UPTEC X 06 008
FEB 2006
ISSN 1401-2138

SARA ERIKSSON

Design of data structures for description of temporal processes in biological systems

Master’s degree project

UPPSALA UNIVERSITET

Molecular Biotechnology Programme

Uppsala University School of Engineering

UPTEC X 06 008	**Date of issue 2006-02**

Author

Sara Eriksson

Title (English)

Design of data structures for description of temporal processes in biological systems

Title (Swedish)

Abstract

A prototype of a temporal database, for storing data from animal studies, was developed. The prototype can store the design of a study and the parameters obtained from the study. In an interface searches in the prototype can be made. The results from studies can be viewed, numerically and graphically, and studies can be compared.

Keywords

Temporal database, temporal biological systems, animal studies

Supervisors

Per Kraulis & Stephen James
Biovitrum AB

Scientific reviewer

Johan Elf
Department of Chemistry and Chemical Biology, Harvard University

Project name	Sponsors
Language **English**	Security
ISSN 1401-2138	Classification
Supplementary bibliographical information	Pages **22**

Biology Education Centre Biomedical Center Husargatan 3 Uppsala
Box 592 S-75124 Uppsala Tel +46 (0)18 4710000 Fax +46 (0)18 555217

Design of Data Structures for Description of Temporal Processes in Biological Systems

Sara Eriksson

Sammanfattning

Inom biologin har nya tekniker, t.ex. genom-projekten, skapat förutsättningar för att i detalj beskriva biologiska system. De biologiska systemen är väl dokumenterade i olika biologiska databaser, t.ex. sekvensdatabaserna. I dessa databaser är det endast de strukturella aspekterna av systemen som är representerade. Trots att tiden går som en röd tråd genom alla biologiska system, finns få allmänna databaser eller liknande system som beskriver biologiska förlopp över tiden. Om man t.ex. är intresserad av cellcykeln finns ingen databas som beskriver när gener aktiveras eller när proteiner skapas respektive försvinner, med tiden som grundläggande koordinat.

Syftet med detta projekt var att skapa en prototyp för en temporal databas där farmakologerna på Biovitrum kan lagra information från sina studier. I dagsläget sparas data i olika databaser, men det finns inget gemensamt system där studier kan lagras och jämföras. Information om hur en parameter som beskriver en aspekt av ett biologiskt system varierar över tiden ska kunna lagras, detta var ett grundkrav. Prototypen designades utifrån en omfattande studie om 11β-HSD1, som är ett potentiellt mål för läkemedel mot diabetes. Både en prototyp för lagring av data samt ett gränssnitt för sökningar i systemet och visualisering av resultat byggdes. Slutligen testades prototypen samt gränssnittet och utvärderades tillsammans med forskare.

Examensarbete 20 p i Molekylär bioteknikprogrammet

Uppsala universitet, februari 2006

Table of content

1 INTRODUCTION

1.1 Databases for biological systems

New technology in biology and the big genome projects have created possibilities to describe all components in biological systems. The structural aspect of different biological systems are well described in for example the sequence databases. Another important aspect is the temporal aspect because time is always a parameter in biological processes. For instance different proteins interact with each other at different points of time, one mRNA is transcribed before another and glucose levels differ in time. At this point there are not many databases or other systems where this kind of information is stored, where biological systems are described with time as the main coordinate. If one wants to know what happens from a temporal perspective one has to read review articles or textbooks.

1.2 Biovitrum

Biovitrum is a biotech company that has research, development and commercial operations. The research and development is concentrated to the fields of diabetes, obesity and inflammation.

1.2.1 Pharmacology

Experimental studies on animals are performed at the department of pharmacology at Biovitrum. These research studies often last for a few days up to several weeks. During a study different parameters are measured at one or several points in time and lots of data are collected. For example parameters like weight and food intake can be measured daily, whereas liver samples can only be taken at the end of the study.

It should be noted that the information about a study has an essential temporal component. The treatment and measurements performed *in vivo* are always related to a specific time which is important to record. Therefore, this type of data seems to require a database that can handle temporal parameters.

1.2.2 Databases

The pharmacologists at Biovitrum store data from their studies in different databases. They have one database called AHS+ in which information about each animal is stored. For instance when the animal arrived, what kind of treatment it has been exposed to and when the experiment was terminated. The regulatory authorities require that this type of data be recorded for all *in vivo* work.

When a new study is planned a study plan has to be written. In a study plan the experimental set up of the study is described. The study plan is important because only the experiments specified in the plan may be performed. All study plans at Biovitrum are saved in a database called BERIT.

In the Electronic Lab Notebook (ELN) all experiments have to be entered. The reason is that when a company wants to get a patent it has to be able to prove when ideas came up and specific experiments where performed.

These databases are only loosely integrated with each other, at present the pharmacologists have no common system for storing all their research data. In none of these databases can all information on the design and results of a study be entered, viewed or compared.

1.3 The aim of the project

The aim of this project is to design, build and test a system where temporal data from biological processes can be stored.

The design of the system is based on a couple of experimental datasets. The information which is important that the system can handle is based on these datasets. From this information some suggestions to data structures should be proposed. The main thing the system must be able to store is information about how a parameter that describes an aspect of a biological system varies over time.

The system does not have to be a database because that would take a lot longer than the 20 weeks this project has. Building a prototype for storing the data and an html-interface is enough. In the interface it should be possible to search in the prototype for a key word. The results should be presented both numerically and graphically.

The last part of the project is to enter some experimental dataset and test the system. Together with researchers at the department the system will be evaluated to find advantages and disadvantages. These will provide the basis for improvements proposed in this report.

There is a general need to explore temporal databases in biology and medicine. Since it is well-known that temporal database design is a difficult area (Renolen 1999), it is important to choose a good test case. In this work, *in vivo* experiments, typically performed in biomedical research, was selected as the basis for the database design.

1.4 Geographical Information System

At the moment there is no common database for temporal processes in biological systems. There is very little written about the problem of storing temporal data from biological systems at all. I therefore had to look at other fields for information, ideas and inspiration about this problem.

A field that has been working with this problem for a while is Geographical Information Systems (GIS). There are many definitions of GIS because it has developed from a number of scientific areas. One definition is: “A computerized information system for handling and analyses of geographical data” (Eklund 2003). The temporal problem that has risen in GIS is the changing of maps over time. In a city for example new houses are built, others are torn down, there might be a flood in a river through the city and lots of other things can happen that effects the map of the city. If the map is redrawn for every change all information about the past is lost. This is a big problem if you are interested in the development over time in the city. Here the researchers have a temporal problem. They have different ways and ideas of how a database with the time as a parameter should be built, but so far they have not agreed on which way is the best (Galton 2004).

1.5 Presentation of a system in GIS

A system can be described either by a field-based or an object-based view.

1.5.1 Field-based system

In a field-based system the whole system is divided into spatial fields. These fields get a value from a mapping of spatial locations. A field gives an answer to a question: "What is the world like at such-and-such a place?" (Galton 2004). For instance the world would be described by a map with lines, defined by x-, y- and z-coordinates, that indicate fields for the different continents and countries. The name of the continents and countries will be the values of the fields.

Snapshots are a complete description of the system at a specific time point (Galton 2004; Renolen 1999). In a snapshot for a time point each field is assigned one value, for instance each area that indicates a country gets the value of the country's name. If the name of the country is changed then the field gets a new value and if the border is changed the field gets new coordinates. These changes will be seen in a new snapshot. The temporal development of a system can be described with many snapshots at different times. This is a very simple method, and a problem is that in cases where there are only small changes between the snapshots a lot of the data will be equal (Galton 2004; Renolen 1999). This means that the same data will be stored at many places and therefore occupy unnecessary space. The database will be bigger then necessary and searches in the database will be slowed down. The history of a field-based system can either be described by collecting the values for all fields for each time point, or collecting the value at every time points for each field (Galton 2004).

1.5.2 Object-based system

Another way to describe a system is by dividing it into different objects, an object-based system. A description of the world would for example have seven objects, one for each continent. Then each continent can be split into new objects one for each country in this continent.

An object has attributes that describes its states and operations that can perform different tasks. The attributes can be either temporal or non-temporal. There are two types of operations; observers and mutators. An observer gives a state of an object and mutator changes a state of the object (Renolen 1999). When a temporal attribute gets a new value, both the value and the time point of the change is saved. If it is a continuous change then the stop time and the rate also has to be recorded. With this information stored the history of a temporal attribute can be recovered (Renolen 1999).

1.5.3 3+1 dimensions or 4 dimensions

1.5.3.1 GIS

In the spatiotemporal GIS there is a problem because there are four dimensions, the x-, y- and z-coordinates and on top of that time. The world can be described either as a 3+1 dimensions, where the three spatial coordinates are described equally and the time in a different way, or 4 dimensions, where all coordinates are described the same way. The later version is more accurate but it is often easier to have a 3+1 dimension description.

In a field-based system there is not much difference between 3+1 and 4 dimensions, they are described in the same way (Galton 2004). In an object-based system Galton uses the terms *continuants* and *events* to describe the spatial and temporal objects respectively. In the 3+1 dimension description an object can only be one of these, either a continuant or an event. In a

4 dimension description Galton talks about *multi-aspect phenomena* objects; these can be described as both continuants and events. One example of this is a flood. A flood can be described as a continuant like a lake; it is a body of water located at a specific place. But it can also be an event because like an occupation a flood has a beginning, duration and an end (Galton 2004).

1.5.3.2 Biological systems

In a biological system the dimension problem also occurs. Let us consider the cell as an example. The x-, y- and z-coordinates in a field-based system may not be as clear here as in the geographical systems, but of course can a cell be divided into coordinates like a map. In the object-based system the spatial objects can be the cell parts like the nucleus, mitochondria or membrane. The temporal objects can be for example the replication of DNA, transcription of mRNA and ion concentrations.

An animal can be also described spatial with x-, y- and z-coordinates or spatial objects like the brain, liver and blood. Then the actions and values for each field or object can be specified.

At this early stage of the prototype we consider an animal as one whole object. The animal is a temporal object and we will have no spatial objects. One reason for this is that the *in vivo* experiments at Biovitrum always use a group of animals. The group can be considered the molecule and the animals the atoms that build the molecule. Another reason is that the focus is on the time dimension. This way the number of dimensions is reduced to one.

2 DESIGN OF THE PROTOTYPE

To help with the design of the database prototype an in vivo test case was needed. We decided that the published Biovitrum dataset about the enzyme 11β-HSD1 would be most appropriate. This dataset is believed to be reasonably representative to work as a test case.

2.1 11β-HSD1 role in type 2 diabetes

The enzyme 11β-HSD1 catalyses the interconversion between cortisone and cortisol and the enzyme 11β-HSD2 catalyses the opposite reaction, figure 1. 11β-HSD1 is a target that has been suggested for type 2 diabetes mellitus drugs (Alberts *et al.* 2003; Stuling and Waldhäusl 2004). Patients with type 2 diabetes have a loss of insulin sensitivity in important target tissues, such as liver, muscle and adipose tissue. It is know that insulin is an antagonist of glucocorticoids (Stulnig and Waldhäusl 2004). Cortisol is the active form of the two glucocorticoids (Alberts *et al.* 2003; Stuling and Waldhäusl 2004). In rodents 11β-HSD1 converts 11-dehydrocorticosterone into corticosterone (Alberts *et al.* 2003).

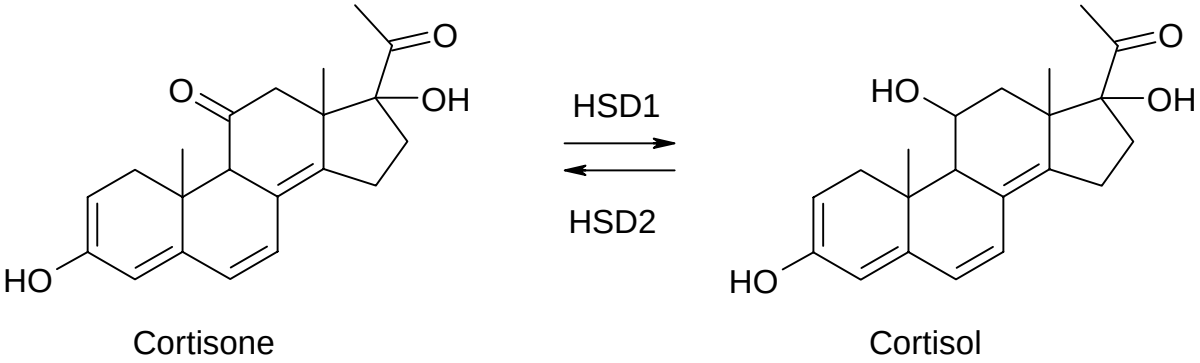

Figure 1: 11β-HSD1 catalyses the reaction from cortisone to the active form cortisol and 11β-HSD2 catalyses the opposite reaction of the glucocorticoids.

2.2 The studies on 11β-HSD1

At Biovitrum research studies have been done on an isoform-selective inhibitor of mouse 11β-HSD1 3-chloro-2-methyl-N-{4-[2-(4-methyl-1-piperazinyl)-2-oxoetyhl]-1,3-thiazol-2-yl} benzenesulfonamide (BVT.2733). The purpose of this study was to see if inhibition of 11β-HSD1 in mouse models of type 2 diabetic, can affect blood glucose levels, glucose tolerance and insulin sensitivity (Alberts *et al.* 2003).

To see what information that is important that the prototype can store a study has to be examined closer, both the design of a study and the data that come out from the study. The animals were administered doses orally twice a day, with 12-hours interval at 07.00-08.00 and 19.00-20.00 for three or four days. The dose was either the chemical substance BVT.2733 dissolved in the vehicle (the solvent) (12% beta-hydroxypropylcyclodextrin and 0.3% sodium chloride) or only the vehicle (Alberts *et al.* 2003). The animals that were given only vehicle are part of a control group. In a study the animals can be of different kinds. In this study all were mice, but there were four different mouse strains; C57BL which are a normal mice, and ob/ob, db/db and KKAy which are all hyperglycaemic and hyperinsulinemic (Alberts *et al.* 2003). The animals were kept one per cage with light on for 12 hours and in dark for 12 hours, the light was turned on at 05.00-06.30. The temperature in the cages was 22º ± 1º C. The animals were divided into groups based on their 4 hour fasting blood glucose, 4 hour fasting blood glucose and plasma insulin or body weight (Alberts *et al.* 2003).

The study on 11β-HSD1 consists of smaller studies. In the studies the strains of the animals and what kinds of experiments are performed differ. All this is carefully described in the study plan. The different experiments performed in these studies where glucose, serum cholesterol, triglyceride and free fatty acid analyses, oral glucose tolerance test (OGTT), mRNA analysis and clamp studies. For technical details see Alberts *et al.* 2003. Most of these experiments were done after three or four days, during which the animals have been treated with compound or only vehicle. For some of the experiments, like OGTT, the animals where fasted for a while before the test. During the studies parameters like body weight and food intake were measured daily (Alberts *et al.* 2003).

This study shows that BVT.2733 lowered circulation glucose and insulin levels in three different mouse models of type 2 diabetes, and not in normal mice. This supports previous suggestions that selective 11β-HSD1 inhibitors may help patients with type 2 diabetes to lower their blood glucose (Alberts *et al.* 2003).

2.3 What the prototype should be able to handle

2.3.1 Input

Into the prototype should the data from the different studies be entered, both information about the design and the actual measurement data.

As mentioned before, it is important that the prototype is able to handle temporal data. The temporal data can be of two types either periods or values at specific time points. For instance if the light is on in the cages from 05.00 and for 12 hours forward every day and off the rest of the time, it means that it is an interval of light from 05.00-17.00. Then this can be entered as a period with start time 05.00 and end time 17.00, with the value of "light", then another period from 17.00-29.00 with the value of "dark" and so on for all days in the study.

The other type of temporal data takes care of the parameters that are measured at a specific time point. Here we can take the body weight as an example. If the weight of an animal is measured daily then this parameter should get a new value every day linked to the time point of the measurement.

Important for both types of temporal data are that the old values have to be saved when a new value is entered. Otherwise the history of the parameter will be lost and it will not be a temporal parameter.

The prototype should also be able to handle general information that is not of a temporal nature. For example the name of the study plan, what the species and strain of the animal are, the name of the target, food type and what kind of compound and vehicle it is. Exactly which parameters that are important will be discussed with the pharmacologist.

2.3.2 Output

The output from the prototype should be both the design of a study and the values of the parameters obtained in the study. A graphical presentation of the design and the time schedule of a study in an interface are desirable. Then differences between different studies will easily be viewed. The interface should also have both a graphical and numeric presentation of the parameters.

2.4 Structure of the prototype

After reading the articles about GIS and 11β-HSD1 a couple of suggestions for the structure of how the data from a study could be stored were made. An object-based description of a study felt most natural. But what kind of classes and how they should be ordered was not obvious.

2.4.1 Alternative 1

The first alternative has a top class, called Study, which describes the general parameters for the study such as Study number, start date and general conditions. Then it has subclasses, called Organism, where all information about each organism is saved, species, strain, genotype, observations and specific conditions, see figure 2 A.

2.4.2 Alternative 2

In this second alternative a study is divided into three levels. Here the topclass is also Study, but under Study comes subclasses called Group and last Organisms. In Group are parameters that are the same for a group of organisms. And in Organism are only individual parameters like weight, age and glucose values, see figure 2 B.

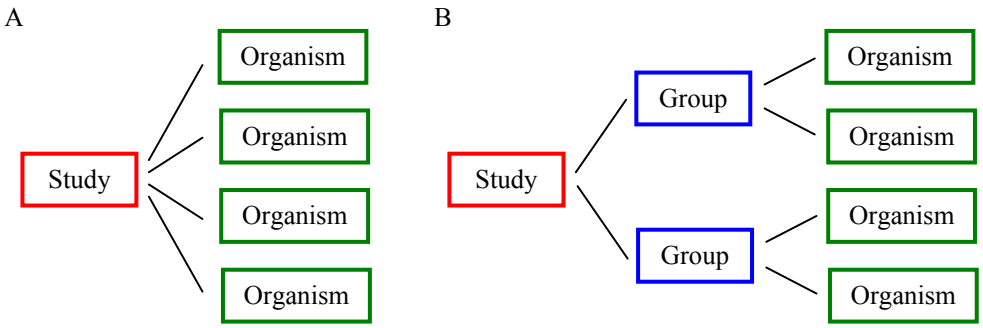

Figure 2: There are two alternative of the class structure. Either two levels; Study and Organism (A), or three levels; Study, Group and Organism (B)

2.5 Design issues

An event is something that happens to the organism for example change in temperature, starvation or dose administration of a substance. But is a measurement of a parameter also an event, or is that something different? A measurement is more of an output then an input. What seems most natural?

At Biovitrum animal experiments are done on mice and rats. Are there big differences between how these animals are treated and on what kind of parameters that are measured? The question is whether a mouse and a rat can be described by the same class or if there has to be subclasses of Organism. A question concerning the DIO-animals also came up. A DIO-animal is an animal that has been fed fat food its whole life. Should it be indicated from the beginning that it is a DIO-animal or can that be seen from the food intake and weight?

These are some of the questions that arose during the search for a good structure of the prototype. Another question was which parameters are necessary and which are not, and to which class should they belong? To find answers to these questions, interviews with pharmacologists where done. The pharmacologists also gave suggestions and ideas about the prototype and the interface.

2.6 Results from the interviews

The interviews with the researches gave me answers to the questions in chapter 2.5 Design issues. They also gave me ideas and suggestions how to design the database prototype.

2.6.1 Answers to the questions

The interviews resulted in the following answers:

- Measurements of a parameter can be considered to be events. A difference between input and output is not necessary.
- There is no big difference between a mouse and a rat regarding their parameters and other attributes. Therefore, a mouse and a rat can be instances of the same class. If the

mouse or rat is a DIO-animal this should be indicated at the implementation, since it can be hard to see if it is a DIO or not from the measurement data.

The question about the parameters will be answered later when the structure of the prototype is described, see chapter 3.1 Structure

2.6.2 Suggestions from the pharmacologists

The following suggestions were given by the pharmacologists:

- The data from every animal must be stored, not only the mean values and statistics. If only the standard deviation describes the spread around the mean value a lot of information will be lost. Let us take the example of glucose values that are measured over time. If the standard deviation for the values at all time points are high, then it is interesting to see if it is the same animal or animals that contribute outlier values, or if the spread is genuinely large for all animals.
- Other things that came up were that it is more important what kind of food the animals are fed instead of the time it is given. There are different brands of food and they can vary in sugar and fat content.
- It is also important to know from what breeder the animals are. There might be some differences between the same kinds of animal if they come from different breeders.
- There are indications that the social environment can affect the animals. Therefore it can be of interest to know if the animals where kept by themselves or together with other animals in cage.
- It is not necessary to store every detail about a study in this prototype. Instead there can be links to other databases, for example BERIT, which can be used to find more information about the study.
- At Biovitrum different kinds of studies are performed, some kinds are common while others are rare. It should only be to most common types of studies that are stored in the prototype. According to the researchers it is not useful to add rare studies. These studies will complicate searches and comparisons in the prototype.

3 BUILDING THE PROTOTYPE AND THE INTERFACE

3.1 Structure

After discussion with the pharmacologists and the project supervisor a structure for the organization of the data was developed. In the normal studies there are groups of organisms that are exposed to the same treatment. Therefore alternative 2, figure 2 B, is the best structure to use for this prototype. The conditions that are the same for all animals in a group can be stored in attributes of the Group. All organisms in this group will then inherit these. If alternative 1 would be used, equal data are stored in many Organisms. This takes up unnecessary space. The specific parameters for each animal can be stored in attributes in Organism.

3.1.1 Tprofile

Tprofile is the class that handles the temporal objects. A Tprofile stores the values of a property of an object linked to time points, and therefore allows the history of the object to be accessed easily. The value of the time is hours, one time step is one hour. These are the attributes:

- *id:* A string with the name of the instance.

- *tProf*: A dictionary with the time points and the values of the instance. In *key* is the time point and in *value* is the value at that time point.
- *period:* A list of tuples with the times and values for an interval: (start time, stop time, first value, last value)
- *unit:* A string with the unit of the Tprofiles values.
- *tDiff:* A float that gives the difference between the first and last time point in the Tprofile.

An instance of a Tprofile has the following functions:

- *addEvent(t,value):* Adds a new value at time point t to the attribute *tProf*
- *addPeriod(t1,t2,v1,v2):* Adds a new interval to the attribute *period.* The interval is between t1 to t2 with the start value v1 and end value v2. It also adds corresponding events to the attribute *tProf*
- *getValue(t='none'):* Returns the value of the Tprofile at time t, if no t is given the first value is returned and if *tProf* does not have a value for t "no value" is returned.
- *getTimeDiff():* Returns a list [tmin, tmax, tmax-tmin] where tmin is the first time point of the Tprofile and tmax the last.
- *write():* Returns the values of *period*, if it exists, otherwise the values of *tProf.*

Temporal data can be of two types, either values at specific time points or values for intervals. For the first type the values are saved in the attribute *tProf*. For instance the weight of an animal is measured every morning, so at a specific time point every day the weight gets a new value. This value is saved in a *tProf* with the key as the time point and the value is the weight at this point. In this dictionary the weight for the animal throughout the study is stored and the development over time can be viewed.

If the parameter is measured over an interval the data is saved in the list *period.* The data is stored in the form of a tuple with four items: start time of the interval, t1, end time of the interval, t2, the value at t1, v1, and the value at t2, v2. For instance if the animal is starved for a period of time t1 is the time of the start of the starvation and t2 the time the animal is fed again. In this case both v1 and v2 are "fast". Even though it is an interval and the data is stored in *period* it is also be stored in *tProf*. The reason for this is that it will simplify searches at specific time points in the database in the future. In the *tProf* a value is added at every hour of the interval. The interval is approximated to be a linear change from the first value to the last and the values that are added to the *tProf* are calculated from this.

3.1.2 Study

The most general information about the dataset is stored in the study. Study has the following attributes.

- *id:* A string with the study plan number, or equivalent information for the study.
- *target:* A string with the name of the drug target.
- *startDate:* A list with the date of the start date of the study [year, month, day], for example [2005,10,4].
- *light_dark:* A Tprofile with the intervals of the time the animals have light respectively darkness.
- *temperature:* A float with the planned temperature in the rooms where the animals are kept.
- *information:* A string where any kind of information about the study can be stored.

- *groupDict:* A dictionary with the groups that belong to this study. The keys are the ids of the groups and the values are pointers to the right group.

3.1.3 Group

The subclass Group has Study as superclass. Here the parameters, that are specific for a group of animals in an experiment, are specified.

- *id:* A string with the name of this group of animals
- *study:* A pointer to the specific study that this group belongs to.
- *dose:* A Tprofile with time points for administration of substance and information about the dose.
- *substance:* An instance of the class Substance which describes the substance, such as compound, and its unit, type of vehicle and how the substance is administrated.
- *strain:* A string with the type of strain for the animals in this group.
- *breeder:* A string with the name of the breeder of these animals.
- *dio:* A string that is "yes" if the animals are DIO-animals, otherwise "no".
- *sex:* A string with the sex of the animals.
- *FoodType:* A string with the type of food the animals are fed.
- *noOrginBox:* A float with the number of organisms there are in each cage.
- *orgDict:* A dictionary with the organisms that belongs to this group. The keys are the ids of the organisms and the values are pointers to the right organism.
- *orgParam:* A dictionary with all parameters that are measured for the animals in this group. The keys are the names of the parameter and in the corresponding value is a list of the organisms that have this parameter.

Whenever an instance of the class Group is made, the group is added to the groupDict of the study that it belongs to.

3.1.4 Organism

The subclass group has Group as superclass. Here the individual animal is described.

- *id:* A string with the name or number that defines this animal.
- *group:* A pointer to the specific group that this organism belongs to.
- *parameters:* A dictionary with the parameters for this organism. The keys are the names of the parameters and the values are Tprofiles for the parameteres.
- *age:* A Tprofile with the age of the animal. The age is given in weeks.

Whenever an instance of the class Organism is constructed, the organism is added to the orgDict of the group that it belongs to.

3.2 Interface

3.2.1 Making a search

At the start page of the interface, searches in the system can be made. The user can type in a key word for the search and specify what type of key word it is. There are four alternatives:

- *Study:* If the key word is the id of a study.
- *Substance:* If the key word is a name of a chemical substance.
- *Measurement parameter:* If the key word is measurement parameter.
- *Other:* If the key word is none of the above.

There are two buttons that specify the types of searches that can be executed:

- *Search:* This button starts a search for the given key word.
- *List all:* This button starts a search for all studies, substances or measurement parameters in the prototype. It does not work for the type "Other".

The search function first checks what kind of key word it is. In case it is one of the first three alternatives the search is rather easy. The function knows where in the studies it should look for the key word. It is more difficult when it is of the type "Other". The function then has to go through all studies, groups and organisms and compare the key word to all their parameters and the values of the parameters. Since some of the values are lists, tuples or even tProfiles the function also has to check these values' items or parameters. The search function returns the names of the group and study for the hits, it also returns in what attributes the key word were found. If the hit is not a study the group is left out.

All the hits are listed in a new html-page. Here the user can choose which hits he or she are interested in. Depending on the type of the hits the results are shown in different ways. If it is studies then all studies are listed with some general information, and the list of the groups that belongs to this study and some data about each group. There are links to pages about both individual studies and groups. If the hits are groups a page with a list with short information about the groups and links to their respective group page will open. For hits that are organisms a similar page will open with the groups that the organisms belongs to. In both the result pages the user can choose to make plots with groups from one or more studies, see chapter 3.2.3 Plots.

3.2.2 Pages with information

3.2.2.1 Study page

At the top of this page is some general information about the study and a table with the attributes of the study and their values. Below this is a new table and in the first column are the names of the groups listed. These names are also links to pages about the groups. In the other columns are the attributes of the groups. The user can also choose to draw a plot of one of more of the parameters of the groups. For more details about the plot, see chapter 3.2.3 Plots.

3.2.2.2 Group page

A table with the attributes of the group, is shown on the top of the page about a group. The only differences between this table and the table in the page about the study are that here it is only the actual group listed and a new column is added. In this column is the name of the study that the group belongs to, and this name is a link to the study page. Even at this page the user can draw plots of the parameters, for more details see chapter 3.2.3 Plots. At the bottom of the page is a list with all the organisms that belong to this group. The names are links to a page about the specific organism.

3.2.2.3 Organism page

An organism is described in a page that starts with the id, group, study and age of the organism. The group and study name are each links to pages about themselves. The age is specified in weeks from the age at the start to the end of the study. There are also two tables at this page. The first with the numeric values of all parameters of this organism that is not

periods. In the first column are the different time points for measurements and in the other columns the parameters and their values. In the other table are the periods listed. For each kind of period are the different intervals with the belonging value listed.

3.2.3 Plots

When one group or several groups are picked for plotting, all the parameters for the organisms in the group are listed. The user can then choose which parameters he or she wants to plot. In the group page where it is only one group that is plotted the user gets to choose what should be plotted. It can either be all organisms values, the mean value of the organisms or both all organisms values and the mean value. If the plot is made from another page where it can be more than one group, it will always be the mean values of the organisms in each group that are plotted. If the values are nonnumeric the most frequent value is the mean value. If it is both numeric and nonnumeric, the values are separated in two lists. Then the mean value is calculated from the list with most items.

After the parameters have been chosen, a plot with the values on the y-axis and the time on the x-axis will be plotted in a new window, figure 3 A. The plot will be a scatter plot with lines between the markers except for the parameters that have *periods.* If both all organisms and the mean value are plotted it will only be a line between the mean values. For these, special period markers, will indicate the intervals. All markers are links to either a page about the group (the mean value markers) or the organism (the others).

A problem with this kind of plotting is that it might be a big difference in the time span of the different parameters. Then it will be hard to look at the parameters with the smaller time span. There are different ways to solve this problem, but in this interface the following solution was chosen: In case the time span for a parameter is less then 5% of the maximal time span there will be a link to another page where just these parameters are plotted in a new plot with the shorter time span, figure 3 B.

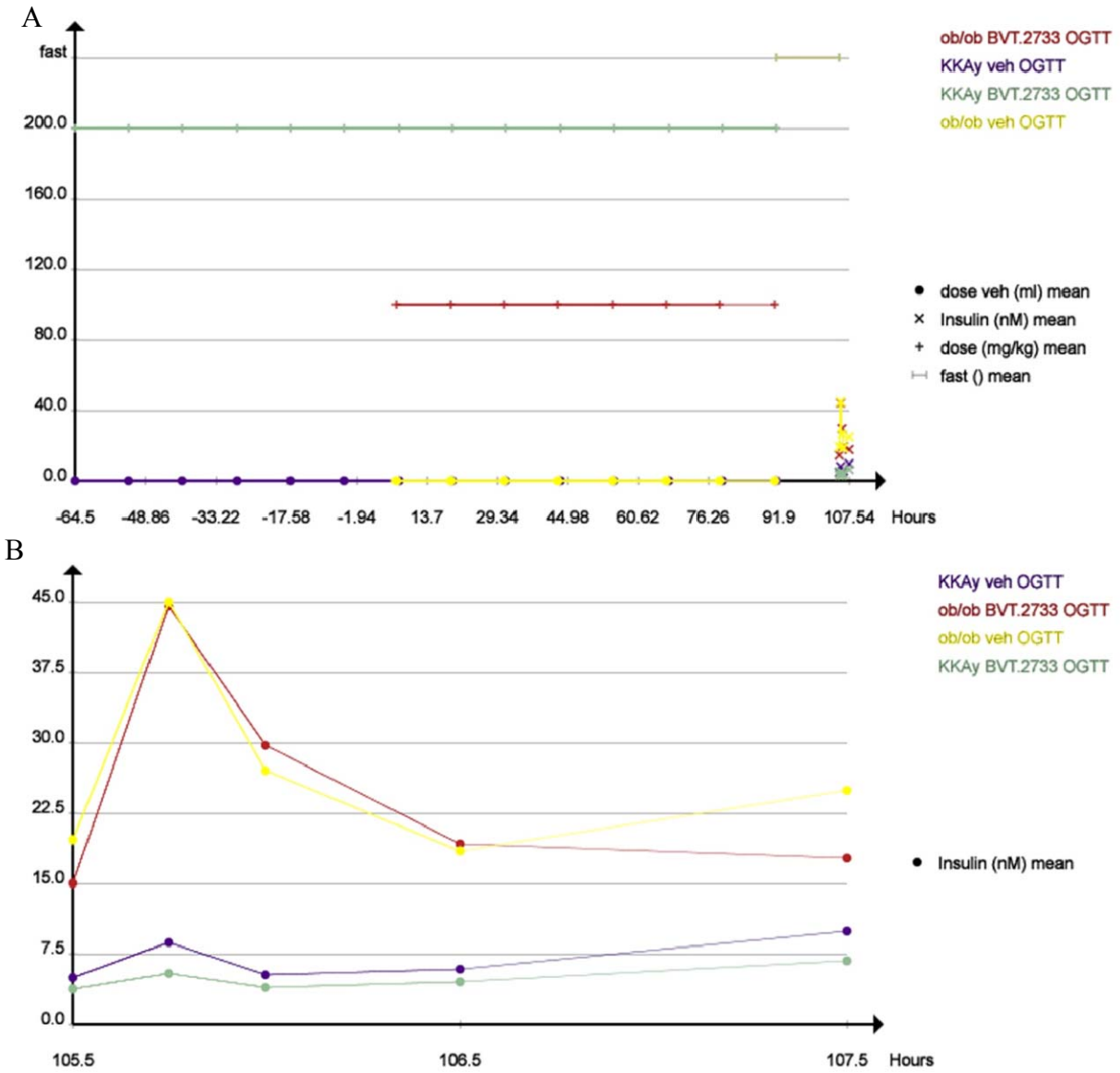

Figure 3: Plots from the interface. The parameters fast, dose, dose veh and insulin are all plotted together. The vehicle dose is 0.4 ml, which is the reason why it appears to be zero (A). Since fast is an interval it has a special kind of marker. Insulin has a much shorter time scale than dose and the markers are blurred together. Therefore insulin is plotted in a separate graph, with a different time scale (B).

Another problem arises when measurements on lots of parameters are taken at the same time, for example at the end of the study. In these cases it might be hard to see which values that belong to which parameter. Therefore, these parameters are plotted in a new graph, where the y axis is the same, but the different parameters are plotted on the x-axis instead of the time, figure 4. This makes it easier to see the values for the different parameters.

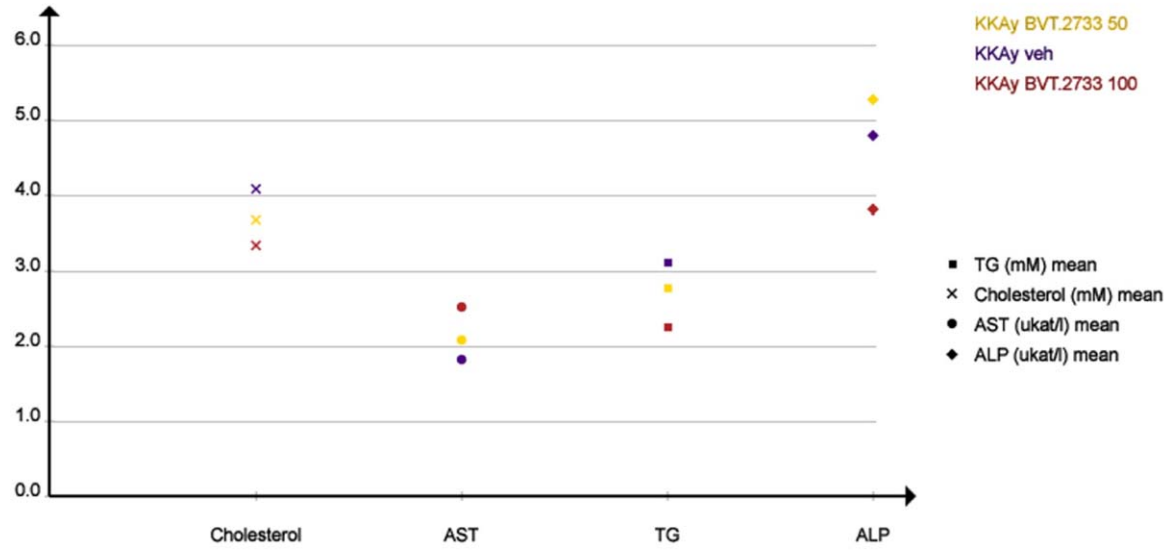

Figure 4: The cholesterol, AST, TG and ALP values are all measured once. The samples are all taken at the same time, at the end of the study. In a plot with the time on the x-axis it is hard to separate these parameters; therefore they are plotted in a new plot with the parameters on the x-axis.

3.3 Technical solutions

The prototype code is written in Python which is an object based language. This project is a prototype of a temporal database, a first draft of the structure of the database. This does not mean that a whole database has to be built. In this early stage the focus is on the structure and not the design. Instead of making a SQL database or some other type, the data is stored in something Python calls Pickles. Data can be saved as a collection of Python objects and pickle is one kind of this. The data can later be unpickled and that recreates the data as Python objects that can be used.

Since there are no predefined ways to make plots or diagrams in Python all plot functions had to be made from scratch. The plots in the interface are drawn with Scalable Vector Graphics, SVG, which is a XML-based graphic standard. With SVG you define a square to draw in. Then lines, curves and text can be drawn in this window. For each line, start point and end point, is described by x- and y-coordinates. All attributes can be specified for each element, like the thickness of the stroke, the type of the text and the colour of the stroke and the filling (Eisenberg 2002). Using these components a template for a diagram has been made with axes, ticks, labels and different kinds of markers. This template is used for all plots in the interface.

Both Python and SVG readers are Open Source software, which is convenient. Another reason to use Python and SVG is that they make it rather easy to build the prototype and the interface.

3.4 Assumptions and simplifiers

To simplify the prototype the dimensions are reduced from four, three spatial and one temporal, to one. The prototype has no spatial objects.

In this prototype it is assumed that all studies have similar design, the organisms in a group have the same kind of treatment throughout the whole study. There is a study design called Modified Cross-over used in telemetry. In an example study of this kind there are four different organisms and four different kinds of treatments. The treatment of an organism differs every time, so all organism gets all treatment at one time, see table 1. In a study like this a lot of external conditions can be faced out. A study of this kind cannot be entered in the prototype database, since the prototype cannot handle different kinds of treatment of the same organism. But at this stage we are only interested in entering "normal or typical" studies.

Table 1: A schema over the type of treatment the organisms will get each day. The organisms are treated with four different kinds of treatment a new one every time.

Day	Organism 1	Organism 2	Organism 3	Organism 4
1	Vehicle	Type 1	Type 2	Type 3
2	Type 3	Vehicle	Type 1	Type 2
3	Type 2	Type 3	Vehicle	Type 1
4	Type 1	Type 2	Type 3	Vehicle

Another assumption made is that the intervals in Tprofile are always a linear change from the start value to the end value. This may be right in a few cases, but for most attributes it is just an approximation.

4 TESTING THE PROTOTYPE AND THE INTERFACE

After the prototype was developed and the data entered, the prototype and the interface were tested by me, my supervisor and a researcher. During the testing different faults and limitations of both the prototype and the interface were found. Some of these were easy to fix and have been taken care of whereas others demands more work and there is not enough time. In the next section the limitations and improvements are discussed and suggestions to solve the problems are given.

4.1 Limitations and improvements

4.1.1 Deviations of temporal objects

There is no possibility to add deviations to Tprofiles right now. This is something that has to be added, especially for Tprofile that are attributes to the higher classes. For instance, if all animals in group A are supposed to get 200 mg/kg substance, this is implemented in the group's attribute *Dose.* But if something goes wrong at administration at time point T and animal org1 gets 100 mg/kg instead of 200 mg/kg substance, this has to be recorded. If the value of *Dose.tProf* at T is changed to 100 mg/kg in group A, this will change the value for all organisms in group A. This is not right, since all except for org1 got 200 mg/kg.

To solve this problem there has to be some other attribute where deviations can be stored. This can be an attribute in Tprofile. A dictionary called *Deviations* with the id of the organism that has the deviation linked to a list. In this list tuples are saved with the wanted time, time of the deviation, the wanted value and the new value. The function *getValue(t='none')* in Tprofile has to be changed in this case. This function first has to check if there is a deviation at the desired time and in that case give the new value instead of the supposed value. In the plot functions this also has to be noted so it plots the value of the deviation, if there is a deviation.

4.1.2 The search function

4.1.2.1 Perfect hits

There are many improvements that need to be done in the search function before a final database can be made. Right now only words that are a perfect match will be a hit. For instance, if the user is interested in all studies where any type of glucose value is measured there will be a problem. If he or she types in "glucose" for a key word the studies that have for instance "blood glucose" as a parameter will not be a hit.

A way to fix this is if the search function divides every string into individual words and compares each word to the key word. A problem with this method is that there might be some false positives. For instance if it in the Study attribute *Information* says: "In this study the insulin will be measured but not the glucose. We are at this point not interested in the glucose levels." Then this study will be a hit even though this study does not have a parameter for any kind of glucose levels.

4.1.2.2 Numeric searches

A similar problem occurs when searches on numerical values are done. If the key word is "1", only perfect matches will be hits and "1.0" will not be a hit. This is wrong, but the problem is that the key word is always stored as a string. The string "1" is not the same string as "1.0" even though there numerical values are the same. Therefore the type of key word that contains numerical values has to be changed from a string to float.

4.1.2.3 Detailed searches

Another search limitation that is not solved yet is if you want to search for a specific value on a parameter. Right now searches can only be made on the name or value of the parameter. If the user is interested in all studies where there have been two animals in the same cage he or she can either search on "noOrginBox" or "2". The first alternative will find all groups in the prototype as hits, since all group instances have the attribute noOrginBox. The second variant will result in a list of hits with all groups that have noOrginBox = 2, but also all other groups with parameters with the value "2". Neither of these alternatives are good, it would be an advantage if both name and value of parameter can be specified if wanted. Even better would be if an interval for the value can be specified in the search function. For instance if the user wants to find all groups that have two or three animals in the boxes, 2 < noOrginBox < 3, or all groups with more then 3 animals in the box, noOrginBox >= 3. It would also be good if the time could be specified, for example if you want to find all organisms where glucose values have been measured 3 days after the start of the study. To make these kinds of searches possible both the interface and the search function has to be improved. The interface must expand so a search can be more specified.

4.1.3 Input of data

The input of data into the database is a big problem; so far it has been made manually. The experimental data stored in the prototype came from Excel documents with different structures. There is no standard for how the data should be stored; this complicates the input a lot. If there were a standard for how the Excel documents should look, then the data could be read directly from this document. But it might be hard to find a standard that will work for all kinds of different studies. If there are exceptions from the standard in the documents it will complicate the reading of the files. Perhaps a few alternatives for the most common types of studies can be composed.

Another way to store the data in the prototype is to let the user type in the data directly. There might be some problem with studies that do not fit in to the frames. An advance with this method is that it is easier to standardize the values than in the other method. For instance the class Group has an attribute for the breeder of the animals; this is a string that can contain any kind of text. One user might type in: "Taconic Farms" another "Taconic Farms, Denmark" and a third "Taconic Farms (Ry, Denmark)". Then a search on the key word "Taconic Farms" will only get the first study as a hit even though all these are the same breeder, see chapter 4.1.2.1 Perfect hits. This problem can be solved with a popup menu of alternative breeders, this way they all have the same name for the same breeder. Of course there has to be a way to give a new alternative in case there is not one that matches the user's breeder. When a user types in a new alternative this should be added to the list in the popup menu. These kinds of popup menus should be available for all attributes that have standardised answer. But how do you know if an attribute has standardised answers or not? That is a question that I do not have an answer to right now.

If popup menus will be used for input of data, then they can also be used for searches. In case the user wants to search on a parameter that has standard alternatives then he or she should be able to choose from a popup menu. This is a way to get around the problem discussed above with the search on "Taconic Farms"; the key word will be exactly the same as the names in the prototype.

4.1.4 Output of data

4.1.4.1 Statistics

In the mean value plot there should be some kind of statistics. They could also have the standard deviations, not only the mean value. This gives a more accurate picture of the values. This is most important in the plots with different groups. Because in this plot there is no possibility to plot all organisms, it is always just the mean value. The spread around the mean value can easily be seen when both all organisms' values and mean values. This can be calculated at the same time as the mean values.

The number of organisms, *n*, may be different for the different parameters in the same group. Due to some kind of error sometimes values for all parameters for an organism is not available. Therefore the value of *n* must be shown somewhere, perhaps in a text when the mouse pointer points at a marker for a mean value. In the plot the standard deviation can be indicated by a vertical bar centred on the mean value.

4.1.5 Design improvements of the interface

The focus of this project has not been on the design, therefore there is not much effort put on the interface. Here follows some suggestions for improvements on the design of the interface:

- An overall improvement in the layout of the interface, which makes it more user friendly.
- At all the pages where there are checkboxes there should be a button or checkbox called "Mark all" that checkmarks all checkboxes.
- At the start page there is a button called "List all". This button does not work if the search type is "Other"; therefore this button should be made unable when "Other" is chosen.

- When the mouse pointer is pointing at a marker in the svg of a plot the value of the data point should be visible. This can be done with the svg-commando “onmouseover”.
- Add a link to BERIT, the database with the study plans, to the page about a study. Then more information about the study can easily be found. The link could be to the database, but preferably to the right study in BERIT.
- It would be good if the choice of parameter for plotting can be saved when altered between the page about a study and a group and the other way around. It you are looking at a plot of a study and find a group that you are interested in looking more closely at, you click on the marker and are linked to the group page. Here you want to see the values for all organisms in this group. Then it would be good if the parameters chosen in the study page where already checked.
- A problem arises when the tick labels on the x-axis are long. If the names of the tick labels are longer then the space between the ticks, the names are written on top of each other. To solve this problem the labels can either be written on different y-coordinates every other high, then the next one low and so on. Or the labels can be written vertically instead of horizontally.
- Another small improvement that can be done to give a better overview of a time dependent graph is to divide the x-axis into steps of 12 or 24 hours depending on the time scale. This will make it easier to understand the time span. Of course this should not be done on the plots that have a short time span; these should still have the x-ticks in hours.

4.1.6 Other improvements

The attribute temperature in Study is a float. Instead this should be a Tprofile that describes the temperature in the rooms at different time points. If the goal is to keep the temperature stable at a certain temperature this will be the normal. And only deviations from the norm will be saved.

When the interface draws the plots the different SVG files are all saved in a folder called “Svg-files”. The program has to make sure this folder exists when it starts, otherwise create a new folder by this name. This folder also has to be emptied when the program is turned off.

In the pages about a specific study or group, tables can be added with numeric values about the parameters of the groups. The table should contain the mean values and statistic information like the standard deviation of the numeric parameters. Right now the only way to see this information is to look in the plots, but sometimes it is interesting to see the actual numeric values.

A request came concerning the Organism page. It would be good to have the ability to plot the values for just this organism. It should be a similar plot to the group plot, but with just one organism.

5 CONCLUSIONS

The project has proceeded well and even though it is far from completed it is a good start. Data can be entered to the prototype. The interface can be used to view entered data. So far there is not a flashy interface, but the basic data can be viewed and compared. The design of

prototype seems to be working and can be a good start for further work. The next step should be to build a real database; this has to be done before the prototype can be used at Biovitrum.

To make the prototype more general, improvements have to be made. This is necessary to make it suitable for other types of data, for example cell biological and human clinical data. This generalization of the prototype is important, since there are currently no common systems available. This project has aimed at testing some approaches to solving the issues concerning temporal databases.

6 ACKNOWLEDGEMENTS

I would like to thank my supervisors Per Kraulis and Stephen James for making this project possible, and the researchers Göran Selén, Maj Sundbom and Claes Carneheim for discussions, answers to my questions and testing of the prototype. I also wish to thank everyone at the Department of Pharmacology at Biovitrum for good company.

7 REFERENCES

1. Eklund L 2003 Geografisk informationsbehandling - metoder och tillämpningar, 3rd edition, Formas, Stockholm, Sweden.
2. Galton A 2004 Fields and Objects in Space, Time, and Space-time. *Spatial Cognition and Computation* 4:39-68.
3. Renolen 1999 Concepts and Methods for Modelling Temporal and Spatiotemporal Information. Ph.D. thesis, Norwegian University of Science and Technology.
4. Alberts P *et al.* 2003 Selective Inhibition of 11β-Hydroxysteroid Dehydrogenase Type 1 Improves Hepatic Insulin Sensitivity in Hyperglycemic Mice Strains. *Endocrinology* 114:4755-4762.
5. Stulnig TM, Waldhäusl W 2004 11β-Hydroxysteroid dehydrogenase Type 1 in obesity and Type 2 diabetes. *Diabetologia* 47:1-11.
6. Eisenberg DJ 2002 SVG Essentials, 1st edition, O'Reilly & Associates, Inc, Sebastopol, CA, USA.