



ELSEVIER

Artificial Intelligence in Medicine 6 (1994) 459–482

Artificial
Intelligence
in Medicine

An object-oriented approach to knowledge representation in a biomedical domain

Marco Ensing^a, Ray Paton^b, Piet-Hein Speel^{a,*}, Roy Rada^b

^a *University of Twente, Dept. of Computer Science, Postbus 217, 7500 AE Enschede, The Netherlands*

^b *Department of Computer Science, University of Liverpool, P.O. Box 147, Liverpool, L69 3BX UK*

Received January 1994; revised April 1994

Abstract

An object-oriented approach has been applied to the different stages involved in developing a knowledge base about insulin metabolism. At an early stage the separation of terminological and assertional knowledge was made. The terminological component was developed by medical experts and represented in CORE. An object-oriented knowledge acquisition process was applied to the assertional knowledge. A frame description is proposed which includes features like states and events, inheritance and collaboration. States and events are formalized with qualitative calculus. The terminological knowledge was very useful in the development of the assertional component. It assisted in understanding the problem domain, and in the implementation stage, it assisted in building good inheritance hierarchies.

Keywords: Knowledge acquisition; Object-oriented modelling; Medical terminology

1. Introduction

The building of knowledge bases is a very difficult task. The work reported in this paper illustrates the value of separating terminological knowledge from assertional knowledge when building a knowledge base. An object-oriented approach to the design and construction of the knowledge base has been followed to facilitate the exploitation of the distinction between terminological knowledge and assertional knowledge.

Biomedical processes are explained in many textbooks by models in which organs are sending messages to each other. These organs can easily be seen as the

* Corresponding author. Email: Speel@cs.utwente.nl

objects in an object-oriented system. Object orientation has proved to be an elegant and powerful way to organize knowledge [9,24].

Because of the object-oriented approach to the development of the knowledge based system (KBS), the actual development stages differ from the traditional ones. Usually, in the knowledge acquisition process for developing a KBS, the following stages are distinguished: elicitation, conceptualization, formalization and implementation [7]. In the elicitation stage, the knowledge of the domain is elicited from textbooks, journals and domain experts in an informal, usually verbal form. In the conceptualization stage, the elicited data is interpreted, which means that the concepts are analyzed and their implicit relationships are made explicit. This results in the construction of a conceptual base. The formalization stage involves the translation of the conceptual base into a representation using a formal knowledge representation system (KRS). The syntax and semantics of the knowledge is specified. The syntax defines the sentences which are legal in the language and the semantics defines the relation between sentences and the conceptualization of the world [20]. In the implementation stage, the formal representation is implemented in a computer-executable language.

An object-oriented approach in the knowledge acquisition process, distinguishes the following stages: domain characterization, object-oriented modelling, formalization and implementation [4]. In the domain characterization a great mass of detailed information is obtained, and organized under different points of view of the domain. The domain characterization of the blood glucose regulation was already done in the MEKAS project. *Object-oriented modelling* (OOM) results in several tables which list the objects and the relationships between objects. After OOM of the domain, the constraints on attributes and behaviour of objects are formalized. Furthermore, the communication between objects has to be formalized in order to analyze the overall behaviour of the system. Because the emphasis is on validating an object-oriented approach and not on the granularity and coverage of the KBS itself, a small domain was chosen. This made it possible to go through the successive stages in developing a KBS in reasonable time. The emphasis is on the object-oriented knowledge acquisition of the assertional knowledge.

To distinguish between the terminological component and the assertional component the following convention is used: *concepts* refer to the terminological component and *objects* refer to the assertional component. Fig. 1 clarifies the convention. In the terminological component the organ 'Liver' is referred to as the *concept* 'Liver'. The meaning of the word *object* differs depending on which stage it is in. The same organ 'Liver' in OOM is referred to as an *object*. The formalization gives a frame description for every *object*. Finally the *objects* are implemented in Smalltalk, and referred to as an instance of a class.

2. Related projects

This section summarizes two projects concerned with the development of the KBS. The MEKAS approach addresses the problem of how to elicit knowledge,

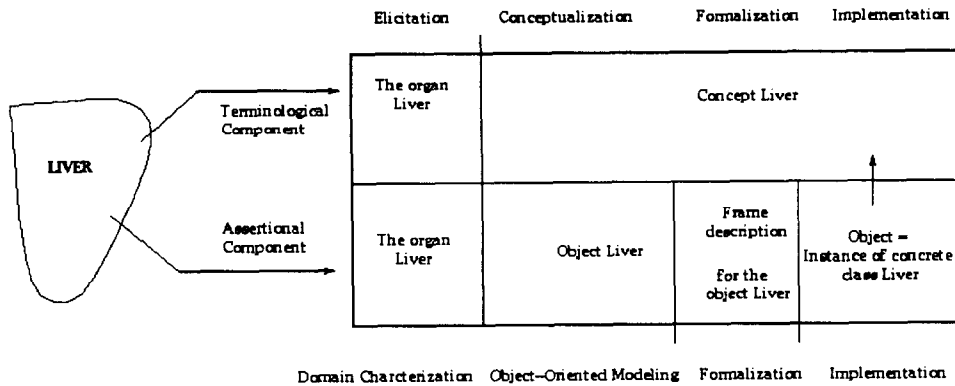


Fig. 1. The organ Liver and its terminology.

but in a broader context of how to 'make sense of the great mass of detail and information obtained, so that it can be organized and represented' [13]. The way in which people organize domains is strongly influenced both consciously and subconsciously by their theoretical knowledge. In this sense MEKAS has made a deliberate decision regarding the importance of theory in the structural and functional organization of domains. For this reason, MEKAS seeks to acquire information about the underlying theory which has influenced the development of the domain from the expert's point of view [14]. The result of a domain characterization of blood glucose regulation was used as the main source for object-oriented modelling of the domain [8].

The GALEN project is developing a terminological knowledge base, the Coding Reference (CORE) model [2]. The CORE model contains 'sensible' medical concepts and their relationships independent of the language in which they are expressed. The concepts in the CORE model are defined using the formalization language GRAIL [15, 17]. GRAIL belongs to the large class of structured knowledge representation systems which also includes semantic nets and frame-based systems.

Concepts are defined by 'particularization', that is by adding a set of defining criteria to a base to distinguish the concept from other concepts with the same base (see Fig. 2). This method of defining concepts is similar to that of KL-ONE [19].

The main function of a GRAIL model is to store knowledge of which compositions are sanctioned (i.e. what is allowed). For example, the statement 'Tissue which hasStructuralComponent HepatocyteCell' in Fig. 3 has to be allowed before one can define the complex concept 'HepatocytePlate'. Instead of giving constraints to restrict compositions, sanctions are explicitly defined. This results in those statements which are legal in creating new composite concepts.

Much of the activity in refining GRAIL models consists of making sanctions

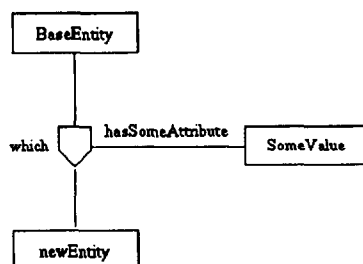


Fig. 2. Particularization of a concept. (BaseConcept which hasSomeAttribute Criteria) name NewConcept.

more or less general. Sanctioning rules are defined by a triple. A triple has the syntax:

Concept *triple* relation Concept *qualifier*.

The following rule states that it is grammatically correct to define that a certain organ produces a certain biomedical material:

Organ *triple* produces BiomedicalMaterial *grammatical*.

A grammatically correct but medically incorrect statement sanctioned by the former rule is *Heart produces Insulin*.

Fig. 4 shows how the KBS evolved from the MEKAS and the GALEN projects. A domain characterization of blood glucose regulation was available from the MEKAS project. This was used as the main source for object-oriented modelling of the domain. The available experimental version of the GALEN CORE model did not include medical concepts needed for conceptualizing blood glucose regulation. These concepts were defined in order to use the CORE model for the terminological components of the KBS.

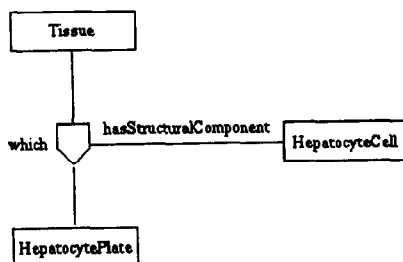


Fig. 3. Definition of a composite concept. (Tissue which hasStructuralComponent HepatocyteCell) name HepatocytePlate.

3. Different kinds of knowledge

In most domains a distinction can be made between knowledge concerning terminology, resulting in a terminological component, and knowledge describing phenomena, resulting in an assertional component.

These different kinds of knowledge require different kinds of KRS [3]. This distinction brings with it the advantage that the development, augmentation and maintenance of both components can be performed without side-effects to the other component [3].

Hybrid systems have been developed which separate terminological knowledge from assertional knowledge. However, in practice, the line between terminological and assertional knowledge is sometimes unclear [6,20].

3.1. Terminological knowledge

Technical vocabulary cannot be avoided within the knowledge elicitation process of a complex domain. A knowledge engineer has to acquire the technical jargon in order to understand domain experts. Consider the sentence:

'Susceptibility phenotypes are necessary but not sufficient to cause insulin-dependent diabetes mellitus'.

The description of terms, like 'insulin-dependent diabetes mellitus', constitutes terminological knowledge [12].

One way to get some meaning of a concept is by creating composite concepts from primitive, or atomic concepts. For instance, the complex concept 'insulin-de-

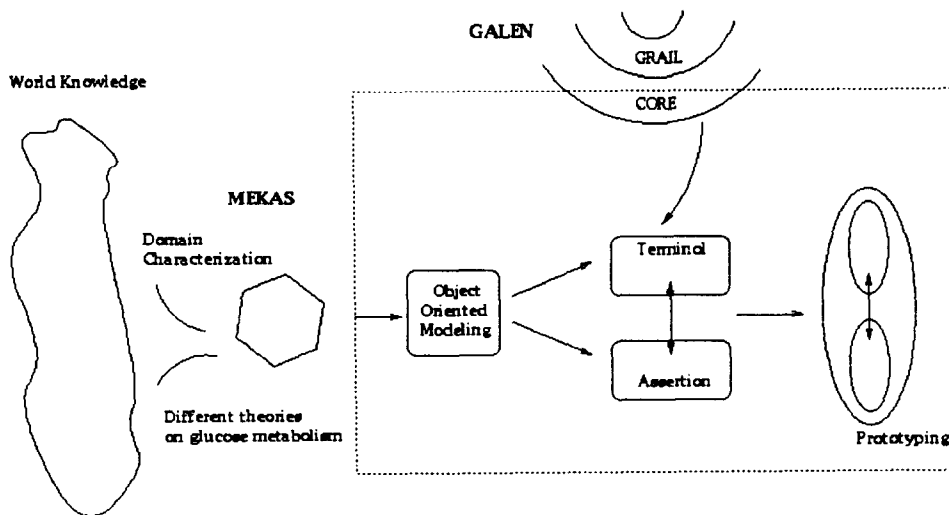


Fig. 4. The different projects. The dashed box indicates the actual work.

pendent diabetes mellitus' can be created from the primitive concepts 'insulin' and 'diabetes mellitus'. At least now it can be said that this concept is a kind of diabetes mellitus. This approach was used to represent concepts in the terminological component.

Rector et al. [16] proposed a terminological KBS containing only 'sensible' medical concepts (see Section 2). This KBS, the CORE model, includes three types of knowledge:

- (1) Conventional knowledge
- (2) Descriptive knowledge
- (3) Formal knowledge.

Conventional knowledge defines atomic and composite concepts. Compositions have to be allowed in the model by descriptive knowledge (see Section 2). In the representation of a composite concept '(*diabetes mellitus* which dependsOn *insulin*) name: *insulin-dependent diabetes mellitus*', the concepts 'insulin' and 'diabetes mellitus' have to exist and the relationship, 'dependsOn', between the concepts has to be sanctioned by the model. Once represented, formal knowledge can deduce that 'insulin-dependent diabetes mellitus' is a kind of 'diabetes mellitus'.

3.2. Assertional knowledge

The relationships between atomic or complex concepts representing theories, or describing phenomena are said to be assertional. For example in the sentence 'the glucose concentration in the blood plasma affects the insulin production in the pancreas' there is an assertional relation between the glucose concentration in the blood plasma and the insulin production in the pancreas'.

From a medical perspective a domain can be divided into three different kinds of knowledge [21]. First, knowledge based on associations between diseases and their signs and symptoms, call *associational knowledge*¹. Second, certain parts of medical practice are clearly based on well-understood quantitative models of human physiology. These models represent some aspects of the body functions. This kind of knowledge is called *empirical quantitative models*. Finally, models which express the function and malfunction of the body based on underlying biological and physiological theories are referred to as causal pathophysiological models. Haimowitz et. al. [6] state that representing associational knowledge only is not sufficient in a complex domain like medicine.

3.3. Experts

The link between both components is defined as follows: every object, representing a medical entity in the assertional component, must refer to its terminolog-

¹ Note that associational knowledge is a subset of the assertional knowledge.

ical concept in the terminological component. This way, integration of different medical information systems becomes feasible. Objects referring to the same concept, represent the same medical entity.

Both components depend on both experts. Building a taxonomy of concepts has to be represented in a KRS. However once a KRS has been developed by knowledge engineers who guide the domain expert in classifying concepts, the actual work is best done by domain experts.

Knowledge for the assertional component is often elicited from domain experts. However, a major problem is representing this knowledge in a form that can be used by a computer. The KRS used depends on the task of the KBS. The know-how of selecting the best KRS can best be performed by the knowledge engineer.

Yet, as will be shown later, the structure of the terminological knowledge assists the knowledge engineer in the knowledge acquisition for the assertional knowledge. The CORE model helps both in understanding the problem domain by way of abstraction, and in classifying knowledge in the assertional component in order to get a good inheritance system.

4. Object-oriented approach

The most significant features of Schaschinger's object-oriented modelling method is that the behaviour of objects and their interaction is much more emphasized than other modelling techniques [18,23]. The OOM method he proposes consists of the following steps:

- (1) *Define the domain* – Understanding the problem domain. The input for the first step are the systems requirements and the knowledge engineer's basic knowledge of the problem domain. The output is a basic understanding of the problem and its domain as well as some intuitive, initial objects and the initial behaviour of the system.
- (2) *Identify objects* – The output is a list of objects.
- (3) *Identify behaviour* – The output is a list of processes performed by objects in order to create the overall behaviour of the system. This step is performed in parallel with step 2.
- (4) *Add attributes to the objects* – Results in adding the data components to objects.
- (5) *Identify structure* – Classify objects. The output is a classification of objects to clarify the problem domain and to manage complexity by setting up levels of abstraction by means of an inheritance hierarchy.
- (6) *Identify collaborations* – Identification of these relationships is based on the dynamic aspects of the domain and the relationships among the objects in order to define the communication between objects.
- (7) *Define constraints, preconditions and postconditions* – Define constraints for the attributes and relationships of objects and define preconditions and postconditions for the objects' behaviour.

In general, these steps are performed in an iterative way. Certain relations help identify the *structure* and *collaboration* between objects.

5. Knowledge representation systems

The tables which result from object-oriented modelling have to be translated into a formal knowledge representation. The preference for one knowledge representation over another depends on the task of the KBS. The task of the KBS is to simulate the glucose regulation. Simulations are naturally modelled with states and events. An object-oriented system creates an event-driven system. This would be an ideal implementation language to build the KBS. Another advantage of object-oriented systems is their ability to develop rapidly a prototype of the system. Especially if no clear specifications of the KBS are available, making a prototype of the system can avoid high development costs.

It turns out that object-oriented systems have many similarities with frame-based systems. However, there are some differences. These differences required a slightly different frame-based system. An object-oriented frame description is introduced. In our representation seven slots are proposed. These slots formalize the different issues of an object-oriented system. The seven slots resulted in the following frame description:

Frame name: Type

Type refers to an Abstract or Concrete class.

Attributes

Class Knowledge (CK)

data valid for all objects of this class.

Instance Knowledge (IK)

data valid for instances of the class, i.e. objects.

Constraints

constraints on values in attribute slot.

Table 1
Identification of concepts and overall behaviour

Identified concepts; overall behaviour	
Liver	Transforming glucose to glycogen and the reverse process
Circulation	Transport medium, also referring to the blood plasma
Pancreas	Dependent on concentration of glucose starts to produce the hormone insulin or glycogen
Gut	Receiving carbohydrates which are transported to the liver

Relationships
 Inheritance
 ISA *superclass*
 hasA *subclass*
 Collaboration
 identification of collaborative working objects.
 hasKnowledgeOf
 LogicalProperties
 defining functions from the attribute values to a boolean value.
 Functions
 defining functions which charge attribute values.
 Events
 defining processes which can be applied by the object.
 Behaviour
 conditional statements: IF Condition THEN Action, where Condition is defined by LogicalProperties and Action is defined by Events.
 End Frame

6. Terminological knowledge

The requirement was to build a KBS which simulates blood glucose regulation. The major knowledge source was the MEKAS domain characterization of blood glucose regulation. This was supplemented by interviews with a domain expert. Table 1 illustrates a few concepts and their descriptions for the overall behaviour.

In order to formalize the identified concepts, use was made of the conceptual models in GRAIL. It must be noted that there is no single 'right way' of constructing such a model. There is usually more than one equally valid and coherent way of representing the same information.

The concepts 'Liver', 'Circulation', 'Pancreas' and 'Gut' were defined as atomic concepts. The classification of these atomic concepts was based on the UMLS network [11] and the 'higher level schemata' of the CORE model. The 'higher level schemata' contain general semantic categories defined by GRAIL. Fig. 5 shows a simplified view of the taxonomy of concepts. All the relationships between concepts are 'specializations' (e.g. Liver -ISA- Organ, Organ -ISA- SpecificStructuralComponent).

The 'higher level schemata' divide the medical domain into different perspectives (e.g. anatomical, functional, physiological, etiological, etc.). Fig. 6 shows a functional perspective. A gland is a functional entity for describing the ability of a component to produce hormones. This can be divided into micro-gland and macro-gland.

The formalization language GRAIL was implemented in Smalltalk. Smalltalk has been used largely because it allows rapid and efficient development of the system. Formalization in GRAIL creates a network of the defined concepts. In

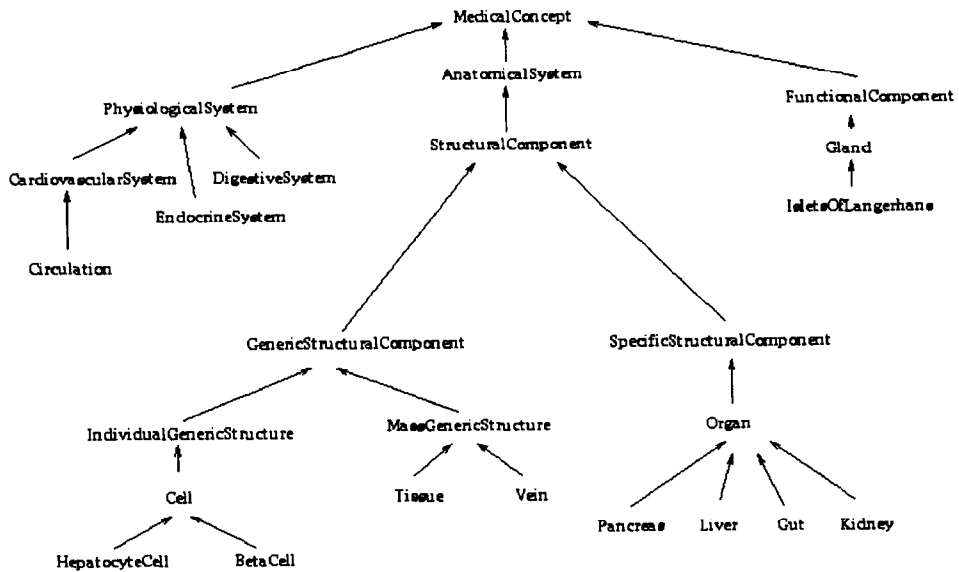


Fig. 5. Simplified view of the taxonomy of concepts.

order to classify automatically composite concepts in the network, some of the object-oriented principles have been carried over from Smalltalk.

The network created can be accessed by Smalltalk methods. For example, sending the message 'formalSubs' to an entity in the network, will return those entities which have been classified by the system as sub-entities. The network created was used as the implemented terminological component. The interface with the assertional component will be described in Section 9.

7. Assertional Knowledge: Object-oriented modelling

This section gives the results of object-oriented modelling of the assertional knowledge of blood glucose regulation. The general steps of the object-oriented modelling method are further developed.

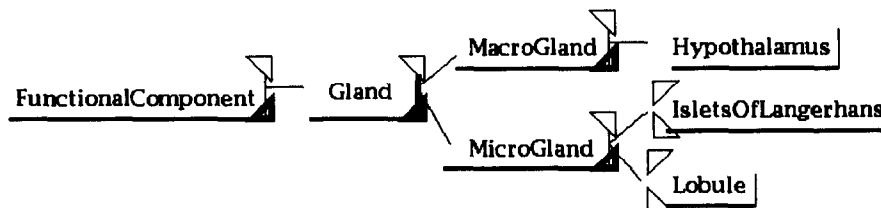


Fig. 6. Functional perspective of a part of the domain.

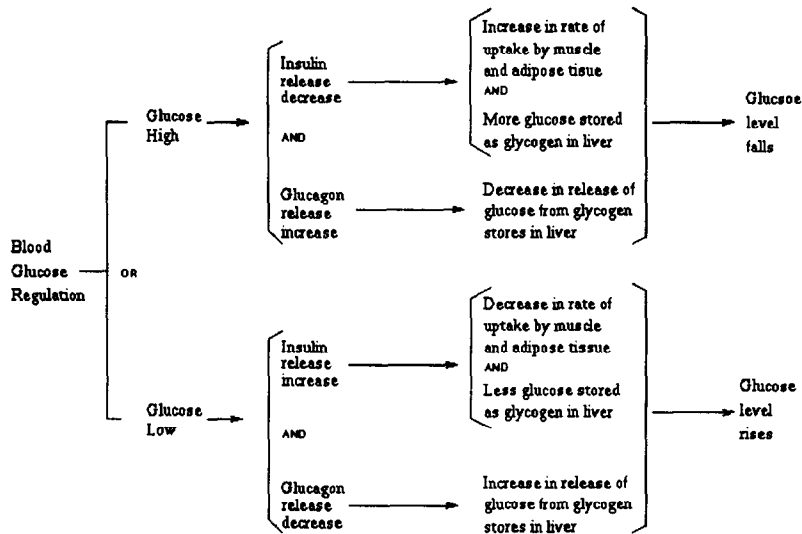


Fig. 7. Glucose level changes as represented in an option network. Logical relationships on how the glucose level changes (taken from [8]).

7.1. Define the domain

In addition to the conceptualization of the terminological knowledge, the result of the domain characterization by MEKAS was used as a major knowledge source. This was supplemented with interviews with a domain expert and knowledge from medical textbooks [5,10]. Fig. 7 is an output of the domain characterization of blood glucose regulation. The figure presents an option network in which the major factors influencing the glucose concentration are represented.

The task of the KBS is to simulate the behaviour of blood glucose regulation which will assist in explaining the process. Therefore, besides looking at the output of the domain characterization, textbooks were studied in order to understand how glucose regulation is explained. Textbooks describe the regulation of glucose by explaining the feedback system which keeps the glucose concentration on the desired level [10]. As long as there is an abnormal concentration in one of the components, the system will activate processes in order to bring it back to a normal value. This continuous control flow where the output of the system is fed back to its input is known as a *feedback system*. Although the glucose concentration in the blood is influenced by many processes, they do not all have effect to the same degree. For simplicity only the amount of glucose, insulin and glucagon will influence the glucose regulation simulated.

7.2. Identify objects

When a complete terminological KBS is available, e.g. the CORE model, the identification of objects is performed for the first time in the development of the

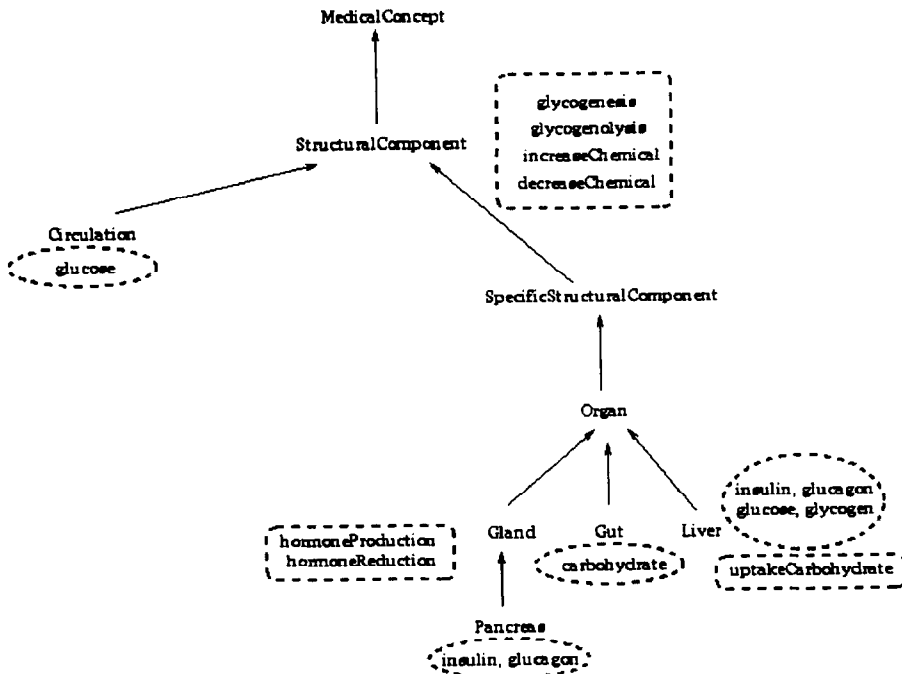


Fig. 8. Inheritance hierarchy of classes with assigned responsibilities. The ovals represent assigned attributes, the boxes represent behaviour.

KBS. The identification of objects is in this case concerned with validation of its existence in the terminological KBS.

The CORE model will also be an important source for identifying objects. Once some concepts are found in the CORE model, the relations with other concepts can give valuable insight into the domain, and help identify other objects used to describe phenomena.

7.3. Identify structure

The aim of an identify structure step is to manage complexity by setting up levels of abstraction by means of an inheritance hierarchy. The inheritance hierarchy of objects is based on the taxonomy of concepts in the terminological component. This can be justified by the fact that medical terms classified under the same parent have certain properties in common, especially a functional classification of organs which can identify common behavioural properties.

The 'higher level schemata' of the CORE model supplies many abstract classes. Using these abstract classes makes it possible to assign common responsibility high up in the inheritance hierarchy. Fig. 8 shows the results of this step. Other steps of the object-oriented modeling are not detailed here as they did not explicitly refer to the CORE terminological component.

8. Assertional knowledge: Formalization

The processes identified earlier are formalized with qualitative calculus. Let $increase[x] @$ be the next qualitative value of $[x] @$ in the quantity space. Let $decrease[x] @$ be the preceding qualitative value of $[x] @$ in the quantity space. The processes identified can be formalized as shown in Table 2:

Conditions for invoking these events are represented with conditional statements. Together with the collaboration with other objects, the behaviour of the system is defined as follows:

```

Gut
    sendChemicalTo (Liver, Carbohydrate)
Liver
    IF [[Insulin] - normal] = +
        THEN activate glycogenesis process
    IF [[Glucagon] - normal] = +
        THEN activate glycogenolyse process
    IF [[Carbohydrate] - normal] = +
        THEN activate uptakeCarbohydrate process
    sendChemicalTo (Circulation, Glucose)
Circulation
    sendChemicalTo (Kidney, Glucose)
    sendChemicalTo (Pancreas, Glucose)
    sendChemicalTo (Liver, Glucose)
Pancreas
    IF [[Glucose] - [Insulin]] = +
        THEN hormoneProduction(Insulin)
    ELSE IF [[Glucose] - [Insulin]] = -
        THEN hormoneReduction(Insulin)
    IF [rev([Glucose]) - [Glycogen]] = +
        THEN hormoneProduction(Glycogen)
    ELSE IF [rev([Glucose]) - [Glycogen]] = -
    
```

Table 2

Process models	
<i>Process(variable)</i>	<i>Increase or decrease in variable</i>
Glycogenesis(Glucose, Glycogen)	$\partial Glucose = -$, $\partial Glycogen = +$
Glycogenolysis(Glucose, Glycogen)	$\partial Glucose = +$, $\partial Glycogen = -$
Uptake(Glucose, Carbohydrate)	$\partial Glucose = +$, $\partial Carbohydrate = -$
HormoneProduction(hormone)	$\partial hormone = +$
HormoneReduction(hormone)	$\partial hormone = -$
IncreaseChemical(chemical)	$\partial chemical = +$
DecreaseChemical(chemical)	$\partial chemical = -$

THEN hormoneReduction(Glycogen)
 sendChemicalTo (Liver, Insulin)
 sendChemicalTo (Liver, Glycogen)

For every object identified earlier a frame description is given. The following frame gives an example of a part of the frame description Liver. Slots marked with a * are inherited from its superframes. These slot values are not defined at this frame but higher up in the inheritance hierarchy.

Liver: Concrete

Attributes

CK:* Q =

(Defines the quantity space)

CK: ChemicalSet(Insulin, Glucagon, Glucose, Glycogen)

(Defines the chemicals)

IK: ChemicalSetValues(Insulin → Qval, Glucagon → Qval,
 Glucose → Qval, Glycogen → Qval)

(Associate the chemicals with a qualitative value)

Constraints*

ChemicalSetValueItem = Chem → Qval, with:

Chem ∈ ChemicalSet and Qval ∈ Q

Instance creation:

\forall Chem ∈ ChemicalSet, ChemicalSetValueItem = Chem → normal

Relationships

Inheritance

-ISA Organ (superclass)

-hasA <empty> (no subclasses)

Collaborations

-hasKnowledgeOf: Circulation(Glucose)

LogicalProperties*

Qval₁, Qval₂ ∈ Q

[Qval₁ - Qval₂] = + iff Qval₁ > Qval₂

(Defines the condition part for the conditional statements)

Functions*

decreaseChemical(Qval) = formerValue(Qval)

Qval₁, Qval₂ ∈ Q

formerValue(Qval₁) = Qval₂ such that:

Qval₂ < Qval₁ and

A Qval ∈ Q, If Qval₂ > 1Qval < Qval₁ then Qval =

Qval₂

Events

Process Glycogenolysis

increaseChemical(Glycogen)

decreaseChemical(Glucose)

```

End
  (Defines the action part for the conditional statements)
Behaviour
  IF [[Insulin] - normal] = +
    Then process Glycogenesis
End Liver

```

9. Assertional knowledge: Implementation in Smalltalk

The assertional component of blood glucose regulation was implemented in the object-oriented programming language Smalltalk. Implementation is done by structuring the application by a Model-View-Controller (MVC) architecture [1]. Models² handle data storage and processing. Every model can be associated with views and controllers. Views and controllers create the user interface. The view displays the output on screen and the controller enables the user to interact with, or control, the simulation.

Data and processing data is stored in the model component. Therefore, almost all the slots of the frame are implemented in this component. The activation of the behaviour slot is carried out by the user. This slot is implemented in the controller part. Fig. 9 displays a part of the MVC architecture for the frame 'Liver'.

The state of an object, the attribute slot, is implemented in class and instance variables. The Class Knowledge (CK) is implemented as class variables. The Instance Knowledge (IK) is implemented as instance variables. The processes in the events slot are defined as methods. Both the variables and methods are defined in the model component. The constraints slot, logicalProperties slot and functions slot are inherited from the model component of superclasses.

For every instance variable a bargraph is shown on the screen (arrow 1). The bargraph view is defined in the view component. The conditional statement in the behaviour slot are implemented in the controller component. The controller component gives the user the possibility to activate an event in the model (arrow 2). Dependent on the state of the model, the action-part of the conditional statement is executed (arrow 3). After the state changes in its own model (arrow 4), the Circulation model is updated (arrow 5).

10. Extension of the knowledge-based system

It is often argued that an important property of object-oriented systems is their convenience for maintaining and extending software. In order to study this feature on the developed KBS, it has been extended in different ways. After the knowledge acquisition process for Gut, Liver, Circulation, and Pancreas, the KBS is

² In this section, the word 'model' refers to Smalltalk model: 'data storage and processing'.

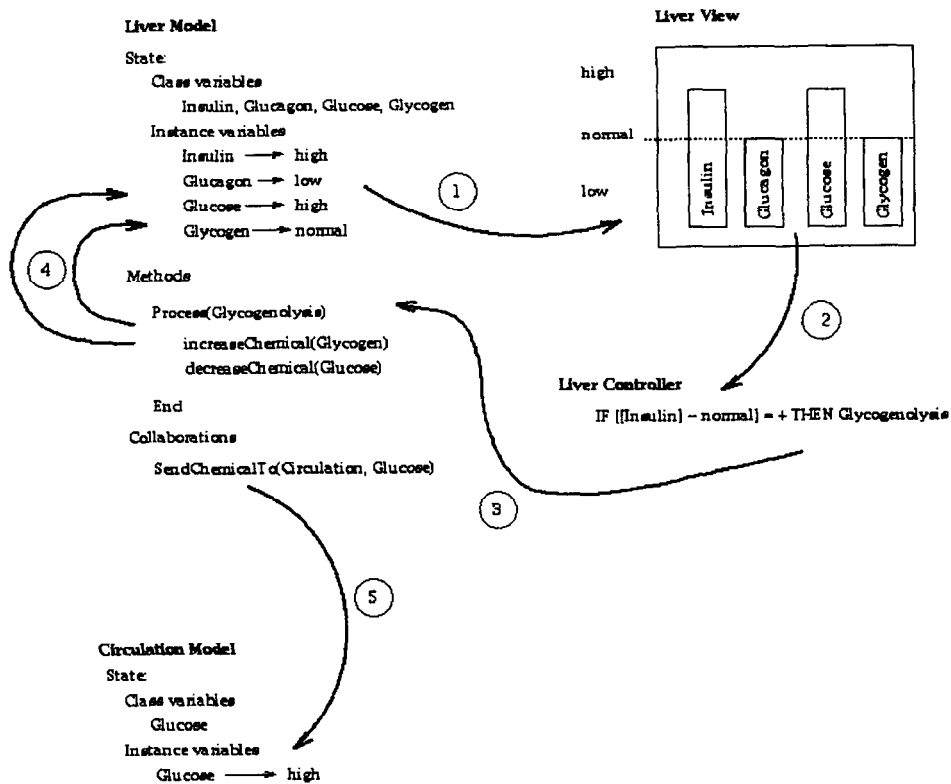


Fig. 9. MVC architecture. Part of the MVC architecture for Liver.

updated. All the steps of the knowledge acquisition process are carried out again in order to represent the additional knowledge.

The KBS was extended with an object Kidney. This object simulates the excretion of glucose beyond a threshold value in the blood plasma. All the steps of the knowledge acquisition process for both the terminological and assertional component were carried out. The conceptualization for the terminological component is quite straightforward.

- * Define the domain: Extension of the KBS with a simulation of the Kidney which excretes the surplus of glucose.
- * Identify concept: 'Kidney'.
- * Classify concept: The concept 'Kidney' was classified as a specialization of 'Organ'.

The concept 'Kidney' was formalized in GRAIL, henceforth, accessible in Smalltalk. The OOM steps applied to the assertional knowledge also did not cause any problems. The frame for 'Circulation' has to be updated. The collaboration slot 'hasKnowledgeOf' is extended with the data value 'Kidney(Glucose)'.

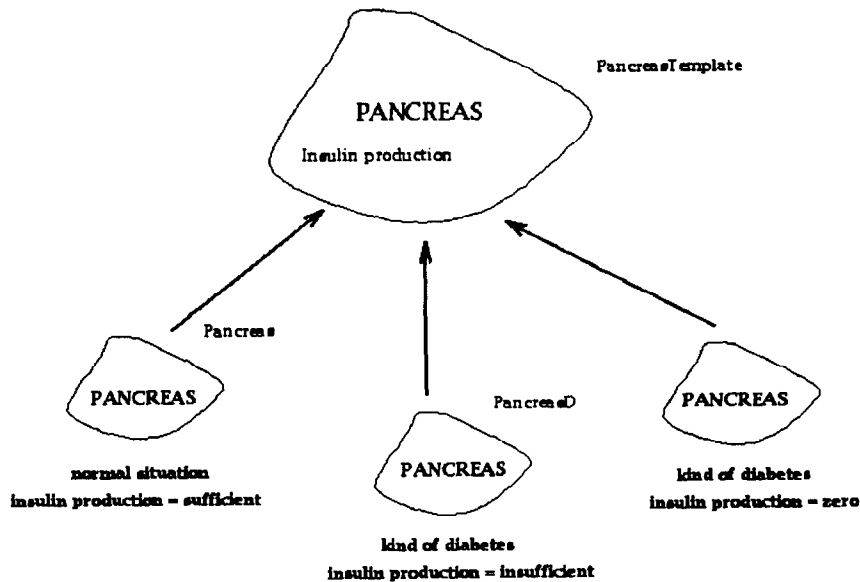


Fig. 10. Inheritance hierarchy of pancreas.

10.1. Representing abnormal behaviour

A simulation of an abnormal process can be represented. This is illustrated by defining an abnormal pancreas which does not have the ability to produce enough insulin (e.g. in the case of classical juvenile onset diabetes³ failure of insulin production [22]). The theory behind this is that the collection of beta cells, referred to as the islets of Langerhans, is not big enough to produce sufficient insulin.

A pancreas with low insulin production is still a pancreas. All the other functions of a normal pancreas can be inherited. In order to avoid overriding default values, an abstract class 'PancreasTemplate' is defined. PancreasTemplate defines the common behaviour of normal and abnormal pancreases (see Fig. 10). PancreasTemplate still defines the process of producing insulin, however this does not specify the quantity of the production. This is specified lower down the hierarchy. For both components the effect of representing abnormal behaviour is described.

It would be out of the scope of CORE to represent a concept for every abnormality. In order to keep integration with other medical systems possible, the PancreasTemplate has to refer to the concepts 'Pancreas' in the CORE model. System knowledge should deduce that specializations of PancreasTemplate refer to the concept 'Pancreas'.

³ There are other kinds of diabetes mellitus which are caused by other abnormalities.

For the assertional component the abnormal pancreas is represented by the object PancreasD. The object-oriented model is formalized in frame descriptions. Besides the frame description for PancreasTemplate and PancreasD, the frame Pancreas is updated. Finally, the frames PancreasTemplate and PancreasD are implemented as an MVC architecture in Smalltalk and the model component of Pancreas was updated.

10.2. Linking underlying theories

An explanation-task KBS often requires deep knowledge to explain shallow reasoning. For example, how is glucose used by cells as fuel. This can be explained by the Krebs cycle. This subsection describes how underlying theories can be linked within the system.

Separating the conditions from the actions in a conditional statement during the formalization of the assertional knowledge, makes linking underlying theories easier. The conditional statements in the behaviour slot represent reasoning on a shallow level. The activation of processes defined in the frame are on a shallow level. It could have been easily defined by the conditional statements directly. For example the behaviour of a Liver could have been formalized as:

Liver: Concrete

```

...
Behaviour
  If [[Insulin] - normal] = +
    Then increaseChemical(Glycogen)
      decreaseChemical(Glucose)
...
...

```

However, using the proposed approach of separating the actions from the conditions creates the advantage that heuristic processes defined in the event slot can be redefined by processes representing deeper knowledge. In other words, shallow reasoning is presented in the behaviour slot and deeper knowledge can be presented in the event slot.

In summary this section shows that the different extensions of the KBS were easily realized. This was achieved by two features of object-oriented systems. First, the inheritance hierarchy creates a template for new defining classes of objects. Only a few slots have to be defined in adding a new component. Polymorphism made it possible that an abnormal component can respond to a message in a different way than the normal component. Finally, separating the conditions from the actions in behaviour slots and event slots creates several advantages:

- * Events can be defined higher up in the hierarchical structure, hence, valid for all its subclasses.
- * The accuracy of processes is flexible. If required, a process in the event slot can be represented by deeper knowledge which explains underlying theories.

11. Simulation

This section describes the simulation of blood glucose regulation performed by the KBS. The first menu shown in Fig. 11 contains two commands, the first command 'Terminology link' consists of a verification that every object used in the assertional component has an associated concept. All the objects which do not have an associated concept are listed in a Window (i.e. a Warning view).

The second command; 'start simulation' in Fig. 11 gives two options: Normal or Diabetes. Selecting 'Normal' starts up a simulation of the normal glucose regulation. The second choice, 'Diabetes', simulates the glucose regulation in case the Pancreas is not able to produce sufficient insulin (a kind of Diabetes Mellitus).

After starting the simulations, the window of Fig. 12 appears on screen (i.e. a BgrMain View). This window consists of two parts, the upper part shows a picture of the main organs involved in the blood glucose regulation. The picture refers to the statement that the process can be seen as a system [8]. The lower part presents the history of the activated processes. The history is presented in a backward order, that is, the last activation is presented on top of the list.

Besides the main window, an organ selector window appears on screen (see Fig. 13). This window gives the user the choice of selecting which organs are shown on screen. If the user is only interested in the excretion of the glucose by the Kidney, showing the simulation of the Gut is not functional.

For every selected organ a window is displayed on screen. These windows are automatically placed like the organs in the picture in the main view. This helps the user to associate the window with the organ it is representing. An organ window consists of the following fields (see Fig. 14):

- * *view*. A button which selects the window on a picture or process view of organ. The picture-view makes it possible to zoom in or out on an organ (see Fig. 14). The process-view presents a process which is associated with a certain picture. On the highest level, the overall behaviour of the organ is presented as a barchart view (see Fig. 15).
- * *value*. A button which allows the user to increase or decrease a chemical value at any time during the simulation. This feature makes it possible to simulate the

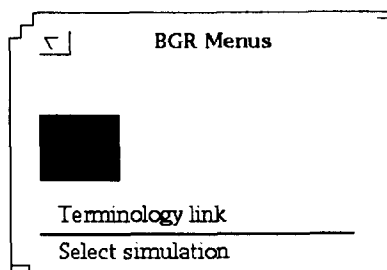


Fig. 11. BGR-simulation. Main menu of the BGR-simulation.

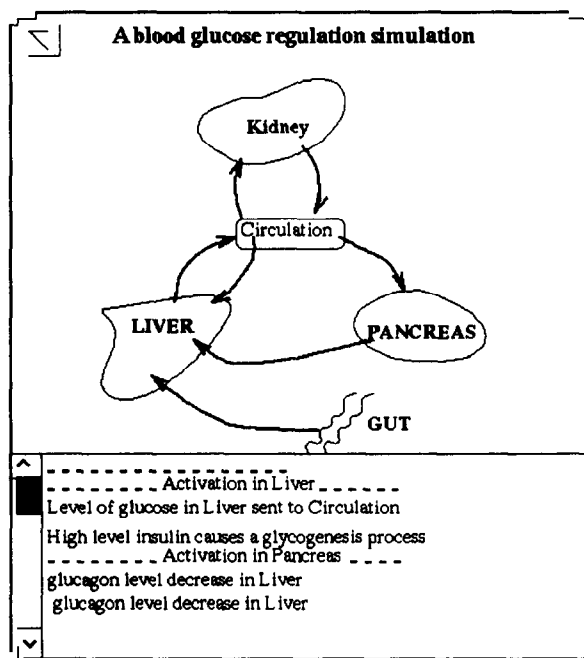


Fig. 12. BgrMain View. The upper part shows a picture, the lowest part gives the history of the simulation.

influence of an injection of insulin in the blood on the glucose regulation by diabetes.

* *reset*. A button which sets the qualitative value of the chemicals to the value normal.

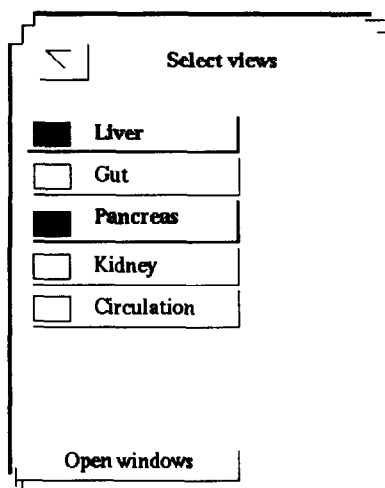


Fig. 13. Organ selector menu.

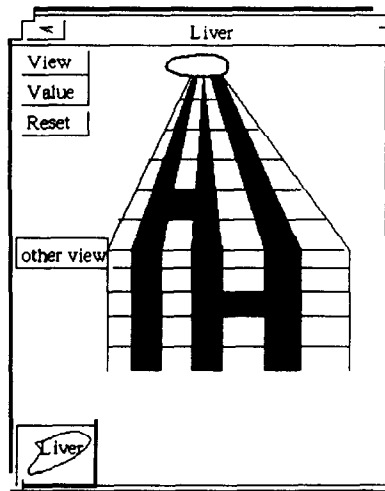


Fig. 14. Organ view: picture view. Organ view, on a picture view.

* *icon*. A small picture referring to the drawing in the main window. This turned out to be necessary when underlying processes are explained. At that time it is easy to forget in which organ the process takes place.

The link with CORE only has to be carried out in testing the final implementation of the system. Not all objects used have to be defined by CORE. Only objects representing medical entities need to be associated with a concept in CORE. Objects can be associated with a concept through its superclass. For example, PancreasD can refer to concept 'Pancreas' by association of PancreasTemplate with concept 'Pancreas' (see Fig. 16).

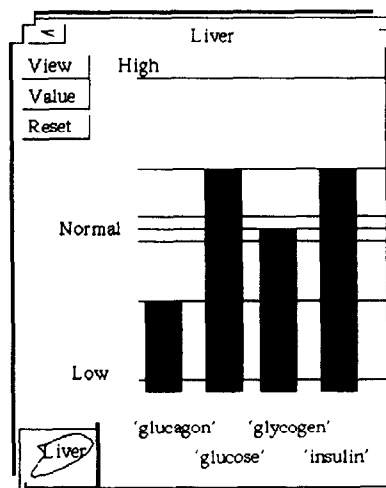


Fig. 15. Organ view: barchart view. Organ view, on a barchart view.

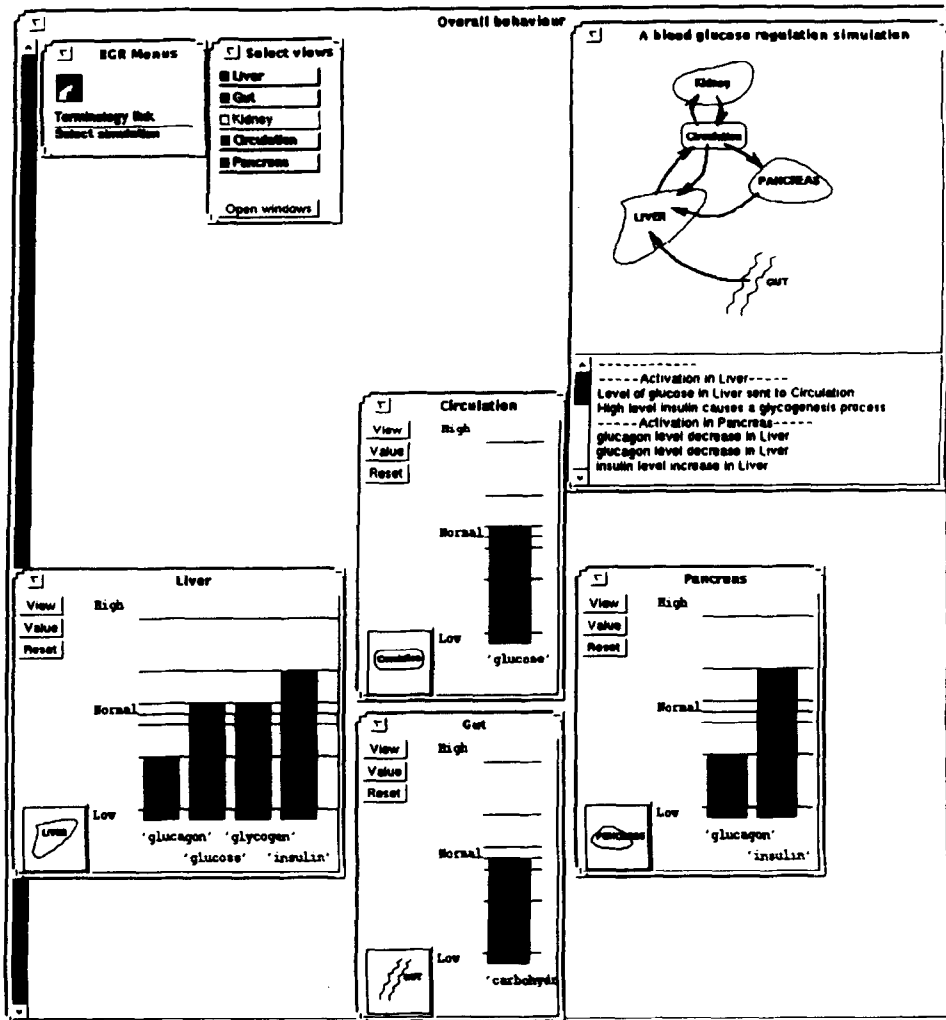


Fig. 16. Full simulation on the glucose regulation.

12. Conclusions

An object-oriented approach to knowledge acquisition for assertional knowledge was tested. An object-oriented modelling method was applied to the problem domain. A domain characterization of blood glucose regulation was used as the main source of knowledge for the development of the KBS. The domain characterization was the first abstraction performed in representing knowledge of the real world.

An OOM method accentuates the hierarchical structure of objects. The CORE model played a major role in the construction of such a structure. This helps the knowledge engineer to structure the mass of detailed knowledge in a domain. In

particular, the OOM method used emphasizes the behaviour of objects and their interactions more than other modelling techniques [18]. Special attention has been paid to the collaboration between objects. As a consequence, this modelling technique provides valuable results for storing knowledge in objects.

The frame description used for the formalization of the KBS consists of seven different slots (i.e. attribute, constraints, relationships, logicalProperties, functions, events and behaviour). The different features of an object-oriented system were formalized in these slots. The 'state' of an object was defined by an attribute slot and constraint slot, the possible 'events' were formalized by functions slot, events slot, logicalProperties slot and behaviour slot. Collaboration and inheritance relations were defined in the relation slot.

The state of an object consists of qualitative values. Using qualitative values corresponds to the reasoning method often used by medical experts. Qualitative calculus was used in formalizing the events and the constraints on the attribute values. Dividing events over so many slots creates the opportunity to assign slots higher up in the hierarchy. In order to study maintenance and enhancement of knowledge in an object-oriented system, the KBS was updated. Inheritance and polymorphism in object-oriented systems made this easy to realize.

A major problem of representing assertional knowledge is to represent the domain knowledge in such a way that it can be used effectively by a computer. Selection of the best KBS is best performed by a knowledge engineer. On the other hand, separation of terminological and assertional knowledge does not impose that the assertional component should be developed independently of the terminological knowledge. On the contrary, the knowledge in a terminological KBS assists the knowledge engineer in building the assertional component. A core of medical concepts assists by providing background knowledge of anatomy and pathophysiology which is outside the field of expertise of a knowledge engineer. The hierarchical structure of objects was adopted from the taxonomy of concepts. At a conceptual level, such a structure helps to manage the complexity of a domain. At an implementation level, the hierarchical structure of medical concepts has been a knowledge source for the construction of an inheritance hierarchy of classes. In this inheritance hierarchy, common behaviour was classified as high up in the hierarchy as possible.

Using a thesaurus, like the UMLS metathesaurus, can decrease the development time for a KBS, however this is not as powerful as the CORE model of the Galen project. The CORE model is to be preferred over a thesaurus, because it contains more detailed semantic relations. For example the 'broader-than' relationship in a thesaurus includes both ISA and part-of relationships in the CORE model.

Acknowledgments

The support of European Community GALEN project is gratefully acknowledged. The authors thank Malcolm Talyor, Paul van der Vet, Ad Feelders and Nicolaas Mars for their contributions.

References

- [1] *Objectworks Smalltalk User's Guide* (Sep. 1992.)
- [2] GALEN deliverable 1, Technical report Medical Informatics Group, Department of Computer Science, University of Manchester, Manchester, UK, Dec. 1992.
- [3] R.J. Brachman and H.J. Levesque, Competence in knowledge representation, *Proc. Nat. Conf. on Artificial Intelligence* (1982) 189–192.
- [4] M. Ensing, An Object-Oriented Approach to Knowledge Representation in a Biomedical Domain, Department of Computer Science, University of Liverpool, UK, 1993.
- [5] W.F. Ganong, Review of medical physiology, in: Appleton and Lange, eds., *Human Physiology* (Prentice-Hall, 1987).
- [6] I.J. Haimowitz, R.S. Patil and P. Szolovits, Representing medical knowledge in a terminological language is difficult, *Proc. Twelfth Annual Sym. on Computer Applications in Medical Care* (1988) 101–105.
- [7] F. Hayes-Roth, D.A. Waterman and D.B. Lenat, *Building Expert Systems* (Addison Wesley, 1983).
- [8] D. Jones, Application of SAAGS methodology (Specification, Anticipation, Acquisition, Generation), Technical Report 30, *MEKAS*, Department of Computer Science, Liverpool University, UK, 1992.
- [9] W. Kreutzer and B. McKenzie, *Programming for Artificial Intelligence; Methods, Tools and Applications* (Addison-Wesley, Sydney, 1991).
- [10] J.F. Lamb, C.G. Ingram, I.A. Johnston and R.M. Pitman, *Essentials of Physiology* (Blackwell Scientific, London, 1986).
- [11] National Library of Medicine, UMLS knowledge sources, *Technical Report Third Experimental Edition*, US Department of Health and Human Services, USA, August 1992.
- [12] G. De Moor, Directory of the European standardization requirements for health care informatics and programme for the development of standards, Version 1.5 CEN TC 251, European Standardization Committee (CEN), University Hospital of Gent, Institute of Medical Informatics, Gent, Belgium, Sep. 1992.
- [13] R.C. Paton, H.S. Nwana, T.J.M. Bench-Capon and S. Hughes, Foundations of a structured approach to characterizing domain knowledge, Department of Computer Science, University of Liverpool, UK, 1992.
- [14] R.C. Paton, H.S. Nwana, T.J.M. Bench-Capon and M.J.R. Shave, Informal models and interpretation, Department of Computer Science, University of Liverpool, Liverpool, UK, 1992.
- [15] I. Piumarta, Galen Concept Representation Language (GCRL); Documentation for GCRL Engine and Compiler, first revision GCRL 1.0 edition, Medical Informatics Group, Department of Computer Science, University of Manchester, Manchester, UK, Jan. 1993.
- [16] A.L. Rector, W.A. Nowlan and S. Kay, Conceptual knowledge: The core of medical information systems, *Proc. Seventh World Congress on Medical Informatics* (North-Holland, 1992) 1420–1426.
- [17] A.L. Rector, Generalized architecture for languages encyclopedias and nomenclatures in medicine, Technical Annex A2012, Department of Computer Science, University of Manchester, Manchester, UK, Dec. 1992.
- [18] H. Schaschinger, Object-oriented analysis and modelling with the ESA station, *Microprocessing and Microprogramming* 35 (Sep. 1992) 221–228.
- [19] J.G. Schmolze and R.J. Brachman, An overview of the KL-ONE knowledge representation system, *Cognitive Sci.* 9 (2) (1985) 171–216.
- [20] P.H. Speel, P.E. van der Vet, W. ter Stal and N.J.I. Mars, Formalization of an ontology of ceramic science in classic, *Proc. Seventh Int. Symp. on Methodologies for Intelligent Systems (ISMIS)*, Oak Ridge National Laboratory, Trondheim, Norway, (June 15–18 1993) 110–124.
- [21] P. Szolovits, Types of knowledge as bases for reasoning in medical AI programs, *Artificial Intelligence in Med.* (1985) 209–225.
- [22] W.M. Tunbridge and P.D. Home, Diabetes and endocrinology in clinical practice, in: *Clinical Practice Series* (E. Arnold, London, Melbourne, Auckland, 1991)
- [23] A.C. Verschuren, An object-oriented modelling technique for analysis and design of complex (real-time) systems, University of Eindhoven, Eindhoven, 1992.
- [24] P. Wegner, Dimensions of object-based language design, *OOPSLA '87 Proc.* (Oct. 1987) 168–182.