax.print.*;
16 import javax.print.attribute.*;
17 import javax.print.attribute.standard.*;
18
19 class ImageDialog extends JDialog implements ActionListener
20 {
21         JLabel l1;
22         JButton ok;
23         JPanel p1,p2,p;
24
25
26         public ImageDialog()
27         {
28
29
30
31                 Color c = Color.orange;
32                 l1 = new JLabel(" Imode success in Japan");
33
34                 Font myFont = new Font("JLabel",Font.PLAIN,11);
35                 l1.setFont(myFont);
36
37                 ok = new JButton("Close");
38                 ok.setFont(myFont);
39                 ok.addActionListener(this);
40                 ok.setBorder(BorderFactory.createEtchedBorder());
41                 ok.setBackground(c);
42
43                 p = new JPanel(new FlowLayout());
44                 p.add(ok);
45
46
47                 JButton info_button = new JButton("");
48                 p1 = new JPanel(new FlowLayout());
49                 p1.add(l1);
50
51                 info_button.add(p1);
52
53                 info_button.setBorder(BorderFactory.createEtchedBorder());
54
55                 getContentPane().add("North",info_button);
56                 getContentPane().add("South",p);
57                 pack();
58         }
59
60         public void actionPerformed(ActionEvent ae)
61         {
62                 if(ae.getSource() == ok)
63                         setVisible(false);
64         }
65 }
66
```

// Warning.java

```
1 import java.awt.event.*;
2 import java.applet.*;
3 import java.io.*;
4 import java.awt.*;
5 import java.text.*;
6 import java.util.*;
7 import java.net.*;
8 import javax.swing.*;
9
10 class Warning extends JPanel implements ActionListener
11 {
12          JLabel l1;
13          JButton ok;
14          JPanel p1,p2,p;
15
16          public Warning()
17          {
18                    Font myFont = new Font("JLabel",Font.PLAIN,12);
19                    Color c = Color.orange;
20                    l1 = new JLabel(" Do not worry, Be happy! Every thing will be alright.");
21                    l1.setFont(myFont);
22                    ok = new JButton("Close");
23                    ok.setFont(myFont);
24                    ok.addActionListener(this);
25                    ok.setBorder(BorderFactory.createEtchedBorder());
26                    ok.setBackground(c);
27                    p = new JPanel(new FlowLayout());
28                    p.add(ok);
29                    JButton info_button = new JButton("");
30                    p1 = new JPanel(new GridLayout(1,1));
31                    p1.add(l1);
32                    info_button.add(p1);
33                    setLayout(new GridLayout(1,1));
34          add(info_button);
35          }
36
37          public void actionPerformed(ActionEvent ae)
38          {
39                    if(ae.getSource() == ok)
40                              setVisible(false);
41          }
42 }
43
```

// CodeViewer.java

```
1 import javax.swing.*;
2 import javax.swing.event.*;
3 import javax.swing.text.*;
4 import javax.swing.border.*;
5 import javax.swing.colorchooser.*;
6 import javax.swing.filechooser.*;
7 import javax.accessibility.*;
8
9 import java.awt.*;
10 import java.awt.event.*;
11 import java.beans.*;
12 import java.util.*;
13 import java.io.*;
14 import java.applet.*;
15 import java.net.*;
16
17 public class CodeViewer
18 {
19    private static HashMap reservedWords = new HashMap();
20    private boolean inMultiLineComment = false;
21    private String backgroundColor = "";
22    private String commentStart = "";
23    private String commentEnd = "";
24    private String stringStart = "";
25    private String stringEnd = "";
26    private String reservedWordStart = "";
27    private String reservedWordEnd = "";
28    String sourceCode;
29
30    static
31    {
32       loadHash();
33    }
34
35    public CodeViewer()
36    {
37    }
38
39    public void setCommentStart(String commentStart) {this.commentStart = commentStart;}
40    public void setCommentEnd(String commentEnd) {this.commentEnd = commentEnd;}
41    public void setStringStart(String stringStart) {this.stringStart = stringStart;}
42    public void setStringEnd(String stringEnd) {this.stringEnd = stringEnd;}
43    public void setReservedWordStart(String reservedWordStart) {this.reservedWordStart = reservedWordStart;}
44    public void setReservedWordEnd(String reservedWordEnd) {this.reservedWordEnd = reservedWordEnd;}
45    public String getCommentStart() {return commentStart;}
46    public String getCommentEnd() {return commentEnd;}
47    public String getStringStart() {return stringStart;}
48    public String getStringEnd() {return stringEnd;}
49    public String getReservedWordStart() {return reservedWordStart;}
50    public String getReservedWordEnd() {return reservedWordEnd;}
51    public String syntaxHighlight( String line ) {return htmlFilter(line);}
52
53    private String htmlFilter( String line )
54    {
55       if( line == null || line.equals("") )
56       {
57          return "";
58       }
59       line = replace(line, "&", "&#38;");
60       line = replace(line, "\\\\", "&#92;&#92;" );
61       line = replace(line, "" + (char)92 + (char)34, "&#92;&#34");
62       line = replace(line, "<", "&#60;");
63       line = replace(line, ">", "&#62;");
64       return multiLineCommentFilter(line);
65    }
66
67    private String multiLineCommentFilter(String line)
68    {
69       if (line == null || line.equals(""))
70       {
71          return "";
72       }
73       StringBuffer buf = new StringBuffer();
74       int index;
75       if (inMultiLineComment && (index = line.indexOf("*/")) > -1 && !isInsideString(line,index))
76       {
```

```
77          inMultiLineComment = false;
78          buf.append(line.substring(0,index));
79          buf.append("*/").append(commentEnd);
80          if (line.length() > index+2) {
81              buf.append(inlineCommentFilter(line.substring(index+2)));
82          }
83          return buf.toString();
84      }
85      else if (inMultiLineComment)
86      {
87          return line;
88      }
89      else if ((index = line.indexOf("/*")) > -1 && !isInsideString(line,index))
90      {
91          inMultiLineComment = true;
92          buf.append(inlineCommentFilter(line.substring(0,index)));
93          buf.append(commentStart).append("/*");
94          buf.append(multiLineCommentFilter(line.substring(index+2)));
95          return buf.toString();
96      }
97      else
98      {
99          return inlineCommentFilter(line);
100     }
101 }
102
103
104 private String inlineCommentFilter(String line)
105 {
106     if (line == null || line.equals(""))
107     {
108         return "";
109     }
110     StringBuffer buf = new StringBuffer();
111     int index;
112     if ((index = line.indexOf("//")) > -1 && !isInsideString(line,index))
113     {
114         buf.append(stringFilter(line.substring(0,index)));
115         buf.append(commentStart);
116         buf.append(line.substring(index));
117         buf.append(commentEnd);
118     }
119     else
120     {
121         buf.append(stringFilter(line));
122     }
123     return buf.toString();
124 }
125
126 private String stringFilter(String line)
127 {
128     if (line == null || line.equals(""))
129     {
130         return "";
131     }
132     StringBuffer buf = new StringBuffer();
133     if (line.indexOf("\"") <= -1) {
134         return keywordFilter(line);
135     }
136     int start = 0;
137     int startStringIndex = -1;
138     int endStringIndex = -1;
139     int tempIndex;
140     while ((tempIndex = line.indexOf("\"")) > -1) {
141         if (startStringIndex == -1) {
142             startStringIndex = 0;
143             buf.append( stringFilter(line.substring(start,tempIndex)) );
144             buf.append(stringStart).append("\"");
145             line = line.substring(tempIndex+1);
146         }
147         else {
148             startStringIndex = -1;
149             endStringIndex = tempIndex;
150             buf.append(line.substring(0,endStringIndex+1));
151             buf.append(stringEnd);
152             line = line.substring(endStringIndex+1);
```

```
153         }
154     }
155
156     buf.append( keywordFilter(line) );
157
158     return buf.toString();
159 }
160
161 private String keywordFilter( String line )
162 {
163     if( line == null || line.equals("") )
164     {
165         return "";
166     }
167     StringBuffer buf = new StringBuffer();
168     HashMap usedReservedWords = new HashMap();
169     int i=0, startAt=0;
170     char ch;
171     StringBuffer temp = new StringBuffer();
172     while( i < line.length() )
173     {
174         temp.setLength(0);
175         ch = line.charAt(i);
176         startAt = i;
177         while( i<line.length() && ( ( ch >= 65 && ch <= 90 )
178                 || ( ch >= 97 && ch <= 122 ) ) ) {
179             temp.append(ch);
180             i++;
181             if( i < line.length() )
182             {
183                 ch = line.charAt(i);
184             }
185         }
186         String tempString = temp.toString();
187         if( reservedWords.containsKey(tempString) && !usedReservedWords.containsKey(tempString))
188         {
189             usedReservedWords.put(tempString,tempString);
190             line = replace( line, tempString, (reservedWordStart+tempString+reservedWordEnd) );
191             i += (reservedWordStart.length() + reservedWordEnd.length());
192         }
193         else
194         {
195             i++;
196         }
197     }
198     buf.append(line);
199     return buf.toString();
200 }
201
202 private String replace( String line, String oldString, String newString )
203 {
204     int i=0;
205     while( ( i=line.indexOf( oldString, i ) ) >= 0 ) {
206         line = (new
StringBuffer().append(line.substring(0,i)).append(newString).append(line.substring(i+oldString.length()))).toString();
207         i += newString.length();
208     }
209     return line;
210 }
211
212 private boolean isInsideString(String line, int position)
213 {
214     if (line.indexOf("\"") < 0) {
215         return false;
216     }
217     int index;
218     String left = line.substring(0,position);
219     String right = line.substring(position);
220     int leftCount = 0;
221     int rightCount = 0;
222     while ((index = left.indexOf("\"")) > -1) {
223         leftCount ++;
224         left = left.substring(index+1);
225     }
226     while ((index = right.indexOf("\"")) > -1) {
227         rightCount ++;
```

```
228         right = right.substring(index+1);
229      }
230      if (rightCount % 2 != 0 && leftCount % 2 != 0) {
231         return true;
232      }
233      else {
234         return false;
235      }
236   }
237
238   private static void loadHash()
239   {
240      reservedWords.put( "abstract", "abstract" );
241      reservedWords.put( "do", "do" );
242      reservedWords.put( "inner", "inner" );
243      reservedWords.put( "public", "public" );
244      reservedWords.put( "var", "var" );
245      reservedWords.put( "boolean", "boolean" );
246      reservedWords.put( "continue", "continue" );
247      reservedWords.put( "int", "int" );
248      reservedWords.put( "return", "return" );
249      reservedWords.put( "void", "void" );
250      reservedWords.put( "break", "break" );
251      reservedWords.put( "else", "else" );
252      reservedWords.put( "interface", "interface" );
253      reservedWords.put( "short", "short" );
254      reservedWords.put( "volatile", "volatile" );
255      reservedWords.put( "byvalue", "byvalue" );
256      reservedWords.put( "extends", "extends" );
257      reservedWords.put( "long", "long" );
258      reservedWords.put( "static", "static" );
259      reservedWords.put( "while", "while" );
260      reservedWords.put( "case", "case" );
261      reservedWords.put( "final", "final" );
262      reservedWords.put( "naive", "naive" );
263      reservedWords.put( "super", "super" );
264      reservedWords.put( "transient", "transient" );
265      reservedWords.put( "cast", "cast" );
266      reservedWords.put( "float", "float" );
267      reservedWords.put( "new", "new" );
268      reservedWords.put( "rest", "rest" );
269      reservedWords.put( "catch", "catch" );
270      reservedWords.put( "for", "for" );
271      reservedWords.put( "null", "null" );
272      reservedWords.put( "synchronized", "synchronized" );
273      reservedWords.put( "char", "char" );
274      reservedWords.put( "finally", "finally" );
275      reservedWords.put( "operator", "operator" );
276      reservedWords.put( "this", "this" );
277      reservedWords.put( "class", "class" );
278      reservedWords.put( "generic", "generic" );
279      reservedWords.put( "outer", "outer" );
280      reservedWords.put( "switch", "switch" );
281      reservedWords.put( "const", "const" );
282      reservedWords.put( "goto", "goto" );
283      reservedWords.put( "package", "package" );
284      reservedWords.put( "throw", "throw" );
285      reservedWords.put( "double", "double" );
286      reservedWords.put( "if", "if" );
287      reservedWords.put( "private", "private" );
288      reservedWords.put( "true", "true" );
289      reservedWords.put( "default", "default" );
290      reservedWords.put( "import", "import" );
291      reservedWords.put( "protected", "protected" );
292      reservedWords.put( "try", "try" );
293   }
294
295
296      public void loadSourceCode(CodeViewer cv,String filename,JTextArea txt)
297   {
298               sourceCode = new String("");
299               char[] buff = new char[50000];
300               InputStream is;
301               InputStreamReader isr;
302               URL url;
303
```

```
304             try {
305                     url = getClass().getResource(filename);
306                     is = url.openStream();
307                     isr = new InputStreamReader(is);
308                     BufferedReader reader = new BufferedReader(isr);
309                     String line = reader.readLine();
310                     while(line != null) {
311                        sourceCode += cv.syntaxHighlight(line) + " \n ";
312                        line = reader.readLine();
313                        txt.setText(sourceCode);
314                     }
315                     sourceCode += new String("");
316             }
317         catch (Exception ex)
318         {
319            txt.setText("sourceCode = Could not load file: " + filename);
320         }
321
322   }
323 }//End of class CodeViewer
```

// VerifyingMessage.java

```
1 import java.awt.event.*;
2 import java.applet.*;
3 import java.io.*;
4 import java.awt.*;
5 import java.text.*;
6 import java.util.*;
7 import java.net.*;
8 import javax.swing.*;
9
10 class VerifyingMessage extends JDialog implements ActionListener
11 {
12          JLabel l1;
13          JButton ok,cancel;
14          JPanel p1,p2,p;
15          private Imode imode;
16
17          public VerifyingMessage(Imode i)
18          {
19                    imode = i;
20                    Color c = Color.orange;
21                    l1 = new JLabel(" Er you sure about your input data?");
22                    Font myFont = new Font("JLabel",Font.PLAIN,11);
23                    l1.setFont(myFont);
24                    ok = new JButton("  Yes  ");
25                    ok.setFont(myFont);
26                    ok.addActionListener(this);
27                    ok.setBorder(BorderFactory.createEtchedBorder());
28                    ok.setBackground(c);
29                    cancel = new JButton("  No  ");
30                    cancel.setFont(myFont);
31                    cancel.addActionListener(this);
32                    cancel.setBorder(BorderFactory.createEtchedBorder());
33                    cancel.setBackground(c);
34                    p = new JPanel(new FlowLayout());
35                    p.add(ok);
36                    p.add(cancel);
37                    JButton info_button = new JButton("");
38                    p1 = new JPanel(new FlowLayout());
39                    p1.add(l1);
40                    info_button.add(p1);
41                    info_button.setBorder(BorderFactory.createEtchedBorder());
42                    //setLayout(new GridLayout(2,1));
43          //add(info_button);
44          //add(p);
45          getContentPane().add(info_button,BorderLayout.CENTER);
46          getContentPane().add(p,BorderLayout.SOUTH);
47          setSize(300,300);
48          pack();
49          }
50
51          public void actionPerformed(ActionEvent ae)
52          {
53                    if(ae.getSource() == cancel)
54                    {
55                              setVisible(false);
56                              imode.text.setText("You can edit your input data");
57                    }
58                    else
59                    {
60                              setVisible(false);
61                              imode.text.setText("Your input data had been registered");
62                              imode.init_sub.setText("");
63                              imode.wom.setText("");
64                              imode.loss.setText("");
65                    }
66
67          }
68 }
69
```

APPENDIX E

animate.htm

```
<!--
        // See KB article about changing this dynamic HTML
        dynamicanimAttr = "dynamicanimation"
        animCancel = "skipanim"
        fpanimationPrefix = "fpAnim"
        animateElements = new Array()
        currentElement = 0
        speed = 1
        stepsZoom = 8
        stepsWord = 8
        stepsFly = 17
        stepsSpiral = 16
        stepsSpiralWord = 19
        stepsElastic = 32
        steps = stepsZoom
        step = 0
        cornerPhase=0
        outEffect=0
        function remSuffix(str)
        {
                ind=str.indexOf("FP")
                str = str.substring(0,ind)
                return str
        }
        function dynAnimOut(el)
        {
                outEffect=1
                dynAnimation(el)
                outEffect=0
        }
        function dynAnimation(obj)
        {
                animateElements = new Array()
                var ms = navigator.appVersion.indexOf("MSIE")
                var macness = navigator.appVersion.indexOf("Macintosh")

                ie4 = (ms>0) && (parseInt(navigator.appVersion.substring(ms+5, ms+6)) >= 4)
                ns6 = false
                if(!ie4)
                {
                        if((navigator.appName == "Netscape") &&
                                (parseInt(navigator.appVersion.substring(0, 1)) >= 5))
                        {
                                ns6=true
                                doc_els=document.getElementsByTagName('*')
                        }
                        else if((navigator.appName == "Netscape") &&
                                (parseInt(navigator.appVersion.substring(0, 1)) == 4))
                        {
                                doc_els=document.layers
                        }
                        else
                                return
                }
                else
                        doc_els=document.all
        if(outEffect && !ie4)
           return
                if(ie4)
                {
                   for (index=document.all.length-1; index >= document.body.sourceIndex; index--)
                        {
                                el = document.all[index]
                                if(outEffect && el != obj)
                                        continue
                                if(outEffect)
                                        animationId = el.id.substring(9,el.id.length)
                                else
                                        animationId = el.id.substring(6,el.id.length)
                                animation=remSuffix(animationId)
                                if(null != animation)
                                {
                                        altcnt=0
                                        if(     animation == "dropWord"                 ||
                                                animation == "flyTopRightWord"          ||
                                                animation == "flyBottomRightWord"       ||
```

```
                                                    animation == "waveWords"                    ||
                                                    animation == "hopWords")
                                          {
                                                    ih = el.innerHTML
                                                    outString = ""
                                                    i1 = 0
                                                    iend = ih.length
                                                    while(true)
                                                    {
                                                              i2 = startWord(ih, i1)
                                                              if(i2 == -1)
                                                              i2 = iend
                                                              outWord(ih, i1, i2, false, "", outEffect ? obj.id :
   el.id)
                                                              if(i2 == iend)
                                                                        break
                                                              i1 = i2
                                                              i2 = endWord(ih, i1)
                                                              if(i2 == -1)
                                                                        i2 = iend
                                                              if (animation == "waveWords")
                                                                        outWordAlt(ih, i1, i2, true,
   animation, altcnt)
                                                              else
                                                                        outWord(ih, i1, i2, true, (outEffect
   ? "Out" : "") + animation,
                                                                                  outEffect ?  obj.id :
   el.id)

                                                              if(i2 == iend)
                                                                        break
                                                              i1 = i2
                                                              altcnt++
                                                    }
                                                    document.all[index].innerHTML = outString
                                                    document.all[index].style.posLeft = 0
                                                    document.all[index].setAttribute(animCancel, true)
                                                    document.all[index].style.visibility="visible"
                                          }
                                 }
                        }
                        i = 0
                        for (index=0; index < doc_els.length; index++)
                        {
                                 el = doc_els[index]
                                 if(0 != el.id.indexOf(fpanimationPrefix))
                                          continue
                                 if (ie4)
                                 {
                                          elprops=el.style
                                          scrollOffsetTop=document.body.scrollTop
                                          docHeight=document.body.offsetHeight
                                          docWidth=document.body.offsetWidth
                                          elW=100
                                          elH=el.offsetHeight
                                 }
                                 else
                                 {
                                          scrollOffsetTop=window.pageYOffset
                                          docHeight=window.innerHeight
                                          docWidth=window.innerWidth
                                          if (ns6)
                                          {
                                                    elprops=el.style
                                                    elW=100
                                                    elH=el.offsetHeight
                                          }
                                          else{
                                                    elprops=el
                                                    elW=el.clip.width
                                                    elH=el.clip.height
                                          }
                                 }
                                 if(outEffect)
                                          animationId = el.id.substring(9,el.id.length)
```

```
149                     else
150                             animationId = el.id.substring(6,el.id.length)
151                     animation=remSuffix(animationId)
152                     if(outEffect && (obj != el))
153                     {
154                             if(el.SRCID != obj.id)
155                                     continue
156                     }
157                     if (null != animation )
158                     {
159                             if(ie4 && null!=el.getAttribute(animCancel, false))
160                                     continue
161                             if(!ie4)
162                             {
163                                     elprops.posLeft=elprops.left
164                                     elprops.posTop=elprops.top
165                             }
166                             el.startL=offsetLeft(el)
167                             if(animation == "flyLeft")
168                             {
169                                     elprops.posLeft = -offsetLeft(el)-elW
170                                     elprops.posTop = 0
171                             }
172                             else if(animation == "flyRight" || animation=="elasticRight")
173                             {
174                                     elprops.posLeft = -offsetLeft(el)+docWidth
175                                     elprops.posTop = 0
176                             }
177                             else if(animation == "flyTop" || animation == "dropWord")
178                             {
179                                     elprops.posLeft = 0
180                                     elprops.posTop = scrollOffsetTop-offsetTop(el)-elH
181                             }
182                             else if(animation == "flyBottom" || animation == "elasticBottom")
183                             {
184                                     elprops.posLeft = 0
185                                     elprops.posTop = scrollOffsetTop-offsetTop(el)+docHeight
186                             }
187                             else if(animation == "flyTopLeft")
188                             {
189                                     elprops.posLeft = -offsetLeft(el)-elW
190                                     elprops.posTop = scrollOffsetTop-offsetTop(el)-elH
191                             }
192                             else if(animation == "flyTopRight" || animation == "flyTopRightWord")
193                             {
194                                     elprops.posLeft = -offsetLeft(el)+docWidth
195                                     elprops.posTop = scrollOffsetTop-offsetTop(el)-elH
196                             }
197                             else if(animation == "flyCorner")
198                             {
199                                     elprops.posLeft = docWidth*0.2-offsetLeft(el)
200
201                                     elprops.posTop = scrollOffsetTop-offsetTop(el)+docHeight
202                             }
203                             else if(animation == "flyBottomLeft")
204                             {
205                                     elprops.posLeft = -offsetLeft(el)-elW
206                                     elprops.posTop = scrollOffsetTop-offsetTop(el)+docHeight
207                             }
208                             else if(animation == "flyBottomRight" || animation == "flyBottomRightWord")
209                             {
210                                     elprops.posLeft = -offsetLeft(el)+docWidth
211                                     elprops.posTop = scrollOffsetTop-offsetTop(el)+docHeight
212                             }
213                             else if(animation == "spiral")
214                             {
215                                     elprops.posLeft = -offsetLeft(el)+docWidth
216                                     elprops.posTop = scrollOffsetTop-offsetTop(el)+docHeight
217                             }
218                             else if((animation.indexOf("waveWords") != -1) || animation=="hopWords")
219                             {
220                                     if(i)
221                                     {
222                                             prevEl=animateElements[i-1]
223                                             elprops.r = offsetLeft(el)-prevEl.startL
224                                     }
```

```
225                     else
226                             elprops.r = offsetLeft(el)
227             }
228             else if(animation == "wipeLR" || animation == "wipeMID")
229             {
230                     if (ie4 && elprops.position=="absolute")
231                     {
232                             el.sizeW=el.offsetWidth
233
234                             if(macness < 0)
235                                     elprops.clip="rect(0 0 0 0)"
236                     }
237                     else if (!ie4)
238                     {
239                             if (ns6)
240                             {
241                                     elprops.clip="rect(0 0 0 0)"
242                             }
243                             else{
244                                     el.sizeW=el.clip.width
245                                     el.clip.width=0
246                             }
247                     }
248             }
249             else if(animation == "wipeTB")
250             {
251                     if (ie4 && elprops.position=="absolute")
252                     {
253                             elprops.clip="rect(0 0 0 0)"
254                     }
255                     else if(!ie4)
256                     {
257                             if (ns6)
258                             {
259                                     elprops.clip="rect(0 0 0 0)"
260                             }
261                             else{
262                                     el.sizeH=el.clip.height
263                                     el.clip.height=0
264                             }
265                     }
266             }
267             else if(animation == "zoomIn")
268             {
269                     elprops.posLeft = 0
270                     elprops.posTop = 0
271             }
272             else if(animation == "zoomOut")
273             {
274                     elprops.posLeft = 0
275                     elprops.posTop = 0
276             }
277             else
278             {
279                     continue
280             }
281             if(!outEffect)
282             {
283                     el.initLeft = elprops.posLeft
284                     el.initTop  = elprops.posTop
285                     el.endLeft  = 0
286                     el.endTop   = 0
287                     elprops.visibility = "hidden"
288             }
289             else
290             {
291                     el.initLeft = 0
292                     el.initTop  = 0
293                     el.endLeft = elprops.posLeft
294                     el.endTop  = elprops.posTop
295                     elprops.posTop = 0
296                     elprops.posLeft = 0
297             }
298             if(!ie4)
299             {
300                     elprops.left=elprops.initLeft
```

```
                                        elprops.top =elprops.initTop
                                }
                                animateElements[i++] = el
                        }
                }
                if(animateElements.length > 0)
                {
                        if(outEffect)
                                window.setTimeout("animate(1);", speed, "Javascript")
                        else
                                window.setTimeout("animate(0);", speed, "Javascript")
                }
        }
        function offsetLeft(el)
        {
                if(ie4 || ns6)
                {
                        x = el.offsetLeft
                        for (e = el.offsetParent; e; e = e.offsetParent)
                                x += e.offsetLeft
                        return x
                }
                else
                {
                        x = el.pageX
                        return x
                }
        }
        function offsetTop(el)
        {
                if(ie4 || ns6)
                {
                        y = el.offsetTop
                        for (e = el.offsetParent; e; e = e.offsetParent)
                                y += e.offsetTop;
                        return y
                }
                else
                {
                        y = el.pageY
                        return y
                }
        }
        function startWord(ih, i)
        {
                for(tag = false; i < ih.length; i++)
                {
                        c = ih.charAt(i)
                        if(c == '<')
                        {
                                if(ih.substring(i+1, i+4) == "IMG")
                                return i;
                                tag = true
                        }
                        if(!tag)
                                return i
                        if(c == '>')
                                tag = false
                }
                return -1
        }
        function endWord(ih, i)
        {
                nonSpace = false
                space = false
                img = false
                if(ih.charAt(i) == '<')
                {
                        img = true
                        i++;
                }
                while(i < ih.length)
                {
                        c = ih.charAt(i)
                        if(c != ' ')
                                nonSpace = true
```

```
377                                     if(img && c == '>')
378                                             img = false;
379                                     if(nonSpace && !img && c == ' ')
380                                             space = true
381                                     if(c == '<')
382                                             return i
383                                     if(space && c != ' ')
384                                             return i
385                                     i++
386                             }
387                             return -1
388             }
389             function outWord(ih, i1, i2, dyn, anim, srcID)
390             {
391                             if(dyn)
392                                     if(!outEffect)
393                                             outString += "<SPAN ID=\"" +  fpanimationPrefix + anim + "FP\"
    style=\"position: relative; visibility: hidden;\">"
394                                     else
395                                             outString += "<SPAN SRCID=\"" + srcID + "\"ID=\"" +  fpanimationPrefix +
    anim + "FP\" style=\"position: relative;\">"
396                             outString += ih.substring(i1, i2)
397                             if(dyn)
398                                     outString += "</SPAN>"
399             }
400             function outWordAlt(ih, i1, i2, dyn, anim, altcnt)
401             {
402                             if(dyn)
403                             {
404                                     if(altcnt%2)
405                                             outString += "<SPAN ID=\"" +  fpanimationPrefix + anim + "LFP\"
    style=\"position: relative;  visibility: hidden;\">"
406                                     else
407                                             outString += "<SPAN ID=\"" +  fpanimationPrefix + anim + "RFP\"
    style=\"position: relative;  visibility: hidden;\">"
408                             }
409
410                             outString += ih.substring(i1, i2)
411                             if(dyn)
412                                     outString += "</SPAN>"
413             }
414             function animate(animOut)
415             {
416                             {
417                                     el = animateElements[currentElement]
418
419                                     if(animOut)
420                                             animationId = el.id.substring(9,el.id.length);
421                                     else
422                                             animationId = el.id.substring(6,el.id.length);
423                                     animation=remSuffix(animationId)
424                                     if (ie4)
425                                             elprops=el.style
426                                     else{
427                                             if (ns6)
428                                                     elprops=el.style
429                                             else
430                                                     elprops=el
431                                     }
432
433                                     if(!step && !animOut)
434                                             elprops.visibility="visible"
435                                     step++
436                                     if(animation == "spiral")
437                                     {
438                                             steps = stepsSpiral
439                                             v = step/steps
440                                             rf = 1.0 - v
441                                             t = v * 2.0*Math.PI
442                                             rx = Math.max(Math.abs(el.initLeft), 200)
443                                             ry = Math.max(Math.abs(el.initTop),  200)
444                                             elprops.posLeft = Math.ceil(-rf*Math.cos(t)*rx)
445                                             elprops.posTop  = Math.ceil(-rf*Math.sin(t)*ry)
446                                     }
447                                     else if(animation == "waveWordsL" || animation=="hopWords" || animation ==
    "waveWords")
```

```
448                             {
449                                     steps = stepsSpiralWord
450                                     v = step/steps
451                                     rf = (1.0 - v)
452                                     t = v * 1.0*Math.PI
453                                     elprops.posLeft = Math.ceil(-rf*Math.cos(t)*elprops.r)
454                                     elprops.posTop  = Math.ceil(-rf*Math.sin(t)*elprops.r)
455                             }
456                             else if(animation == "waveWordsR")
457                             {
458                                     steps = stepsSpiralWord
459                                     v = step/steps
460                                     rf = (1.0 - v)
461                                     t = v * 1.0*Math.PI
462                                     elprops.posLeft = Math.ceil(-rf*Math.cos(t)*elprops.r)
463                                     elprops.posTop  = Math.ceil( rf*Math.sin(t)*elprops.r)
464                             }
465                             else if(animation == "zoomIn")
466                             {
467                                     steps = stepsZoom
468                                     elprops.fontSize = Math.ceil(50+50*step/steps) + "%"
469                                     elprops.posLeft = 0
470                             }
471                             else if(animation == "zoomOut")
472                             {
473                                     steps = stepsZoom
474                                     fontSz=Math.ceil(100+200*(steps-step)/steps) + "%"
475                                     elprops.fontSize = fontSz
476                                     elprops.posLeft = 0
477                             }
478                             else if(animation == "elasticRight")
479                             {
480                                     steps = stepsElastic
481                                     v = step/steps
482                                     rf=Math.exp(-v*7)
483                                     t = v * 1.5*Math.PI
484                                     rx =Math.abs(el.initLeft)
485                                     elprops.posLeft = rf*Math.cos(t)*rx
486                                     elprops.posTop  = 0
487                             }
488                             else if(animation == "elasticBottom")
489                             {
490                                     steps = stepsElastic
491                                     v = step/steps
492                                     rf=Math.exp(-v*7)
493                                     t = v * 2.5*Math.PI
494                                     ry =Math.abs(el.initTop)
495                                     elprops.posLeft = 0
496                                     elprops.posTop  = rf*Math.cos(t)*ry
497                             }
498                             else if(animation == "wipeLR")
499                             {
500                                     steps = stepsElastic
501                                     if(ie4 && elprops.position=="absolute")
502                                             elprops.clip = "rect(0 "+ step/steps*100 +"% 100% 0)"
503                                     else if (!ie4)
504                                     {
505                                             if (ns6)
506                                             {
507                                                     postop = elprops.top
508                                                     posleft = elprops.left
509
    str="position:absolute;top:"+postop+";left:"+posleft+";clip:rect(0px " + step/steps*el.offsetWidth + "px
    "+el.offsetHeight+"px 0px)";
510                                                     el.setAttribute("style",str);
511                                             }
512                                             else
513                                                     elprops.clip.right=step/steps*el.sizeW
514                                     }
515                             }
516                             else if(animation == "wipeTB")
517                             {
518                                     steps = stepsElastic
519                                     if(ie4 && elprops.position=="absolute")
520                                             elprops.clip = "rect(0 100% "+step/steps*el.offsetHeight+"px 0)"
521                                     else{
```

```
522                                                     if (ns6)
523                                                     {
524                                                             postop = elprops.top
525                                                             posleft = elprops.left
526
    str="position:absolute;top:"+postop+";left:"+posleft+";clip:rect(0px "+ el.offsetWidth + "px "
    +step/steps*el.offsetHeight+"px 0px)";
527                                                             el.setAttribute("style",str);
528                                                     }
529                                                     else{
530                                                             elprops.clip.bottom=step/steps*el.sizeH
531                                                     }
532                                             }
533                                     }
534                                     else if(animation == "wipeMID")
535                                     {
536                                             steps = stepsElastic
537                                             if(ie4 && elprops.position=="absolute")
538                                             {
539                                                     elprops.clip = "rect(0 "+el.sizeW/2*(1+step/steps)+"px 100%
    "+el.sizeW/2*(1-step/steps)+")"
540                                             }
541                                             else if(!ie4)
542                                             {
543                                                     if (ns6)
544                                                     {
545                                                             postop = elprops.top
546                                                             posleft = elprops.left
547
    str="position:absolute;top:"+postop+";left:"+posleft+";clip:rect(0px "+ el.offsetWidth/2*(1+step/steps) + "px
    "+el.offsetHeight+"px "+el.offsetWidth/2*(1-step/steps)+"px)";
548                                                             el.setAttribute("style",str);
549                                                     }
550                                                     else{
551                                                             elprops.clip.right=el.sizeW/2*(1+step/steps)
552                                                             elprops.clip.left=el.sizeW/2*(1-step/steps)
553                                                     }
554                                             }
555                                     }
556                                     else if(animation == "flyCorner")
557                                     {
558                                             if(!cornerPhase)
559                                             {
560                                                     steps = stepsElastic/2
561                                                     v = step/steps
562                                                     rf=Math.exp(-v*7)
563                                                     t = v * 2.5*Math.PI
564                                                     ry =Math.abs(el.initTop)
565                                                     elprops.posTop  = Math.ceil(rf*Math.cos(t)*ry)
566                                             }
567                                             else
568                                             {
569                                                     steps = stepsFly
570                                                     dl = el.initLeft / steps
571                                                     elprops.posLeft = elprops.posLeft - dl
572                                                     elprops.posTop = 0
573                                             }
574                                     }
575                                     else
576                                     {
577                                             steps = stepsFly
578                                             if(animation == "dropWord" || animation == "flyTopRightWord" || animation ==
    "flyBottomRightWord")
579                                                     steps = stepsWord
580                                             dl = (el.endLeft - el.initLeft) / steps
581                                             dt = (el.endTop  - el.initTop)  / steps
582                                             elprops.posLeft = elprops.posLeft + dl
583                                             elprops.posTop = elprops.posTop + dt
584                                     }
585                                     if (step >= steps)
586                                     {
587                                             if(!(animation == "wipeLR"      ||
588                                                     animation  == "wipeTB"  ||
589                                                     animation  == "wipeMID" ||
590                                                     (animation == "flyCorner" && !cornerPhase)))
591                                             {
```

```
592                                             elprops.posLeft = el.endLeft
593                                             elprops.posTop = el.endTop
594                                     }
595                                     if(animOut)
596                                     {
597                                             elprops.visibility="hidden"
598                                     }
599
600                                     step = 0
601                                     if(animation=="flyCorner" && !cornerPhase)
602                                             cornerPhase=1
603                                     else
604                                     {
605                                             cornerPhase=0
606                                             currentElement++
607                                     }
608
609                             }
610                             if(!ie4)
611                             {
612                                     elprops.left = elprops.posLeft
613                                     elprops.top = elprops.posTop
614                             }
615                             if(currentElement < animateElements.length)
616                             {
617                                     if(animOut)
618                                             window.setTimeout("animate(1);", speed, "Javascript")
619                                     else
620                                             window.setTimeout("animate(0);", speed, "Javascript")
621                             }
622                             else
623                                     currentElement=0
624                     }
625             }
626             function rollIn(el)
627             {
628                     var ms = navigator.appVersion.indexOf("MSIE")
629                     ie4 = (ms>0) && (parseInt(navigator.appVersion.substring(ms+5, ms+6)) >= 4)
630                     if(ie4)
631                     {
632                             el.initstyle=el.style.cssText;el.style.cssText=el.fprolloverstyle
633                     }
634             }
635             function rollOut(el)
636             {
637                     var ms = navigator.appVersion.indexOf("MSIE")
638                     ie4 = (ms>0) && (parseInt(navigator.appVersion.substring(ms+5, ms+6)) >= 4)
639                     if(ie4)
640                     {
641                             el.style.cssText=el.initstyle
642                     }
643             }
644             function clickSwapStyle(el)
645             {
646                     var ms = navigator.appVersion.indexOf("MSIE")
647                     ie4 = (ms>0) && (parseInt(navigator.appVersion.substring(ms+5, ms+6)) >= 4)
648                     if(ie4)
649                     {
650                             ts=el.style.cssText
651                             el.style.cssText=el.fprolloverstyle
652                             el.fprolloverstyle=ts
653                     }
654             }
655             function clickSwapImg(el)
656             {
657             if(document.all || document.layers)
658             {
659                     ts=el.src
660                     el.src=el.lowsrc
661                     el.lowsrc=ts
662             }
663             }
664 //-->
665
666
```

side.htm

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>by</title>
<script language="JavaScript">
<!-- Hide from old browsers

function on3(name)
{
      document[name].src = eval(name + "on.src");
}
function off3(name)
{
      document[name].src = eval(name + ".src");
}

function on(name)
{
       on3(name);
}
function off(name)
{
       off3(name);
}


link0 = new Image();
link0.src = "animation/anim_01.gif";
link0on = new Image();
link0on.src = "images/val.gif";

link = new Image();
link.src = "menues/home_a.gif";
linkon = new Image();
linkon.src = "menues/home_b.gif";

link1 = new Image();
link1.src = "menues/applet_a.gif";
link1on = new Image();
link1on.src = "menues/applet_b.gif";

link2 = new Image();
link2.src = "menues/model_a.gif";
link2on = new Image();
link2on.src = "menues/model_b.gif";

link3 = new Image();
link3.src = "menues/pdf_a.gif";
link3on = new Image();
link3on.src = "menues/pdf_b.gif";

link4 = new Image();
link4.src = "menues/word_a.gif";
link4on = new Image();
link4on.src = "menues/word_b.gif";

link5 = new Image();
link5.src = "menues/hia_a.gif";
link5on = new Image();
link5on.src = "menues/hia_b.gif";

link6 = new Image();
link6.src = "menues/siving_a.gif";
link6on = new Image();
link6on.src = "menues/siving_b.gif";

link7 = new Image();
link7.src = "menues/ppt_a.gif";
link7on = new Image();
link7on.src = "menues/ppt_b.gif";

link8 = new Image();
link8.src = "menues/poster_a.gif";
```

```
link8on = new Image();
link8on.src = "menues/poster_b.gif";
// -->
</script>
<base target="rbottom">
</head>

<body topmargin="0" leftmargin="0">

<div align="left">
 <table border="0" cellpadding="0" cellspacing="0" width="14%" height="346">
  <tr>
   <td width="100%" height="1" valign="top" align="left">
   <p align="left">
   <img border="0" src="animation/anim_01.gif" align="top" target="main"  width="140" height="110"></td>
  </tr>
  <tr>
   <td width="100%" height="121">
    <div align="left">
    <table border="0" cellpadding="0" cellspacing="0" width="100%" height="313">
     <tr>
      <td width="100%" align="left" height="35">
      <a target="rbottom" href="start.htm">
      <img border="0" src="menues/home_a.gif" target="main" onmouseover="on('link');" onmouseout="off('link')"
name="link" width="140" height="35"></a></td>
     </tr>
     <tr>
      <td width="100%" align="left" height="35">
      <a target="rbottom" title="Switch to JavaSwing Applet" href="JavaApplet/iMODE.htm">
      <img border="0" src="menues/applet_a.gif" target="main" onmouseover="on('link1');" onmouseout="off('link1')"
name="link1" width="140" height="35"></a></td>
     </tr>
     <tr>
      <td width="100%" align="left" height="35">
      <a target="rbottom" title="Switch to Model in PowerSim" href="model/modell.sig">
      <img border="0" src="menues/model_a.gif" target="main" onmouseover="on('link2');" onmouseout="off('link2')"
name="link2" width="140" height="35"></a></td>
     </tr>
     <tr>
      <td width="100%" align="left" height="35">
      <a target="rbottom" title="Switch to the report in PDF format" href="pdf/report.pdf">
      <img border="0" src="menues/pdf_a.gif" target="main" onmouseover="on('link3');" onmouseout="off('link3')"
name="link3" width="140" height="35"></a></td>
     </tr>

     <tr>
      <td width="100%" align="left" height="35">
      <a target="rbottom" title="Switch the report in word format" href="word/report.doc">
      <img border="0" src="menues/word_a.gif" target="main" onmouseover="on('link4');" onmouseout="off('link4')"
name="link4" width="140" height="35"></a></td>
     </tr>

     <tr>
      <td width="100%" align="left" height="35">
      <a target="rbottom" title="Agder University College" href="http://www.hia.no">
      <img border="0" src="menues/hia_a.gif" target="main" onmouseover="on('link5');" onmouseout="off('link5')"
name="link5" width="140" height="35"></a></td>
     </tr>

     <tr>
      <td width="100%" align="left" height="35">
      <a target="rbottom" title="Master of Information Technology" href="http://siving.hia.no">
      <img border="0" src="menues/siving_a.gif" target="main" onmouseover="on('link6');" onmouseout="off('link6')"
name="link6" width="140" height="35"></a></td>
     </tr>

     <tr>
      <td width="100%" align="left" height="35">
      <a target="rbottom" title="Switch the Power Point Presentation" href="PPT/iMODE.ppt">
      <img border="0" src="menues/ppt_a.gif" target="main" onmouseover="on('link7');" onmouseout="off('link7')"
name="link7" width="140" height="35"></a></td>
     </tr>

     <tr>
      <td width="100%" align="left" height="35">
      <a target="rbottom" title="Switch the Poster" href="pdf/poster.pdf">
```

```
145          <img border="0" src="menues/poster_a.gif" target="main" onmouseover="on('link8');" onmouseout="off('link8')"
    name="link8" width="140" height="35"></a></td>
146        </tr>
147
148        </table>
149      </div>
150      </td>
151    </tr>
152  </table>
153 </div>
154
155 <p><img border="0" src="animation/my_animation.gif" width="136" height="146"></p>
156
157 </body>
158
159 </html>
```

reports.htm

```
<html>

<head>
<meta http-equiv="Content-Language" content="no-bok">
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Ny side 1</title>
<style>
<!--
h1
	{margin-bottom:.0001pt;
	page-break-after:avoid;
	font-size:12.0pt;
	font-family:"Times New Roman";
	text-decoration:underline;
	text-underline:single; margin-left:0cm; margin-right:0cm; margin-top:0cm}
 p.MsoNormal
	{mso-style-parent:"";
	margin-bottom:.0001pt;
	font-size:12.0pt;
	font-family:"Times New Roman";
	margin-left:0cm; margin-right:0cm; margin-top:0cm}
h5
	{margin-top:12.0pt;
	margin-right:0cm;
	margin-bottom:3.0pt;
	margin-left:0cm;
	font-size:13.0pt;
	font-family:"Times New Roman";
	font-style:italic}
p.MsoFooter
	{margin-bottom:.0001pt;
	tab-stops:center 216.0pt right 432.0pt;
	font-size:12.0pt;
	font-family:"Times New Roman";
	margin-left:0cm; margin-right:0cm; margin-top:0cm}
-->
</style>
</head>

<body topmargin="0" leftmargin="0">

<p class="MsoNormal"><b>&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;
<span lang="EN-US"><i>References</i></b>:</span></p>
<ul>
 <li>
 <p class="MsoNormal"><span lang="EN-GB" dir="LTR">R. Geoff Coyle: System
 Dynamics Modelling - A Practical Approach, Chapman &amp;amp; Hall, </span>
 <span lang="EN-GB">London, Glasgow. 1996. ISBN 0-412-61710-2. 413 pages.</span></li>
 <li>
 <p class="MsoFooter">
 <span style="font-style: normal; font-variant: normal; font-weight: normal; font-size: 7.0pt; font-family: Times New Roman" lang="EN-GB">&amp;nbsp;</span><span lang="EN-GB" dir="LTR">Jose
 J. Gonzalez: Lectures in </span><span lang="EN-GB">Agder College, Dep. Of
 Communication &amp;amp; Information Technology, 2000. [<a href="http://fag.grm.hia.no/ikt1200" title="Modelling &amp;amp; Simulation" style="font-family: Times New Roman; color: blue; text-decoration: underline; text-underline: single">http://fag.grm.hia.no/ikt1200</a>]</span></li>
 <li>
 <p class="MsoFooter"><span lang="EN-GB" dir="LTR">Maier, F. H.: New Product
 Diffusion Models in Innovation Management- A System Dynamics Perspective.
 System Dynamics Review Vol. 14, No. 4, (Winter 1998): 285-308 </span></li>
</ul>
<h5>&amp;nbsp;&amp;nbsp;&amp;nbsp; <span lang="EN-US">Web sites:</span></h5>
<ul>
 <li>
 <p class="MsoNormal"><span lang="EN-GB" dir="LTR">The UMTS Forum [<a href="http://www.umts-forum.com/" title="The UMTS - Forum" style="font-family: Times New Roman; color: blue; text-decoration: underline; text-underline: single">http://www.umts-forum.org/</a></span><span dir="LTR">]</span></li>
 <li>
 <p class="MsoNormal">
 <span style="font-style: normal; font-variant: normal; font-weight: normal; font-size: 7.0pt; font-family: Times New Roman" lang="EN-GB">&amp;nbsp;</span><span lang="EN-GB" dir="LTR">Eurotechnology
 [<a href="http://www.eurotechnology.com/imode/faq-gen.html" title="Eurotechnology" style="font-family: Times New Roman; color: blue; text-decoration: underline; text-underline: single">http://www.eurotechnology.com/iMODE/faq-gen.html</a>]</span></li>
```

```
 <li>
 <p class="MsoNormal"><span dir="LTR"><i><span lang="EN-GB">NTT DoCoMo</span></i></span><span lang="EN-GB">,
 2001 [<a href="http://www.nttdocomo.com/new/contents/01/whatnew0305.html" title="NTT Do Co Mo" style="font-family: Times New Roman; color: blue; text-decoration: underline; text-underline: single">http://www.nttdocomo.com/new/contents/01/whatnew0305.html</a>]</span></li>
 <li>
 <p class="MsoNormal"><span lang="EN-GB" dir="LTR">Arthur D. Little [<a href="http://www.arthurdlittle.com/" title="Arthur D. Little" style="font-family: Times New Roman; color: blue; text-decoration: underline; text-underline: single">http://www.arthurdlittle.com/</a>]</span></li>
 <li>
 <p class="MsoNormal">
 <span style="font-style: normal; font-variant: normal; font-weight: normal; font-size: 7.0pt; font-family: Times New Roman" lang="EN-GB"> </span><span lang="EN-GB" dir="LTR">Strategy
 Dynamics web site [<a href="http://www.strategydynamics.com/" title="Strategy Dynamics web site" style="font-family: Times New Roman; color: blue; text-decoration: underline; text-underline: single">http://www.strategydynamics.com/</a>]</span></li>
 <li>
 <p class="MsoNormal">
 <span style="font-style: normal; font-variant: normal; font-weight: normal; font-size: 7.0pt; font-family: Times New Roman" lang="EN-GB"> </span><span lang="EN-GB" dir="LTR">NTT
 DoCoMo, INC [<a href="http://www.nttdocomo.co.jp/english/" title="The english version of NTT DoCoMo web site" style="font-family: Times New Roman; color: blue; text-decoration: underline; text-underline: single">http://www.nttdocomo.co.jp/english/</a></span><span dir="LTR">]</span></li>
 <li>
 <p class="MsoNormal">
 <span style="font-style: normal; font-variant: normal; font-weight: normal; font-size: 7.0pt; font-family: Times New Roman" lang="EN-GB"> </span><span lang="EN-GB" dir="LTR">Sun
 Microsystems [<a href="http://java.sun.com/" title="Sun Microsystems web site" style="font-family: Times New Roman; color: blue; text-decoration: underline; text-underline: single">http://java.sun.com</a>]</span></li>
 <li>
 <p class="MsoNormal"><span dir="LTR">J</span><span lang="EN-GB" dir="LTR">ava
 Community web site [<a href="http://community.borland.com/java/" title="Java Community, Borland" style="font-family: Times New Roman; color: blue; text-decoration: underline; text-underline: single">http://community.borland.com/java/</a>]</span></li>
</ul>
<p class="MsoNormal"><span lang="EN-GB"><b><i>Applications</i></b>: </span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
</span></span><span lang="EN-GB" dir="LTR">Microsoft Windows XP home edition</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
</span></span><span lang="EN-GB" dir="LTR">Powersim Studio Student Express 2001</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
</span></span><span lang="EN-GB" dir="LTR">Powersim Solver 2.1</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
</span></span><span lang="EN-GB" dir="LTR">Microsoft Word 2002</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
</span></span><span lang="EN-GB" dir="LTR">Microsoft Excel 2002</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
</span></span><span lang="EN-GB" dir="LTR">Microsoft PowerPoint 2002</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
</span></span><span lang="EN-GB" dir="LTR">Adobe Acrobat ver. 5.0 </span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
</span></span><span lang="EN-GB" dir="LTR">Adobe PhotoShop ver. 7.0</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
</span></span><span lang="EN-GB" dir="LTR">Microsoft FrontPage 2002</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New Roman&amp;quot;">        
```

```
</span></span><span lang="EN-GB" dir="LTR">Microsoft Visio Professional</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New
Roman&amp;quot;">&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;
</span></span><span lang="EN-GB" dir="LTR">Paint Shop Pro 7.1</span></p>
<p class="MsoNormal" style="text-indent: -18.0pt; margin-left: 27.0pt">
<span lang="EN-GB" style="font-family: Wingdings">§<span style="font:7.0pt &amp;quot;Times New
Roman&amp;quot;">&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;
</span></span><span lang="EN-GB" dir="LTR">Java Development Kit (JDK 1.4.1)</span></p>

</body>

</html>
```

index.html

```
<html>

<head>
<title>MODELLING IMODE SUCCESS WITH SYSTEM DYNAMICS</title>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
</head>

<frameset framespacing="0" border="0" cols="156,*" frameborder="0">
  <frame name="left" scrolling="no" noresize target="rbottom" src="side.htm">
  <frameset rows="8%,*">
    <frame name="rtop" target="rbottom" src="header.htm" scrolling="no">
    <frame name="rbottom" src="start.htm" scrolling="auto" target="rtop">
  </frameset>
  <noframes>
  <body>

  <p>This page uses frames, but your browser doesn't support them.</p>

  </body>
  </noframes>
</frameset>

</html>
```

header.htm

```
<HTML>
<HEAD>
<TITLE>MODELLING IMODE SUCCESS WITH SYSTEM DYNAMICS</TITLE>
</HEAD>
<BODY>
<marquee height=20 scrollamount=4 style="CPLOR:#000000; font-family:Comic Sans MS; color:#FFFFFF;
font-size:10pt;
position:absolute; font-style:oblique" border=2 align=right bgcolor="#000080">
<b> MODELLING IMODE SUCCESS WITH SYSTEM DYNAMICS</b>
This paper is the whole part of my M. SC. Degree research in Information and Communication technology Department,
Agder University College.
The work discusses system dynamics models for the dynamics of the success market growth for iMODE, mobile
Internet services,
which has been launched by NTTDoCoMo in Japan since February 1999.
The models in my report address the success growth of the total market for iMODE services for a newly launched
mobile
Internet service that is adopted by customers.
My work provide illustrative models for iMODE services based on dynamic of Word of Mouth, advertising,
pricing and quality of services.
Both differential equations and PowerSim dynamics models notations are presented for these models in my report.
Using system dynamics thinking in my work leads to equations of a model, simulation to understand dynamic behaviour
of alternatives policies.
I would like to thank my mother's loves that have allowed this work to come into existence.
As well as I thank my supervisors for their encouragements and advices to make this work happened.
Finally, I would like to thank all the staff in Agder University College for their helps.


</marquee>
</BODY>
</HTML>
```

autorun.inf

```
[autorun]
open=browsercall.exe iexplore&CDREPLACEindex.html
or
open=browsercall.exe netscape&CDREPLACEindex.html
icon=imode.ico


```

start.htm

```
<html xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:w="urn:schemas-microsoft-com:office:word"
xmlns:st1="urn:schemas-microsoft-com:office:smarttags"
xmlns="http://www.w3.org/TR/REC-html40">

<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-1252">
<meta name=ProgId content=FrontPage.Editor.Document>
<meta name=Generator content="Microsoft FrontPage 5.0">
<meta name=Originator content="Microsoft Word 10">
<title>This work is the whole part of a M</title>
<o:SmartTagType namespaceuri="urn:schemas-microsoft-com:office:smarttags"
 name="country-region"/>
<o:SmartTagType namespaceuri="urn:schemas-microsoft-com:office:smarttags"
 name="PlaceType"/>
<o:SmartTagType namespaceuri="urn:schemas-microsoft-com:office:smarttags"
 name="PlaceName"/>
<o:SmartTagType namespaceuri="urn:schemas-microsoft-com:office:smarttags"
 name="place"/>
<!--[if gte mso 9]><xml>
 <o:DocumentProperties>
  <o:Author>Brukernavn</o:Author>
  <o:LastAuthor>Brukernavn</o:LastAuthor>
  <o:Revision>1</o:Revision>
  <o:TotalTime>0</o:TotalTime>
  <o:Created>2003-04-24T09:02:00Z</o:Created>
  <o:LastSaved>2003-04-24T09:02:00Z</o:LastSaved>
  <o:Pages>1</o:Pages>
  <o:Words>297</o:Words>
  <o:Characters>1579</o:Characters>
  <o:Lines>13</o:Lines>
  <o:Paragraphs>3</o:Paragraphs>
  <o:CharactersWithSpaces>1873</o:CharactersWithSpaces>
  <o:Version>10.2625</o:Version>
 </o:DocumentProperties>
</xml><![endif]--><!--[if gte mso 9]><xml>
 <w:WordDocument>
  <w:GrammarState>Clean</w:GrammarState>
  <w:HyphenationZone>21</w:HyphenationZone>
  <w:Compatibility>
   <w:BreakWrappedTables/>
   <w:SnapToGridInCell/>
   <w:ApplyBreakingRules/>
   <w:WrapTextWithPunct/>
   <w:UseAsianBreakRules/>
   <w:UseFELayout/>
  </w:Compatibility>
  <w:BrowserLevel>MicrosoftInternetExplorer4</w:BrowserLevel>
 </w:WordDocument>
</xml><![endif]-->
<style>
<!--
 /* Font Definitions */
 @font-face
                {font-family:Wingdings;
                panose-1:5 0 0 0 0 0 0 0 0 0;
                mso-font-charset:2;
                mso-generic-font-family:auto;
                mso-font-pitch:variable;
                mso-font-signature:0 268435456 0 0 -2147483648 0;}
@font-face
                {font-family:Batang;
                panose-1:2 3 6 0 0 1 1 1 1 1;
                mso-font-alt:\BC14\D0D5;
                mso-font-charset:129;
                mso-generic-font-family:roman;
                mso-font-pitch:variable;
                mso-font-signature:-1342176593 1775729915 48 0 524447 0;}
@font-face
                {font-family:"\@Batang";
                panose-1:2 3 6 0 0 1 1 1 1 1;
                mso-font-charset:129;
                mso-generic-font-family:roman;
                mso-font-pitch:variable;
                mso-font-signature:-1342176593 1775729915 48 0 524447 0;}
 /* Style Definitions */
```

```
 p.MsoNormal, li.MsoNormal, div.MsoNormal
                {mso-style-parent:"";
                margin-bottom:.0001pt;
                mso-pagination:widow-orphan;
                font-size:12.0pt;
                font-family:"Times New Roman";
                mso-fareast-font-family:"Times New Roman";
                mso-ansi-language:EN-US;
                mso-fareast-language:EN-US; margin-left:0cm; margin-right:0cm; margin-top:0cm}
p.MsoDate, li.MsoDate, div.MsoDate
                {mso-style-next:Normal;
                margin:0cm;
                margin-bottom:.0001pt;
                mso-pagination:widow-orphan;
                font-size:12.0pt;
                font-family:"Times New Roman";
                mso-fareast-font-family:"Times New Roman";
                mso-ansi-language:EN-US;
                mso-fareast-language:EN-US;}
a:link, span.MsoHyperlink
                {mso-ansi-font-size:11.0pt;
                font-family:"Times New Roman";
                mso-ascii-font-family:"Times New Roman";
                mso-hansi-font-family:"Times New Roman";
                mso-bidi-font-family:"Times New Roman";
                color:blue;
                text-decoration:underline;
                text-underline:single;}
a:visited, span.MsoHyperlinkFollowed
                {color:purple;
                text-decoration:underline;
                text-underline:single;}
span.GramE
                {mso-style-name:"";
                mso-gram-e:yes}
@page Section1
                {size:595.3pt 841.9pt;
                margin:70.85pt 70.85pt 70.85pt 70.85pt;
                mso-header-margin:35.4pt;
                mso-footer-margin:35.4pt;
                mso-paper-source:0;}
div.Section1
                {page:Section1;}
 /* List Definitions */
 @list l0
                {mso-list-id:7022095;
                mso-list-type:hybrid;
                mso-list-template-ids:1965165888 -1 -1 -1 -1 -1 -1 -1 -1 -1;}
@list l0:level1
                {mso-level-number-format:bullet;
                mso-level-text:\F0B7;
                mso-level-tab-stop:36.0pt;
                mso-level-number-position:left;
                text-indent:-18.0pt;
                font-family:Symbol;}
@list l0:level2
                {mso-level-number-format:bullet;
                mso-level-text:o;
                mso-level-tab-stop:72.0pt;
                mso-level-number-position:left;
                text-indent:-18.0pt;
                font-family:"Courier New";
                mso-bidi-font-family:"Times New Roman";}
@list l0:level3
                {mso-level-number-format:bullet;
                mso-level-text:\F0A7;
                mso-level-tab-stop:108.0pt;
                mso-level-number-position:left;
                text-indent:-18.0pt;
                font-family:Wingdings;}
@list l0:level4
                {mso-level-number-format:bullet;
                mso-level-text:\F0B7;
                mso-level-tab-stop:144.0pt;
                mso-level-number-position:left;
                text-indent:-18.0pt;
```

```
153             font-family:Symbol;}
154 @list l0:level5
155             {mso-level-number-format:bullet;
156             mso-level-text:o;
157             mso-level-tab-stop:180.0pt;
158             mso-level-number-position:left;
159             text-indent:-18.0pt;
160             font-family:"Courier New";
161             mso-bidi-font-family:"Times New Roman";}
162 @list l0:level6
163             {mso-level-number-format:bullet;
164             mso-level-text:\F0A7;
165             mso-level-tab-stop:216.0pt;
166             mso-level-number-position:left;
167             text-indent:-18.0pt;
168             font-family:Wingdings;}
169 @list l0:level7
170             {mso-level-number-format:bullet;
171             mso-level-text:\F0B7;
172             mso-level-tab-stop:252.0pt;
173             mso-level-number-position:left;
174             text-indent:-18.0pt;
175             font-family:Symbol;}
176 @list l0:level8
177             {mso-level-number-format:bullet;
178             mso-level-text:o;
179             mso-level-tab-stop:288.0pt;
180             mso-level-number-position:left;
181             text-indent:-18.0pt;
182             font-family:"Courier New";
183             mso-bidi-font-family:"Times New Roman";}
184 @list l0:level9
185             {mso-level-number-format:bullet;
186             mso-level-text:\F0A7;
187             mso-level-tab-stop:324.0pt;
188             mso-level-number-position:left;
189             text-indent:-18.0pt;
190             font-family:Wingdings;}
191 @list l1
192             {mso-list-id:1425152379;
193             mso-list-type:hybrid;
194             mso-list-template-ids:-1865504802 -1 -1 -1 -1 -1 -1 -1 -1 -1;}
195 @list l1:level1
196             {mso-level-number-format:bullet;
197             mso-level-text:\F0B7;
198             mso-level-tab-stop:36.0pt;
199             mso-level-number-position:left;
200             text-indent:-18.0pt;
201             font-family:Symbol;}
202 @list l1:level2
203             {mso-level-number-format:bullet;
204             mso-level-text:o;
205             mso-level-tab-stop:72.0pt;
206             mso-level-number-position:left;
207             text-indent:-18.0pt;
208             font-family:"Courier New";
209             mso-bidi-font-family:"Times New Roman";}
210 @list l1:level3
211             {mso-level-number-format:bullet;
212             mso-level-text:\F0A7;
213             mso-level-tab-stop:108.0pt;
214             mso-level-number-position:left;
215             text-indent:-18.0pt;
216             font-family:Wingdings;}
217 @list l1:level4
218             {mso-level-number-format:bullet;
219             mso-level-text:\F0B7;
220             mso-level-tab-stop:144.0pt;
221             mso-level-number-position:left;
222             text-indent:-18.0pt;
223             font-family:Symbol;}
224 @list l1:level5
225             {mso-level-number-format:bullet;
226             mso-level-text:o;
227             mso-level-tab-stop:180.0pt;
228             mso-level-number-position:left;
```

```
229             text-indent:-18.0pt;
230             font-family:"Courier New";
231             mso-bidi-font-family:"Times New Roman";}
232 @list l1:level6
233             {mso-level-number-format:bullet;
234             mso-level-text:\F0A7;
235             mso-level-tab-stop:216.0pt;
236             mso-level-number-position:left;
237             text-indent:-18.0pt;
238             font-family:Wingdings;}
239 @list l1:level7
240             {mso-level-number-format:bullet;
241             mso-level-text:\F0B7;
242             mso-level-tab-stop:252.0pt;
243             mso-level-number-position:left;
244             text-indent:-18.0pt;
245             font-family:Symbol;}
246 @list l1:level8
247             {mso-level-number-format:bullet;
248             mso-level-text:o;
249             mso-level-tab-stop:288.0pt;
250             mso-level-number-position:left;
251             text-indent:-18.0pt;
252             font-family:"Courier New";
253             mso-bidi-font-family:"Times New Roman";}
254 @list l1:level9
255             {mso-level-number-format:bullet;
256             mso-level-text:\F0A7;
257             mso-level-tab-stop:324.0pt;
258             mso-level-number-position:left;
259             text-indent:-18.0pt;
260             font-family:Wingdings;}
261 ol
262             {margin-bottom:0cm;}
263 ul
264             {margin-bottom:0cm;}
265 -->
266 </style>
267 <!--[if gte mso 10]>
268 <style>
269  /* Style Definitions */
270  table.MsoNormalTable
271             {mso-style-name:"Vanlig tabell";
272             mso-tstyle-rowband-size:0;
273             mso-tstyle-colband-size:0;
274             mso-style-noshow:yes;
275             mso-style-parent:"";
276             mso-padding-alt:0cm 5.4pt 0cm 5.4pt;
277             mso-para-margin:0cm;
278             mso-para-margin-bottom:.0001pt;
279             mso-pagination:widow-orphan;
280             font-size:10.0pt;
281             font-family:"Times New Roman"}
282 </style>
283 <![endif]-->
284 <base target="rtop">
285 </head>
286
287 <body lang=NO-BOK link=#0000FF vlink=#800080 style='tab-interval:35.4pt'>
288
289 <div class=Section1>
290
291 <p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'>This work
292 is the whole part of a M. SC. Degree research in Information and Communication
293 technology Dep., </span><st1:place><st1:PlaceName><span lang=EN-GB
294  style='mso-ansi-language:EN-GB'>Agder</span></st1:PlaceName><span lang=EN-GB
295  style='mso-ansi-language:EN-GB'> </span><st1:PlaceType><span lang=EN-GB
296  style='mso-ansi-language:EN-GB'>University</span></st1:PlaceType><span
297  lang=EN-GB style='mso-ansi-language:EN-GB'> </span><st1:PlaceType><span
298  lang=EN-GB style='mso-ansi-language:EN-GB'>College</span></st1:PlaceType></st1:place><span
299 lang=EN-GB style='mso-ansi-language:EN-GB'>, involving the study and analysis
300 of <i style='mso-bidi-font-style:normal'>iMODE</i>, the wireless Internet
301 cellular system<b style='mso-bidi-font-weight:normal'><i style='mso-bidi-font-style:
302 normal'>.<span style='mso-spacerun:yes'>  </span></i></b><span class=GramE><i
303 style='mso-bidi-font-style:normal'><span style='mso-bidi-font-weight:bold'>iMODE</span></i></span>
304 was launched in </span><st1:country-region><st1:place><span lang=EN-GB
```

```
 style='mso-ansi-language:EN-GB'>Japan</span></st1:place></st1:country-region><span
lang=EN-GB style='mso-ansi-language:EN-GB'> in February 1999 by <i>NTT DoCoMo</i>,
and is responsible for significant growth in the market.<i> NTT DoCoMo</i> is
</span><st1:country-region><st1:place><span
 lang=EN-GB style='mso-ansi-language:EN-GB'>Japan</span></st1:place></st1:country-region><span
lang=EN-GB style='mso-ansi-language:EN-GB'>’s largest wireless carrier and the
second largest in the World.<i> </i><span
style='mso-spacerun:yes'> </span>During my research I discovered that, to date,
there are more than 36 million users in </span><st1:country-region><st1:place><span
 lang=EN-GB style='mso-ansi-language:EN-GB'>Japan</span></st1:place></st1:country-region><span
lang=EN-GB style='mso-ansi-language:EN-GB'> of <i style='mso-bidi-font-style:
normal'>iMODE</i>-service, the majority of users are young people, and however,
middle aged and elderly people also use the <i style='mso-bidi-font-style:normal'>iMODE
</i><span style='mso-bidi-font-style:italic'>service</span>.<o:p></o:p></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'><o:p> </o:p></span></p>

<p class=MsoDate><i><span lang=EN-GB style='mso-ansi-language:EN-GB'>NTT DoCoMo
</span></i><span lang=EN-GB style='mso-ansi-language:EN-GB'>has created an
attractive value chain and a mutually reinforcing ecosystem around the <i>iMODE</i>
mobile Internet services.<o:p></o:p></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'><o:p> </o:p></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'>Choosing <i
style='mso-bidi-font-style:normal'>System dynamics</i> approach for this
purpose has many advantages including:<o:p></o:p></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'><o:p> </o:p></span></p>

<ul style='margin-top:0cm' type=disc>
 <li class=MsoNormal style='mso-list:l0 level1 lfo1;tab-stops:list 36.0pt'><i
    style='mso-bidi-font-style:normal'><span lang=EN-GB style='mso-ansi-language:
    EN-GB'>System dynamics</span></i><span lang=EN-GB style='mso-ansi-language:
    EN-GB'> is a powerful method to analyse why the managed systems do not
    behave as the users or vendors wish.<o:p></o:p></span></li>
 <li class=MsoNormal style='mso-list:l0 level1 lfo1;tab-stops:list 36.0pt'><i
    style='mso-bidi-font-style:normal'><span lang=EN-GB style='mso-ansi-language:
    EN-GB'>System dynamics</span></i><span lang=EN-GB style='mso-ansi-language:
    EN-GB'> is supporting a strategic point of view.<o:p></o:p></span></li>
 <li class=MsoNormal style='mso-list:l0 level1 lfo1;tab-stops:list 36.0pt'><i
    style='mso-bidi-font-style:normal'><span lang=EN-GB style='mso-ansi-language:
    EN-GB'>System dynamics</span></i><span lang=EN-GB style='mso-ansi-language:
    EN-GB'> consists of two phases, qualitatively and quantitatively. <o:p></o:p></span></li>
</ul>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'><o:p> </o:p></span></p>

<p class=MsoNormal><i><span lang=EN-GB style='mso-ansi-language:EN-GB'>System
dynamics</span></i><span lang=EN-GB style='mso-ansi-language:EN-GB'> implies
modelling a real-life phenomenon in terms of mathematical equations. Through
millions of computations per second, the computer model can simulate the
behaviour of given system through time. That makes it easy to understand how
our system behaves.<o:p></o:p></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'><o:p> </o:p></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'><span
style='mso-spacerun:yes'> </span><o:p></o:p></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'>Using <i
style='mso-bidi-font-style:normal'>System dynamics</i> allows the more complex
development models to investigate the innovation diffusions process.<span
style='mso-spacerun:yes'>  </span><i style='mso-bidi-font-style:normal'>System
dynamics</i> <span style='mso-bidi-font-style:italic'>is a methodology for
studying and managing complex feedback systems.<o:p></o:p></span></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'><o:p> </o:p></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'>The steps
of our project are:<o:p></o:p></span></p>

<p class=MsoNormal><span lang=EN-GB style='mso-ansi-language:EN-GB'><o:p> </o:p></span></p>

<ul style='margin-top:0cm' type=disc>
 <li class=MsoNormal style='mso-list:l1 level1 lfo2;tab-stops:list 36.0pt'><span
```

```
380     lang=EN-GB style='mso-ansi-language:EN-GB'>Definition of the problem.<o:p></o:p></span></li>
381  <li class=MsoNormal style='mso-list:l1 level1 lfo2;tab-stops:list 36.0pt'><span
382     lang=EN-GB style='mso-ansi-language:EN-GB'>Developing a dynamic hypothesis
383     explaining the cause of the problem.<o:p></o:p></span></li>
384  <li class=MsoNormal style='mso-list:l1 level1 lfo2;tab-stops:list 36.0pt'><span
385     lang=EN-GB style='mso-ansi-language:EN-GB'>Building a simulation model.<o:p></o:p></span></li>
386  <li class=MsoNormal style='mso-list:l1 level1 lfo2;tab-stops:list 36.0pt'><span
387     lang=EN-GB style='mso-ansi-language:EN-GB'>Testing the model to establish
388     if it reproduces the behaviour seen in the real world.<o:p></o:p></span></li>
389  <li class=MsoNormal style='mso-list:l1 level1 lfo2;tab-stops:list 36.0pt'><span
390     lang=EN-GB style='mso-ansi-language:EN-GB'>Analysis the success factors.<o:p></o:p></span></li>
391  <li class=MsoNormal style='mso-list:l1 level1 lfo2;tab-stops:list 36.0pt'><span
392     lang=EN-GB style='mso-ansi-language:EN-GB'>Policy recommendations.</span></li>
393 </ul>
394
395 <p class=MsoNormal><span lang=EN-US><o:p> </o:p></span></p>
396
397 </div>
398
399 <p class="MsoNormal"><font color="#FFFFFF"><i>
400 <span style="background-color: #0000FF">Created by Adil Gebory, May 2003, Oslo/Norway</span></i></font></p>
401
402 </body>
403
404 </html>
```