Institute for Language, Logic and Information

PHYSIOLOGICAL MODELLING USING RL

Fred de Geus
Ernest Rotterdam
Sieger van Denneheuvel
Peter van Emde Boas

ITLI Prepublication Series
for Computation and Complexity Theory CT-90-08

University of Amsterdam

The ITLI Prepublication Series

1986
86-01 The Institute of Language, Logic and Information
86-02 Peter van Emde Boas — A Semantical Model for Integration and Modularization of Rules
86-03 Johan van Benthem — Categorial Grammar and Lambda Calculus
86-04 Reinhard Muskens — A Relational Formulation of the Theory of Types
86-05 Kenneth A. Bowen, Dick de Jongh — Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06 Johan van Benthem — Logical Syntax

1987
87-01 Jeroen Groenendijk, Martin Stokhof — Type shifting Rules and the Semantics of Interrogatives
87-02 Renate Bartsch — Frame Representations and Discourse Representations
87-03 Jan Willem Klop, Roel de Vrijer — Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04 Johan van Benthem — Polyadic quantifiers
87-05 Víctor Sánchez Valencia — Traditional Logicians and de Morgan's Example
87-06 Eleonore Oversteegen — Temporal Adverbials in the Two Track Theory of Time
87-07 Johan van Benthem — Categorial Grammar and Type Theory
87-08 Renate Bartsch — The Construction of Properties under Perspectives
87-09 Herman Hendriks — Type Change in Semantics: The Scope of Quantification and Coordination

1988
Logic, Semantics and Philosophy of Language:
LP-88-01 Michiel van Lambalgen — Algorithmic Information Theory
LP-88-02 Yde Venema — Expressiveness and Completeness of an Interval Tense Logic
LP-88-03 — Year Report 1987
LP-88-04 Reinhard Muskens — Going partial in Montague Grammar
LP-88-05 Johan van Benthem — Logical Constants across Varying Types
LP-88-06 Johan van Benthem — Semantic Parallels in Natural Language and Computation
LP-88-07 Renate Bartsch — Tenses, Aspects, and their Scopes in Discourse
LP-88-08 Jeroen Groenendijk, Martin Stokhof — Context and Information in Dynamic Semantics
LP-88-09 Theo M.V. Janssen — A mathematical model for the CAT framework of Eurotra
LP-88-10 Anneke Kleppe — A Blissymbolics Translation Program

Mathematical Logic and Foundations:
ML-88-01 Jaap van Oosten — Lifschitz' Realizabiility
ML-88-02 M.D.G. Swaen — The Arithmetical Fragment of Martin Löf's Type Theories with weak Σ-elimination
ML-88-03 Dick de Jongh, Frank Veltman — Provability Logics for Relative Interpretability
ML-88-04 A.S. Troelstra — On the Early History of Intuitionistic Logic
ML-88-05 A.S. Troelstra — Remarks on Intuitionism and the Philosophy of Mathematics

Computation and Complexity Theory:
CT-88-01 Ming Li, Paul M.B.Vitanyi — Two Decades of Applied Kolmogorov Complexity
CT-88-02 Michiel H.M. Smid — General Lower Bounds for the Partitioning of Range Trees
CT-88-03 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas — Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04 Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette — Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05 Peter van Emde Boas — Machine Models and Simulations (revised version)
CT-88-06 Michiel H.M. Smid — A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07 Johan van Benthem — Time, Logic and Computation
CT-88-08 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas — Multiple Representations of Dynamic Data Structures
CT-88-09 Theo M.V. Janssen — Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10 Edith Spaan, Leen Torenvliet, Peter van Emde Boas — Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11 Sieger van Denneheuvel, Peter van Emde Boas — Towards implementing RL

Other prepublications:
X-88-01 Marc Jumelet — On Solovay's Completeness Theorem

1989
Logic, Semantics and Philosophy of Language:
LP-89-01 Johan van Benthem — The Fine-Structure of Categorial Semantics
LP-89-02 Jeroen Groenendijk, Martin Stokhof — Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
LP-89-03 Yde Venema — Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals
LP-89-04 Johan van Benthem — Language in Action
LP-89-05 Johan van Benthem — Modal Logic as a Theory of Information
LP-89-06 Andreja Prijatelj — Intensional Lambek Calculi: Theory and Application
LP-89-07 Heinrich Wansing — The Adequacy Problem for Sequential Propositional Logic
LP-89-08 Víctor Sánchez Valencia — Peirce's Propositional Logic: From Algebra to Graphs
LP-89-09 Zhisheng Huang — Dependency of Belief in Distributed Systems

Mathematical Logic and Foundations:
ML-89-01 Dick de Jongh, Albert Visser — Explicit Fixed Points for Interpretability Logic
ML-89-02 Roel de Vrijer — Extending the Lambda Calculus with Surjective Pairing is conservative
ML-89-03 Dick de Jongh, Franco Montagna — Rosser Orderings and Free Variables
ML-89-04 Dick de Jongh, Marc Jumelet, Franco Montagna — On the Proof of Solovay's Theorem
ML-89-05 Rineke Verbrugge — Σ-completeness and Bounded Arithmetic
ML-89-06 Michiel van Lambalgen — The Axiomatization of Randomness
ML-89-07 Dirk Roorda — Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone
ML-89-08 Dirk Roorda — Investigations into Classical Linear Logic
ML-89-09 Alessandra Carbone — Provable Fixed points in $I\Delta_0+\Omega_1$

Computation and Complexity Theory:
CT-89-01 Michiel H.M. Smid — Dynamic Deferred Data Structures
CT-89-02 Peter van Emde Boas — Machine Models and Simulations
CT-89-03 Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas — On Space Efficient Simulations
CT-89-04 Harry Buhrman, Leen Torenvliet — A Comparison of Reductions on Nondeterministic Space
CT-89-05 Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree — A Parallel Functional Implementation of Range Queries
CT-89-06 H.W. Lenstra, Jr. — Finding Isomorphisms between Finite Fields
CT-89-07 Ming Li, Paul M.B. Vitanyi — A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Prel. Version)
CT-89-08 Harry Buhrman, Steven Homer, Leen Torenvliet — Honest Reductions, Completeness and Nondeterminstic Complexity Classes
CT-89-09 Harry Buhrman, Edith Spaan, Leen Torenvliet — On Adaptive Resource Bounded Computations
CT-89-10 Sieger van Denneheuvel — The Rule Language RL/1
CT-89-11 Zhisheng Huang, Sieger van Denneheuvel, Peter van Emde Boas — Towards Functional Classification of Recursive Query Processing

Other Prepublications:
X-89-01 Marianne Kalsbeek — An Orey Sentence for Predicative Arithmetic
X-89-02 G. Wagemakers — New Foundations: a Survey of Quine's Set Theory
X-89-03 A.S. Troelstra — Index of the Heyting Nachlass
X-89-04 Jeroen Groenendijk, Martin Stokhof — Dynamic Montague Grammar, a first sketch
X-89-05 Maarten de Rijke — The Modal Theory of Inequality
X-89-06 Peter van Emde Boas — Een Relationele Semantiek voor Conceptueel Modelleren: Het RL-project

1990 SEE INSIDE BACK COVER

Instituut voor Taal, Logica en Informatie
Institute for Language, Logic and Information

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Plantage Muidergracht 24
1018TV Amsterdam

Faculteit der Wijsbegeerte
(Department of Philosophy)
Nieuwe Doelenstraat 15
1012CP Amsterdam

PHYSIOLOGICAL MODELLING USING RL

Fred de Geus
Ernest Rotterdam
Vakgroep Medische Informatica, Rijksuniversiteit Groningen
Oostersingel 59, Gebouw 59, Groningen
Sieger van Denneheuvel
Peter van Emde Boas
Department of Mathematics and Computer Science
University of Amsterdam

ITLI Prepublication Series
for Computation and Complexity Theory
ISSN 0924-8374

Received December 1990

1 Introduction

This article discusses the use of constraints to build quantitative physiological models and the application of these models to assist anaesthetists in decision making. The constraint formalism is a declarative form of knowledge representation. Each constraint embodies a model equation. A set of constraints constitutes a model. Constraint models can be extended and modified easily and the representation of model knowledge is separated from its use. The model knowledge is used by means of a constraint satisfaction mechanism. This allows the application of the same knowledge for different purposes.
In this article a simple physiological model is presented. It is represented in the relational language RL[4,5,6,7,8] and used for two purposes: interpretation and prediction.

2 Model description

This section describes a simple physiological model of the human blood circulation and gas exchange. The model consists of 11 equations; 6 describe the systemic and pulmonary circulation and the remaining 5 the gas exchange. It is a simple model and therefore a poor reflection of reality, but it serves well to demonstrate how a model represented in constraints can be used for decision support. Figure 1 shows a graph representation of the model.

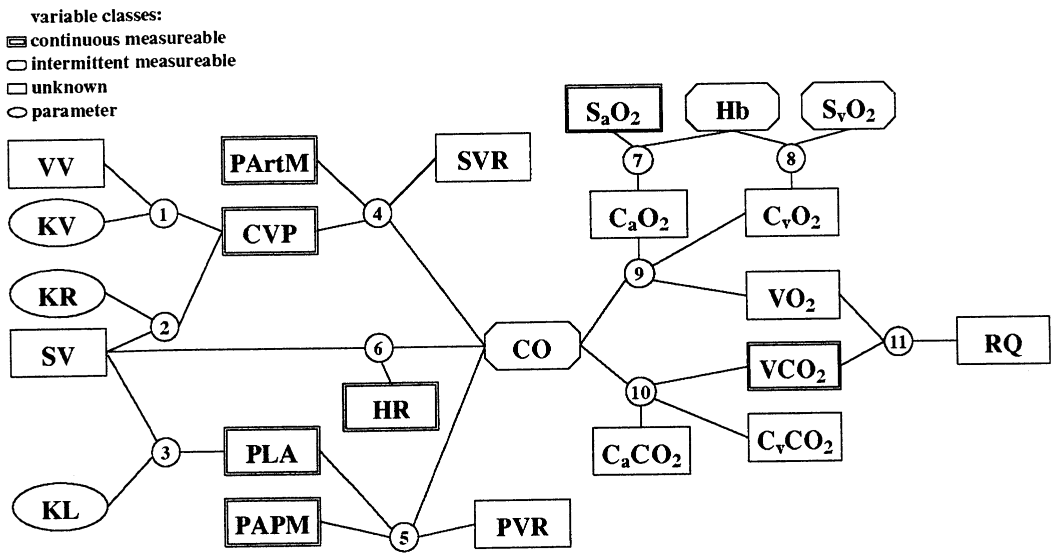

Figure 1: Graph representation of the model

Variable names used in the model are abbreviated according to convention. A summary of the properties of the variables used is shown in table 1. The numbers of the model equations listed below correspond to the numbers in the graph.

Abbre-viation	normal-value	units[a]	Name
CO	5500	ml/min	Cardiac Output
CVP	5-10	mmHg	Central Venous Pressure
$CaCO_2$	0.46	ml/ml	Arterial CO_2 Concentration
CaO_2	0.20	ml/ml	Arterial O_2 Concentration
$CvCO_2$	0.5	ml/ml	Venous CO_2 Concentration
CvO_2	0.15	ml/ml	Venous O_2 Concentration
HR	72	B/min	Heart Rate
Hb	15	gr/dl	Haemoglobin concentration of blood
KL	—	mmHg/ml	Left Heart parameter
KR	—	mmHg/ml	Right Heart parameter
KV	—	mmHg/ml	Venous system parameter
PAPM	13	mmHg	Mean Pulmonary Arterial Pressure
PArtM	100	mmHg	Mean Arterial Pressure
PLA	6	mmHg	Left Atrium Pressure
PVR	0.0013	mmHg/(ml/min)	Pulmonary Vascular Resistance
RQ	0.8	—	Respiratory Quotient
SaO_2	100	%	Arterial O_2 saturation
SvO_2	70	%	Venous O_2 saturation
SV	76	ml	Stroke Volume
SVR	0.018	mmHg/(ml/min)	Systemic Vascular Resistance
$\dot{V}CO_2$	200	ml/min	Expiratory CO_2 Volume
$\dot{V}O_2$	250	ml/min	Inspiratory O_2 Volume
VV	4500	ml	Venous Volume

Table 1: Properties of model variables

[a] mmHg = milimeter mercury, B = beats

1. VV = KV $\times$ ln(CVP)
   A higher or lower venous volume (VV) results in a higher or lower central venous pressure (CVP). When VV gets larger, the veins are stretched and resist further filling. The above formula is derived from the assumption that the change in CVP due to a change in VV, is proportional to CVP: $\delta \mathrm{CVP}/\delta \mathrm{VV} \simeq \mathrm{CVP}$.

2. SV = KR $\times$ ln(CVP)
   For the right side of the heart a similar formula can be assumed for the relationship between the stroke volume (SV) and the CVP. This formula has been chosen to keep the model simple. In reality the stroke volume increases initially, but it decreases again above a certain level of the central venous pressure. This relationship is known as the Starling curve.

3. SV = KL $\times$ ln(PLA)
   This relationship is the equivalent for the left side of the heart.

4. PArtM − CVP = CO $\times$ SVR
   Ohm's law for the systemic circulation. The pressure difference between the mean arterial pressure (PArtM) and the CVP (potential difference) is equal to the cardiac

output (CO) multiplied by the systemic vascular resistance (SVR).

5. PAPM − PLA = CO × PVR
The equivalent relationship for the pulmonary circulation.

6. CO = SV × HR
The CO is the volume of one stroke multiplied by the heart rate (HR).

7. $CaO_2 = 0.000134 \times Hb \times SaO_2$
The arterial oxygen concentration (CaO_2) is proportional to the haemoglobin concentration (Hb) multiplied by the fractional arterial oxygen saturation (SaO_2). The constant 0.000134 is in dl/g.

8. $CvO_2 = 0.000134 \times Hb \times SvO_2$
The equivalent relationship for the venous system.

9. $\dot{V}O_2 = (CaO_2 - CvO_2) \times CO$
The inspiratory oxygen volume ($\dot{V}O_2$) is equal to the oxygen uptake, which is the difference in oxygen concentration between the arterial and the venous blood multiplied by the CO.

10. $\dot{V}CO_2 = (CvCO_2 - CaCO_2) \times CO$
The expiratory carbon dioxide volume ($\dot{V}CO_2$) is equal to the carbon dioxide production, which is the difference in carbon dioxide concentration between the venous and arterial blood multiplied by the CO.

11. $RQ = \frac{\dot{V}CO_2}{\dot{V}O_2}$
The respiratory quotient (RQ) is defined as $\dot{V}CO_2$ divided by $\dot{V}O_2$. When no changes take place in the total body content of carbon dioxide, it has a constant value of 0.8, since the metabolic rate of oxygen used is proportional to the quantity of carbon dioxide produced.

3 Introduction to RL

The language RL[6,7] integrates logic programming, algebraic constraint solving and relational databases. The user can combine equational constraints, Horn clauses and relational database operators to express a knowledge base. This provides maximal freedom and full conceptual transparency to the user.
Knowledge and queries expressed in RL are preprocessed by a constraint solver, and compiled into a database query. The database query is processed by an existing relational database system, so that large amounts of data can be processed efficiently.
An RL program consists of modules, each describing a system of relations. An RL query consists of a relational expression which is evaluated in the context of an existing program. RL/1 is a subset of RL. It eliminates some of the syntactic and semantic complexity, but it preserves the main idea of integrating relational databases, logic programming and algebraic constraint solving. The current implementation of RL/1 focuses on the integration of a

symbolic (numeric) constraint solver with a relational database system[3,4,5]. A prototype system has been implemented consisting of a compiler for RL/1, a constraint solver and a relational database that is coded in Prolog. A version that produces standard SQL is currently being developed. The examples presented in the next sections were processed with this system.

In RL/1 a distinction is made between *extensional* and *intensional* objects. Extensional objects correspond to base tables in the underlying relational database. Intensional objects are defined by *rules*, also called clauses. A rule consists of a *rule head* and a *rule expression* separated by the keyword WHEN. The rule expression can contain system defined predicates or invocations of intensional and extensional objects, separated by a *conjunctive* AND operator. If an intensional object is defined by more than one rule, the rules are *disjunctive.*

Queries in RL/1 result in an answer relation which consists of attributes equal to the attribute list between the keywords INFER and WHEN, and a (possibly empty) set of tuples. Optionally the result of a query command can be stored in a database table:

```
INFER ( attribute-list ) WHEN rule-expression [ TO table ]
```

An important feature of the RL/1 system is the capability to produce symbolic output. Processing a INFER query requires one or more invocations of the constraint solver. The solver output is then used to compile the query into a database request. The intermediate solver output can also be presented to the user directly. The syntax of a symbolic query is as follows:

```
SYMINFER ( attribute-list ) WHEN rule-expression [ TO file ]
```

The 'TO file' option allows the computed symbolic answer to be stored in a text file. Symbolic query commands result in the following answer:

```
condition= {Condition set}
solution=  {Solution set}
```

Either the condition part or the solution part can be absent. The solution set contains elements of the form $x = t$ with x a *wanted* variable and t a term. Wanted variables are to be eliminated from the constraint set. The condition part states under which additional restrictions the obtained solution is valid. The main goal of the constraint solver is to express the wanted variables in terms of *known* variables. The set of wanted variables is equal to the attribute list of the query. The known variables are determined by the rule expression of the query.

Variables occuring in the terms t of the solution set are a subset of the known variables. All wanted variables are present as one of the x variables in the solution set. The condition set consists of constraints that only contain known variables.

4 RL implementation of the model

The physiological model is represented in RL by a single rule defining the intensional object 's' as shown in Figure 2. The constraints have the same numbers as the model equations

in section 2. The '*' notation in the head of a rule indicates that all variables occurring in the associated rule expression are declared as attributes of the object. This notation can also be used in an invocation of an object and is then used to denote all attributes of the object.
In the symbolic queries below the predefined tables 'k1', 'k2', 'k3' ... will be used. If one of these tables appears in the rule expression of a query, the variables in the table become known. In addition tables are defined to group continuous measureable variables, intermittent measureable variables, non measureable variables and patient parameters. Their names are prefixed with a 'k'.

```
module phsystem(number:Ca02:CaC02:CvC02:Cv02:CO,
    number:CVP:HR:KL:KR:KV:PAPM:PArtM:PLA:PVR:RQ:Hb,
    number:Sa02:Sv02:SV:SVR:VC02:V02:VV).
table kintermit(CO,Hb,Sv02).
table kcontinue(PArtM,CVP,PLA,PAPM,HR,Sa02,VC02).
table kunknown(VV,SV,SVR,PVR,Ca02,CaC02,Cv02,V02,CvC02,RQ).
table kconstant(KV,KR,KL).
clause s(*) when
/* 1*/   VV=KV*ln(CVP)
/* 2*/   and SV=KR*ln(CVP)
/* 3*/   and SV=KL*ln(PLA)
/* 4*/   and PArtM-CVP=CO*SVR
/* 5*/   and APM-PLA=CO*PVR
/* 6*/   and CO=SV*HR
/* 7*/   and Ca02=0.0001134*Hb*Sa02
/* 8*/   and Cv02=0.0001134*Hb*Sv02
/* 9*/   and V02=(Ca02-Cv02)*CO
/*10*/   and VC02=(CvC02-CaC02)*CO
/*11*/   and RQ=VC02/C02.
close.
```

Figure 2: The physiological model in RL

5 Using the model

Representing a model in a declarative way enables its use for multiple purposes. This section discusses two ways in which a physiological constraint model can be used to assist anaesthetists in decision making: (1) the interpretation of measurements, and (2) the prediction of effects of treatment.

5.1 Interpretation of measurements

The anaesthetist tries to gain insight in the development of the physiological condition of a patient by monitoring the trend of physiological variables. Unfortunately not all variables can be measured. A physiological model, like the one presented in section 2, can be used to derive values for non measureable variables from the ones measured. The derivation of values for unmeasureable variables is a first step in the interpretation process which leads to the anaesthetists judgements and decision making.
The number of physiological variables which can be measured has increased over the past few decades. This development has raised a demand for computer systems that present all measured information in a coherent way[1,2]. The rapid development of new measurement equipment also requires that a computer system is adaptable to new measurement possibilities.
Physiological variables can be divided into four classes. In figure 1 these categories are reflected in the shapes of variables.

1. continuous measureable variables: in the University Hospital of Groningen a computer system[2] on the theatre stores average values for these variables at 1 minute intervals. These sets of values can be used as input for interpretation.

2. intermittent measureable variables: values of these variables can not be automatically measured, and the measurement methods available demand extra attention of the anaesthetist. They are therefore performed only when necessary. To determine bloodgasses for example, a sample of blood has to be taken. It takes about 18 minutes for a blood sample to be analyzed in the laboratory. Another example is cardiac output, which is measured by injecting cold fluid into the blood stream. This introduces extra volume and can only be done a few times during an operation.

3. patient parameters: variables whose values are characteristic for a particular patient and change so slowly that they can be assumed constant with respect to other variables.

4. unknown variables: variables that can be neither measured nor assumed to be constant.

The possibilities for measurement interpretation offered by a physiological constraint model will be illustrated with an example: the determination of the trend of the CO. This will be followed by a more systematic approach for measurement interpretation.

5.1.1 Example: determination of the CO-trend

Anaesthetists are interested in the trend of the CO. However CO can only be measured a few times during an operation. From figure 1 it can be seen that such an incidental measurement, combined with values for the continuous measureable variables HR, CVP and PLA, determines the patient parameters KR and KL. The computation of KR and KL can be done with RL by making CO, HR, CVP and PLA known variables and KR and KL wanted. The RL query and its solution is:

```
syminfer(KR,KL) when s(*) and k4(CO,HR,CVP,PLA).
solution=  { KR = CO * 1 / ln(CVP) * 1 / HR
             KL = CO * 1 / ln(PLA) * 1 / HR }
```

KR and KL can be assumed to remain equal to the values computed for them since they are parameters. Subsequent measurements for CVP, PLA and HR can be used to compute the CO. This is done with RL by making CVP, PLA, HR, KR, and KL known variables and CO wanted. The RL query and its result is:

```
syminfer(CO) when s(*) and k5(CVP,PLA,HR,KR,KL).
condition= { KR * ln(CVP) = KL * ln(PLA) }
solution=  { CO = HR * KL * ln(PLA) }
```

The solution contains an expression for the CO in terms of the known variables. The condition can be used to check whether the values measured for CVP and PLA are consistent with each other.

5.1.2 General measurement interpretation method

In the example above it was necessary to use the graph in figure 1 to decide which of the variables had to be made known and which of them wanted. The RL queries were tailored to the model and the measurements available. This is undesirable because it implies that new queries have to be developed each time the model is changed or new measurement possibilities become available. As this would practically nullify the advantages of a declarative representation, the following general method to interpret measurements with a constraint model was developed.

When measurements for a set of variables become available, they are made known to RL and all other variables are made wanted. RL produces solutions for variables that can be expressed in terms of the known variables. For the remaining non-measured variables a reduced model can be derived from the original one. In order to make RL perform the reduction, the following procedure is used:

- add equations for the available measurement values of the form 'variable$_i$ = measurement$_i$' to the original model, with 'variable$_i$' a model variable and 'measurement$_i$' a value.
- make the measured variables wanted and the variables for which a reduced model is searched known.

The solution derived by RL contains the added equations 'variable$_i$ = measurement$_i$'. According to the specification of the RL/1 system the condition contains constraints applying to the variables declared known in the query[3]. These constraints make up the reduced model.

The derived relations in the reduced model can be useful for decision support. A relation between two variables for example can be displayed as an X–Y plot. When derived relations are complex however, they can not be used as such for decision support. In that case assumptions have to be made about the values of variables or about the existence of

extra constraints. Addition of assumed constraints will be demonstrated in section 5.2 on prediction of treatment effects.
It can be assumed that variables keep the value determined earlier for them. These values can be used to obtain additional variable solutions and model reductions. The four classes of variables introduced earlier can be ranked according to the validity of the assumption that their values have remained equal. This ranking is used to select the group of variables whose values are used when more solutions or further reductions are needed.
When a set of values from continuous measureable variables becomes available the sequence in which variable classes are used for value derivation and model reduction are: (1) parameter values and (2) intermittent measureable values.
For new values of intermittent measureable variables the sequence is: (1) continuous measurable values, (2) parameter values and (3) other intermittent measureable values.
Application of this general strategy results in the same values for the trend of the CO as the informal strategy that was used previously. When the value 5000 for the variable CO becomes available, RL is first requested for solutions as follows:

```
syminfer(PArtM,CVP,PLA,PAPM,HR,Sa02,VC02,VV,SV,SVR,PVR,Ca02,CaC02,
            Cv02,V02,CvC02,RQ,Hb,Sv02,KV,KR,KL) when s(*) and CO=5000.
condition= {}
solution=  {}
```

None of the other variables can be expressed as a function of CO, so the solution set is empty. The next step is model reduction. CO whose values is known, is made wanted to RL and the other variables are declared known:

```
syminfer(CO) when s(*) and CO=5000 and kcontinue(*) and kunknown(*)
    and kconstant(*) and k2(Hb,Sv02).
condition= {
        VC02 = RQ * V02
        CaC02 * 5000 + CvC02 * -5000 + VC02 = 0
        Ca02 * -5000 + Cv02 * 5000 + V02 = 0
        Cv02 * -7462.686567 + Hb * Sv02 = 0
        Ca02 * -7462.686567 + Hb * Sa02 = 0
        HR * SV * -0.000200 = -1
        PAPM * -0.000200 + PLA * 0.000200 + PVR = 0
        PCVP * 0.000200 + PArtM * -0.000200 + SVR = 0
        KL * ln(PLA) = SV
        ln(CVP) * KR = SV
        ln(CVP) * KV = VV
}
solution=  { CO = 5000 }
```

The reduced model is essentially the original model with 5000 substituted for the CO. It contains a relation between SV and HR which can be presented as an X-Y plot.
The next step is performed with the last set of values for the continuous measureable variables. In the next query it is assumed that these are equal to the values in table 1.

```
syminfer(VV,SV,SVR,PVR,CaO2,CaCO2,CvO2,VO2,CvCO2,RQ,Hb,SvO2,KV,KR,KL) when
    s(*) and CO=5000 and PArtM=72 and CVP=5 and PLA=17 and PAPM=27 and
    HR=72 and SaO2=0.99 and VCO2=200.
condition= {}
solution=  { KR = 43.148259 , KL = 24.510842 }
```

When new sets of values of continuous measureable variables become available, the same three steps are performed. In the third step – the parameter substitution – the derived values for KL and KR are used. As shown below an estimate for the actual CO results. In the first query it is tried to derive values of other variables from the measurement values:

```
syminfer(VV,SV,SVR,PVR,CaO2,CaCO2,CvO2,VO2,CvCO2,RQ,CO,Hb,SvO2,KV,KR,KL)
    when s(*) and PArtM=72 and CVP=5 and PLA=17 and PAPM=27 and
    HR=72 and SaO2=0.99 and VCO2=200.
condition= {}
solution=  {}
```

There are no other variables that can be derived. The solution set is empty. Next the values are used for model reduction:

```
syminfer(PArtM,CVP,PLA,PAPM,HR,SaO2,VCO2) when
    s(*) and PArtM=72 and CVP=5 and PLA=17 and PAPM=27 and
    HR=72 and SaO2=0.99 and VCO2=200.
condition= {
        RQ * VO2 * -0.005000 = -1
        CO * CaCO2 * 0.005000 + CO * CvCO2 * -0.005000 = -1
        CO * CaO2 * -1 + CO * CvO2 + VO2 = 0
        CvO2 * -7462.686567 + Hb * SVO2 = 0
        CaO2 * -7538.067240 + Hb = 0
        CO * -0.013889 + SV = 0
        CO * PVR * -0.100000 = -1
        CO * SVR * -0.014925 = -1
        KL * -2.833213 + SV = 0
        KR * -1.609438 + SV = 0
        KV * -1.609438 + VV = 0 }
solution=  {
        PArtM = 72
        CVP = 5
        PLA = 17
        PAPM = 27
        HR = 72
        SaO2 = 0.99
        VCO2 = 200
}
```

The condition contains the reduced model. Next the parameter values are used to make further value inferences:

```
syminfer(CO,Hb,SvO2,VV,SV,SVR,PVR,CaO2,CaCO2,CvO2,VO2,CvCO2,RQ) when
    s(*) and PArtM=72 and CVP=5 and PLA=17 and PAPM=27 and HR=72 and
    SaO2=0.99 and VCO2=200 and KR=43 and KL=24.5 and KV=2750
condition= { FALSE }
solution=  {
        PVR = 0.002001
        SVR = 0.013406
        SV = 69.413718
        VV = 4425.954500
        CO = 4997.788344
}
```

The condition contains FALSE, this indicates that an inconsistency is detected. In section 5.1.1 a condition between KL, KR, CVP, PLA was derived, which is not satisfied by the given values. Inconsistencies are discussed in the next subsection.

5.1.3 Quantifying inconsistencies

RL derives a condition between the known variables as well as a solution. When the values of the measured variables do not satisfy the condition, they are inconsistent. Small inconsistencies are not alarming, since measurements always contain minor errors and models are only an abstraction of reality. The magnitudes of errors are of greater importance. They can be obtained by introducing variables for errors as follows:

- add equations of the form '$\text{error}_i = \text{value}_i - \text{variable}_i$' to the original model, where 'value_i' is a value which was measured or derived earlier and 'error_i' and 'variable_i' are variables,
- declare the error variables known to RL.

The RL query to find expressions relating the errors of variable values used in last query of the previous subsection is:

```
syminfer() when s(*) and EPArtM=72-PArtM and ECVP=5-CVP and EPLA=17-PLA
    and EPAPM=27-PAPM and EHR=72-HR and ESaO2=.99-SaO2 and EVCO2=200-VCO2
    and EKL=24.5-KL and EKR=43-KR and EKV=2750-KV
    and k10(EPArtM,EVCP,EPLA,EPAPM,EHR,ESaO2,EVCO2,EKL,EKR,EKV).
condition= {
    ln(ECVP * -1.000000 + 5) * -1.755102 + ln(ECVP * -1.000000
    + 5) * EKR * 0.040816 + EKL * ln(EPLA * -1.000000
    + 17) * -0.040816 + ln(EPLA * -1.000000 + 17) = 0
}
```

Identifiers for error variables are formed prefixing the variable names with an E. The condition contains an expression relating the errors of KR, KL, CVP and PLA. The magnitude of the inconsistency can be obtained by substituting zero for n − 1 errors in the condition and to use RL to compute the corresponding value for the remaining error.
When the previous query is extended with:

```
EPLA = 0 and ECVP = 0 and EKRA = 0
```

the result is:

```
solution= { EKL = 0.073378 }
```

The error value found relative to the value of KL is: 0.073378 / 24.5 x 100 % = 0.3%.

5.2 Prediction of treatment effects

Effects of a treatment can be divided into two groups:

- primary: effects that are a direct result of the treatment
- secondary: effects due to physiological reactions provoked by the primary effects.

When an assumption is made about the primary effects of a treatment, the magnitude of its secondary effects can be computed using a physiological constraint model. The secondarily affected model variables are made the wanted variables for RL, both the primary affected variables and the ones known to be unaffected by the treatment are made known. For the primarily affected variables a new value is supplied based on knowledge about the effects of the treatment. The values supplied for the other variables are equal to their current ones. Sometimes it is necessary to add assumptions about the propagation of primary effects to secondary effects.
An example of treatment is the infusion of a plasma expander. A plasma expander has the property to remain in its entirety in the circulation; it does not migrate to the interstitium. The shift in plasma volume is nearly entirely (99.5%) in the venous compartment. The introduction of extra venous volume provokes the following chain of events:

infusion $\rightarrow$ VV$\uparrow$ $\rightarrow$ CVP$\uparrow$ $\rightarrow$ SV$\uparrow$ $\rightarrow$ CO$\uparrow$ $\rightarrow$ PArtM$\uparrow$, PAPM$\uparrow$, $C_VO_2\uparrow$, $C_VCO_2\downarrow$
PArtM$\uparrow$ $\rightarrow$ SVR$\downarrow$
PAPM$\uparrow$ $\rightarrow$ PVR$\downarrow$

($\uparrow$ = increases and $\downarrow$ = decreases)

Physiological regulation compensates for the increase in PArtM and PAPM by a decrease of SVR and PVR. To enable computation of the new pressures and new resistances it is assumed that for the systemic circulation 60% of the change in CO is retrieved in a decrease of SVR and 40% in an increase in the pressure difference (PArtM – CVP). Lung resistance is more reactive to changes in CO; its decrease is brought about by perfusion of parts of the lung that were previously closed off. The figures for the lung circulation are 90% and 10%. These assumptions are added to the model by means of the following additional constraints:

$$CO/CO_{old} = \text{rate}$$
$$SVR_{old}/SVR = 60\% \times \text{rate}$$
$$PVR_{old}/PVR = 90\% \times \text{rate}$$

CO_{old}, SVR_{old} and PVR_{old} are fixed values, equal to the current values of the corresponding variables. Rate is a new variable. The consequences of the infusion of 1.5 liter plasma expander with all variables having their normal values as listed in table 1, can be predicted with the RL query:

```
syminfer(VO2,CVP,SV,CO,PArtM,PAPM,CvO2,CvCO2,SVR,PVR,RQ,CaO2,PLA) when
    s(*) and VV = 6000 and CO / 5500 = rate and 0.018 / SVR = 0.6 * rate
    and 0.013 / PVR = 0.1 * rate and KV = 2450 and KR = 43 and KL = 24.5
    and HR = 62 and SaO2 = 100 and CaCO2 = 0.46 and Hb = 15 and SvO2 = 0.70
    and VCO2 = 200.
condition= {}
solution=  {
        RQ = 0.437448
        CvCO2 = 0.486378
        VO2 = 457.197061
        CvO2 = 0.140700
        CaO2 = 0.201000
        CO = 7582.040816
        PAPM = 145.067920
        PArtM = 176.576528
        PLA = 73.567921
        SV = 105.306122
        CVP = 11.576528
        SVR = 0.021762
        PVR = 0.009430
}
```

Some of the resulting values exceed physiological limits. This shows the limitations of the simple model used.

6 Conclusion

A considerable amount of physiological knowledge exists in algebraic form. A constraint satisfaction program enables the use of an algebraic model for various purposes.
The language RL provides a framework in which algebraic constraints can be combined with logic programming and relational databases. The prototype implementation RL/1 contains a constraint satisfaction module that generates symbolic solutions, which were used to interpret patient measurements, to predict treatment effects and to trace inconsistencies between measurements.
Substitution of measured variable values results in values for other variables and in a reduced model. When the reduced model is still too complex to provide decision support, extra assumptions have to be made. When it is assumed that variables of a particular class have kept the values that were last found for them, values of other variables can be infered.
Other forms of assumption are possible. The physician can do what–if reasoning: "if a particular value is assumed for a variable, what are the corresponding values for other

variables?" The physician can also add extra constraints to the model. This was shown in the prediction of treatment effects.
Interaction with the anaesthetist is required about intermediate results and further assumptions to be made. As time in the operating theatre is limited, an optimal interface is essential. Effective interaction must be possible however, because making assumptions is the way in which physicians cope with incompleteness of information. A constraint solver provides a tool to support this.

7 References

1. Annejet P. Meyler, *Automation in Anesthesia, a Relief?; evaluation of a data acquisition and display system*, thesis of the university of Eindhoven, The Netherlands, ISBN 90-9001441-1, 1986.

2. G.F. Karliczek, A.F. de Geus, G. Wiersma, S. Oosterhaven & I. Jenkins, *Carola, a computer system for automatic documentation in anesthesia*, Int. J. Clin. Monitoring and Computing 4, 211–221, (1987).

3. Van Denneheuvel, S. & Van Emde Boas, P., *Constraint solving for databases,* Proc. of NAIC 1, 173–184, (1988).

4. Van Denneheuvel, S. & Van Emde Boas, P., *The rule language RL/1,* A. M. Tjoa & R. Wagner (eds.), Database and Expert System Applications, Proc. of the International Conference, Vienna, Austria, Aug 1990, Springer Verlag Wien, 381–387, (1990).

5. Van Denneheuvel, S., Van Emde Boas, P., Geus, F. de, Rotterdam, E., *RL, a Language for Constraint Solving, Logic Programming and Database Processing,* Computing Science in the Netherlands, Nov 1990, (to appear).

6. Van Emde Boas, P., *RL, a Language for Enhanced Rule Bases Database Processing,* Working Document, Rep. IBM Research, RJ 4869 (51299), (1986).

7. Van Emde Boas, P., *A semantical model for the integration and modularization of rules,* Proceedings MFCS 12, Bratislava, Springer Lecture Notes in Computer Science 233, 78–92, (1986).

8. Van Emde Boas, H. & Van Emde Boas, P., *Storing and Evaluating Horn Clause Rules in a Relational Database*, IBM J. Res. Develop. 30 (1), 80–92, (1986).

The ITLI Prepublication Series

1990

Logic, Semantics and Philosophy of Language	
LP-90-01 Jaap van der Does	A Generalized Quantifier Logic for Naked Infinitives
LP-90-02 Jeroen Groenendijk, Martin Stokhof	Dynamic Montague Grammar
LP-90-03 Renate Bartsch	Concept Formation and Concept Composition
LP-90-04 Aarne Ranta	Intuitionistic Categorial Grammar
LP-90-05 Patrick Blackburn	Nominal Tense Logic
LP-90-06 Gennaro Chierchia	The Variablity of Impersonal Subjects
LP-90-07 Gennaro Chierchia	Anaphora and Dynamic Logic
LP-90-08 Herman Hendriks	Flexible Montague Grammar
LP-90-09 Paul Dekker	The Scope of Negation in Discourse, towards a flexible dynamic Montague grammar
LP-90-10 Theo M.V. Janssen	Models for Discourse Markers
LP-90-11 Johan van Benthem	General Dynamics
LP-90-12 Serge Lapierre	A Functional Partial Semantics for Intensional Logic
LP-90-13 Zhisheng Huang	Logics for Belief Dependence
LP-90-14 Jeroen Groenendijk, Martin Stokhof	Two Theories of Dynamic Semantics
LP-90-15 Maarten de Rijke	The Modal Logic of Inequality
LP-90-16 Zhisheng Huang, Karen Kwast	Awareness, Negation and Logical Omniscience
LP-90-17 Paul Dekker	Existential Disclosure, Implicit Arguments in Dynamic Semantics
Mathematical Logic and Foundations	
ML-90-01 Harold Schellinx	Isomorphisms and Non-Isomorphisms of Graph Models
ML-90-02 Jaap van Oosten	A Semantical Proof of De Jongh's Theorem
ML-90-03 Yde Venema	Relational Games
ML-90-04 Maarten de Rijke	Unary Interpretability Logic
ML-90-05 Domenico Zambella	Sequences with Simple Initial Segments
ML-90-06 Jaap van Oosten	Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
ML-90-07 Maarten de Rijke	A Note on the Interpretability Logic of Finitely Axiomatized Theories
ML-90-08 Harold Schellinx	Some Syntactical Observations on Linear Logic
ML-90-09 Dick de Jongh, Duccio Pianigiani	Solution of a Problem of David Guaspari
ML-90-10 Michiel van Lambalgen	Randomness in Set Theory
ML-90-11 Paul C. Gilmore	The Consistency of an Extended NaDSet
Computation and Complexity Theory	
CT-90-01 John Tromp, Peter van Emde Boas	Associative Storage Modification Machines
CT-90-02 Sieger van Denneheuvel, Gerard R. Renardel de Lavalette	A Normal Form for PCSJ Expressions
CT-90-03 Ricard Gavaldà, Leen Torenvliet, Osamu Watanabe, José L. Balcázar	Generalized Kolmogorov Complexity in Relativized Separations
CT-90-04 Harry Buhrman, Edith Spaan, Leen Torenvliet	Bounded Reductions
CT-90-05 Sieger van Denneheuvel, Karen Kwast	Efficient Normalization of Database and Constraint Expressions
CT-90-06 Michiel Smid, Peter van Emde Boas	Dynamic Data Structures on Multiple Storage Media, a Tutorial
CT-90-07 Kees Doets	Greatest Fixed Points of Logic Programs
CT-90-08 Fred de Geus, Ernest Rotterdam, Sieger van Denneheuvel, Peter van Emde Boas	Physiological Modelling using RL
Other Prepublications	
X-90-01 A.S. Troelstra	Remarks on Intuitionism and the Philosophy of Mathematics, Revised Version
X-90-02 Maarten de Rijke	Some Chapters on Interpretability Logic
X-90-03 L.D. Beklemishev	On the Complexity of Arithmetical Interpretations of Modal Formulae
X-90-04	Annual Report 1989
X-90-05 Valentin Shehtman	Derived Sets in Euclidean Spaces and Modal Logic
X-90-06 Valentin Goranko, Solomon Passy	Using the Universal Modality: Gains and Questions
X-90-07 V.Yu. Shavrukov	The Lindenbaum Fixed Point Algebra is Undecidable
X-90-08 L.D. Beklemishev	Provability Logics for Natural Turing Progressions of Arithmetical Theories
X-90-09 V.Yu. Shavrukov	On Rosser's Provability Predicate
X-90-10 Sieger van Denneheuvel, Peter van Emde Boas	An Overview of the Rule Language RL/1
X-90-11 Alessandra Carbone	Provable Fixed points in $I\Delta_0+\Omega_1$, revised version
X-90-12 Maarten de Rijke	Bi-Unary Interpretability Logic
X-90-13 K.N. Ignatiev	Dzhaparidze's Polymodal Logic: Arithmetical Completeness, Fixed Point Property, Craig's Property
X-90-14 L.A. Chagrova	Undecidable Problems in Correspondence Theory
X-90-15 A.S. Troelstra	Lectures on Linear Logic