Forschungsbericht
Institut für Automatisierungs- und Softwaretechnik

Hrsg.: Prof. Dr.-Ing. Dr. h. c. P. Göhner

Mario Rebolledo

Situation-based Process Monitoring in Complex Systems Considering Vagueness and Uncertainty

Band 2/2004

IAS-Forschungsberichte

Universität Stuttgart

Situation-based Process Monitoring in Complex Systems Considering Vagueness and Uncertainty

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von
Mario Rafael Rebolledo Wueffer
aus Maracay, Venezuela

Hauptberichter:	Prof. Dr.-Ing. Dr. h. c. Peter Göhner
Mitberichter:	Prof. Dr.-Ing. Jörg Roth-Stielow
Tag der Einreichung:	08.06.2004
Tag der mündlichen Prüfung:	03.11.2004

Institut für Automatisierungs- und Softwaretechnik
der Universität Stuttgart

2004

IAS-Forschungsberichte

Band 2/2004

Mario Rebolledo

Situation-based Process Monitoring in Complex Systems Considering Vagueness and Uncertainty

D 93 (Diss. Universität Stuttgart)

Gedruckt mit Unterstützung des Deutschen Akademischen Austauschdienstes

Shaker Verlag
Aachen 2004

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the internet at http://dnb.ddb.de.

Zugl.: Stuttgart, Univ., Diss., 2004

Printed in Germany.

ISBN 3-8322-3518-3
ISSN 1610-4781

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • eMail: info@shaker.de

Die vorliegende Arbeit entstand während meiner Tätigkeit als Wissenschaftlicher Mitarbeiter am Institut für Automatisierungs- und Softwaretechnik (IAS) der Universität Stuttgart.

Mein besonderer Dank gilt

Herrn Prof. Dr.-Ing. Dr. h. c. P. Göhner, Leiter des Instituts, für die Betreuung dieser Arbeit, die Übernahme des Hauptberichts und die zahlreichen und wertvollen Anregungen und kritischen Diskussionen, die es ermöglichten, dass die Arbeit in der vorliegenden Form durchgeführt werden konnte,

Herrn Prof. Dr.-Ing. J. Roth-Sielow, Leiter des Instituts für Leistungselektronik und Elektrische Antriebe, für die Übernahme des Mitberichts und die Korrektur der Dissertationsschrift,

Faye Costarell, die bei der Sprachkorrektur der Dissertationsschrift diese Arbeit immer als ihre eigene betrachtet hat,

allen Kolleginnen und Kollegen am Institut für die gute Zusammenarbeit und hilfsbereite Unterstützung,

den Studenten, die im Rahmen von Diplomarbeiten und Master Theses zum Gelingen dieser Arbeit beitrugen,

dem Deutschen Akademischen Austauschdienst (DAAD) für die finanzielle Unterstützung und Förderung des Forschungsvorhabens,

Paulo Urbano und Familie Ruge Rebolledo für ihre Hilfe in den entscheidenden Momenten,

und Libia, Mario, Alejandro und meiner Eltern und Geschwister für ihre Geduld, ihren Ansporn und ihr Beisein, das es ermöglichte, immer ein Stück von Venezuela bei mir zu haben.

Stuttgart, im Juni 2004

Mario Rebolledo

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3TS	**T**hree-**T**ank **S**ystem
AI	**A**rtificial **I**ntelligence
ANFIS	**A**daptive **N**euro-**F**uzzy **I**nference **S**ystem
ARMA	**A**uto **R**egressive **M**oving **A**verage
ARMAX	**A**uto **R**egressive **M**oving **A**verage with e**X**ternal inputs
ARX	**A**uto **R**egressive with e**X**ternal inputs
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
CBR	**C**ase-**B**ased **R**easoning
C	Rule **C**onfidence value
DFS	**D**ynamic **F**uzzy **S**ystems, by Krebs and Schäfers [KrSc98]
e.g.	Latin voice *"**e**xempli **g**ratia"*, meaning for example. It is used to introduce an illustrative statement
i.e.	Latin voice *"**i**d **e**st"*, meaning that is or that means. It is used to introduce a clarifying statement
FI	**F**uzzy **I**nterval
FIA	**F**uzzy **I**nterval **A**rithmetic
FIS	**F**uzzy **I**nference **S**ystem
FIR	**F**uzzy **I**nductive **R**easoning by Nebot [Nebo94]
FRenSi	**F**uzzy **Re**gio**n** **Si**mulation by Keller Leitch and Wyatt [KeLe99]
FuSim	**Fu**zzy (qualitative) **Sim**ulation by Shen and Leitch [ShLe93]
GSPN	**G**eneralized **S**tochastic **P**etri **N**et
GSPS	**G**eneral **S**ystem **P**roblem **S**olver by Klir [Klir85]
IAS	Institute of Industrial Automation and Software Engineering of the University of Stuttgart, from its original German denomination "**I**nstitut für **A**utomatisierungs- und **S**oftwaretechnik"
LAI	**L**ower **A**pproximation **I**nterval

MATLAB	**MAT**rix **LAB**oratory
MIC	**M**ethyl **I**so-**C**yanate
MIMO	**M**ultiple **I**nput and **M**ultiple **O**utput
PC	**P**ersonal **C**omputer
QSim	**Q**ualitative **Sim**ulator by Kuipers [Kuip86]
QuaSi	**Qua**litative **Si**mulator by Bonarini and Bontempi [BoBo94]
RI	**R**ough **I**nterval
RIA	**R**ough **I**nterval **A**rithmetic
s.	**S**ee reference (figure, table, section/chapter, etc)
SPN	**S**tochastic **P**etri **N**ets
SQMA	Situation-based Qualitative Modeling and Analysis, from its original German denomination "**S**ituationsbasierte **Q**ualitative **M**odellbildung und **A**nalysis"
SQMD	**S**ituation-based **Q**ualitative **M**odeling and **D**iagnosis
UAI	**U**pper **A**pproximation **I**nterval

Glossary

A priori: Deductive reasoning; the process of reasoning made from assumed principles or effects without facts or experience, knowledge or concepts applied prior to experiences before or without examination, hypothesis or theory not supported by observed factual study, without facts or experience.

Boolean: Expression of two possibilities. Data type or variable in programming language that can only have one of two possible values: true or false. Adjective describing a logical system treating variables such as propositions and computer logic elements through the operators AND, OR, NOT, XOR, IF…THEN and combinations of these.

Class: Set, collection, group or configuration containing members with certain common characteristics. Kind, category, group or family of these members. Arrange, group or rate according to qualities or characteristics. A set of objects that share a common structure and behavior.

Exactness: Quality of a measurement or calculus process that denotes the degree of correspondence between the obtained value with the real value of the event or dimension. It is defined in absolute terms.

Framework: Set of assumptions, concepts, values and practices that constitute a way of viewing reality.

Fuzzy: Adjective meaning unclear or lacking definition. Throughout this work, "fuzzy" is the term defined by L. A. Zadeh in 1965 for techniques he developed for a multivalued logic to express and manage vagueness.

Fuzzy Model: Model based on fuzzy sets and/or fuzzy numbers.

Geometrical Explosion: Occurs when the multiplicity of phenomena, elements or possibilities in a system or problem grows out of control. Also called "Numerical", "Exponential", "Combinatorial" or "Polynomial" explosion.

Granularity: Information granularity is the resolution at which an observation is made, or given information is represented. The higher the resolution the lower the granularity, and vice versa.

Heuristic: Scholarly attempt to solve a given problem, but without guarantee of success. The strategy is often based on experience and common sense, and is appropriately used when

dealing with complex problems, when there is no analytic solution, or when existing algorithms are too complicated.

Imperfect Information: Encompasses all problems that can be present in information that is gathered, stored or managed about a real system. Also called imperfect data.

Imprecision: Used in this document as a case of vagueness (see precision and vagueness below) in computed or measured values.

Industrial Automation System: The combination of technical system, automation hardware and software (control system, safety-related applications, information management, etc.), and humans who operate the technical processes.

Interacting modeling method: Modeling method, which considers interdependencies between variables and components. By contrast, **noninteracting modeling methods** are incapable of representing interdependencies between variables and components.

Markov chain: A model of sequences of events where the probability of an event occurring depends on the fact that a preceding event occurred. Named after Andrei Markov.

Markov process: Process governed by a Markov chain. A process where each state change is determined only by a given transition probability. State transition history plays no role in system dynamics.

Model: Schematic description of a system, theory or phenomenon that accounts for known or inferred properties. A copy, resemblance, or example, more or less exact. Simplified description of a complex process; e.g. a working model of a machine can do, on a small scale, the work the machine does or is expected to do. Description of observed behavior simplified by ignoring certain details (**approximate model**). Models allow understanding and predicting the behavior of complex systems within the scope of the model, but may give incorrect descriptions and predictions for situations outside the realm of their intended use.

Possibility: In logic and mathematics is the collection of related theories and concepts that emerge from the Fuzzy Set Theory, but beyond it, deals with incomplete information and uncertainty, instead of with vagueness and imprecision. By definition, the possibility is a fuzzily elastic constraint for the values that can be assigned to a variable.

Precision: Quality of an instrument that denotes the degree of correspondence in the expression of a measured or calculated value and the real (but generally unknown) result of the corresponding appraisal. Precision also concerns the minimal deviation that can be assessed in a variable. Precision is defined relative to the representation framework and

instruments used by measuring or calculating. Not to be confused with **exactness**, which is the real value and does not consider the assessment procedure. *Adjective*: **Precise**.

Probability: Measuring or determining, quantitatively, the likelihood an event or experiment will have a particular outcome. The idea of mathematical probability began in the 17th-century, with Bayes, Pascal and Fermat, attempting to answer questions arising in games of chance.

Process: Totally operative and well-delimited part of a system, which is capable of transforming and storing material, energy or information.

Process State: Minimal set of independent process variables, whose current values unmistakably describe the process behavior and determine its future, if no further factors influence the process.

Qualitative modeling: A qualitative reasoning technique, developed in the 1980's to represent incomplete and inaccurate data in models of complex systems. A qualitative model uses variables with qualitative values and qualitative operators, avoiding problematic numerical integration of differential equations.

Reachable states: In nondeterministic automata, are process states that can be reached from the current process state.

Safety-critical Automation System: System performing safety-critical tasks in order to decrease or eliminate hazards that may endanger humans, the environment, automation equipment, or industrial installations.

Set: Collection of elements or objects related through an idea, concept or classification criterion. In 1933, Kolmogorov's axioms defined probabilities in terms of sets of events, which are also applicable for other kinds of sets (such as fuzzy, classic and rough sets). Set terminology includes:

- Member or Element	An elementary object of a set
- Universe of discourse	Collection of all possible objects and sets
- Subset	A collection of objects contained within a set
- Complement of A	All objects that are not in A
- Intersection $A \cap B$	The collection of objects that are in both A and B
- Union $A \cup B$	The collection of objects either in A or B (or both)
- Empty set $\varnothing$	A set containing no elements

Simulation: Imitation or representation of a potential behavior or an experimental testing. Theoretical account based on a similarity between the model and the phenomena that are to be explained. Attempt to predict behavior aspects of a system by creating a model aimed at a particular application case.

Spurious situations: Situations which are reachable according to the realized computations, but that, as a matter of fact, cannot take place in the real systems because of not considered constraints. Spurious situations are characteristic of nondeterministic and noninteracting models. From **spurious**, adjective meaning corresponding to something without having its genuine qualities; of falsified or erroneously attributed origin; or of a deceitful nature or quality.

State automaton: Collection of states including the initial state, input events, output events, and a state transition function. A state automaton takes the current state and input events and returns the new set of system outputs and the next state. A state automaton (also called state machine) can also be defined as a function that maps an ordered sequence of input events into a corresponding sequence of output values.

System: Uppermost organization level of processes, which summarizes the entire technical processes.

Technical Process: Defined as "A process in which physical parameters are recorded and influenced by technical means" [DIN66201].

Uncertain: Unreliable, untrustworthy, dubious, doubtful or not having certain knowledge. *Noun*: **uncertainty**.

Universe of discourse: Subject of a database or model, i.e. the part of the "world" under discussion. There is no need for the universe of discourse to model concrete objects, or even part of the natural world. Nor is there a requirement that the universe of discourse completely map all aspects of the subject world, though many databases adopt the additional assumption that they are complete representations (the "closed world" assumption).

Vague: Not clearly expressed, not clearly defined, stated in indefinite terms, does not have a precise meaning. Also means not grasped or understood, indistinct, not sharply outlined, hazy, obscure. *Noun*: **vagueness**.

Zusammenfassung

Die Geschichte hat immer wieder gezeigt, dass die industrielle Entwicklung Risiken mit sich bringt, die nicht übersehen werden sollten. Diese Risiken betreffen die Gefährdung von Menschen, Umwelt und den Produktionseinrichtungen selbst. Aus diesem Grund sollte die beständige Entwicklung neuer Produktionstechnologien durch eine angemessene Entwicklung industrieller Sicherheitsmethoden begleitet werden.

Sicherheitssysteme für komplexe Prozesse basieren in der Regel auf einer präzisen Überwachung der Prozesszustände in Bezug auf "korrektes" Verhalten. Das Problem besteht darin, zu bestimmen, wann ein Prozessverhalten korrekt ist. Für diesen Zweck werden häufig Prozessmodelle benutzt, die das jenige Prozessverhalten widerspiegeln, welches aus Sicht der Produktion als „sicher" oder „geeignet" bewertet werden kann.

Die Entwicklung präziser Modelle wird allerdings aufgrund der zunehmenden Komplexität der zu überwachenden Prozesse beständig schwieriger. Existierende Methoden können den Komplexitätsgrad aktueller Produktionssysteme kaum handhaben. Die Anwendbarkeit von analytischen Modellierungsmethoden ist auf sehr einfache Prozesse begrenzt. Empirische Modelle, die z.B. auf Statistiken oder Techniken der künstlichen Intelligenz basieren, müssen auf die Darstellung von einzelnen Variablen oder kleinen Teilsystemen beschränkt werden, damit die Modelle handhabbar bleiben und nutzbare Informationen liefern. Für die Überwachung komplexer Prozesse werden häufig qualitative Modelle verwendet. Diese Modelle basieren auf der Abstraktion von Prozessinformationen auf Basis von Wertebereichen, die das Prozessverhalten symbolisch darstellen. Jedoch ist die Anwendbarkeit qualitativer Modellierungsmethoden wegen der Größe der resultierenden Modelle auch teilweise eingeschränkt.

In dieser Forschungsarbeit wird ein neuer Ansatz für die Prozessüberwachung auf Basis qualitativer Modelle vorgeschlagen. Er integriert zweckmäßig vage und ungewisse Informationen ins Modell, die ansonsten während der Modellierung vernachlässigt werden würden. Die vorgeschlagene Methode integriert Elemente der Rough-Sets-Theorie und der stochastischen qualitativen Automaten in das Situationsbasierte Qualitative Modellbildungs- und Analyseverfahren (SQMA). Die resultierenden Modelle sind deutlich präziser als qualitative Modelle ähnlicher Größe. Ebenso erlaubt die neue Methode die Entwicklung von qualitativen Modellen, die - bei vergleichbarer Genauigkeit - kompakter als traditionelle qualitative Modelle sind. Damit wird die Anwendbarkeit der SQMA-Prozessüberwachung für Systeme erweitert, welche eine größere Komplexität als die aufweisen, die mit den aktuellen Techniken überwacht werden können.

Abstract

History has demonstrated during the 20th century that industrial development carries hazards that should not be ignored because they endanger humans, the environment and production facilities. For this reason, continuous development of new production technologies should be accompanied by a comparable development in industrial safety technologies.

Safety-critical applications in complex processes are usually based on a precise monitoring of operation conditions, according to a “correct” process operation. The problem is determining if a behavior or an operation condition is “correct”. For this, models are generally used, which are able of reproducing “safe” or “appropriate” process behaviors.

The difficulty of precise modeling grows continuously, because of the increasing complexity of the supervised processes. Rigorous deterministic modeling is limited to simple processes, while approximate models based on statistics or Artificial Intelligence techniques, for example, must be restricted to modeling single variables or small subsystems to be manageable and deliver useful information. A monitoring technique usually employed for complex processes relies on abstraction of the process behavior in qualitative models by using symbolic value ranges to represent required information. However, also the applicability of qualitative modeling techniques is eventually restricted by the resulting model size.

In this research work, a new process monitoring approach, based on qualitative models, efficiently depicts valuable vague and uncertain information that is currently discarded during the modeling. The proposed method expands the ability of Situation-based Qualitative Modeling and Analysis (SQMA) to monitor complex processes by integrating elements of the Rough Set Theory and Stochastic Qualitative Automata. The resulting models are considerably more precise than other similar-sized qualitative models. At the same time, the new method develops more compact and precise qualitative models than traditional qualitative models of the same precision.

1 Introduction

"... every set of phenomena can be interpreted consistently in various ways, in fact, in infinitely many ways. It is our privilege to choose among the possible interpretations the ones that appear to us most satisfactory, whatever may be the reasons for our choice. If scientists would remember that various equally consistent interpretations of every set of observational data can be made, they would be much less dogmatic than they often are, and their beliefs in a possible ultimate finality of scientific theories would vanish".

F.R. Moulton (1939) cited by Bezdek [Bez94].

1.1 Significance of model-based process monitoring

The modern chemical industry began in Germany, between 1908 and 1912, when catalysts were used to combine nitrogen and hydrogen, to form ammonia. This launched an era of chemical and nuclear technology. The development of the world's first synthetic fertilizers started in 1913, when BASF opened the first synthetic ammonia factory at Oppau, Germany. The little town grew as more workers were employed to process a mixture of nitrogen and ammonia to manufacture ammonium nitrate fertilizer. Although the mixture of ammonia and nitrogen is flammable, it is difficult to ignite. Nobody worried about the huge stockpiles of a compacted mixture of ammonium sulfate and ammonium nitrate stored at the Oppau site, and the compound was believed to be so safe that, when the mixture became compacted, blasting powder was used to loosen the compound. This procedure was repeated thousands of times, and nothing happened. Then something went wrong on September 21, 1921. Two devastating explosions rocked Oppau, killing approximately 600, injuring 1.500, and leaving thousands homeless, and out of work [LRP01].

Lessons were not learned after the Oppau disaster. In July 1976, an accident at the Suisse Roche chemical factory near Seveso, Italy, that manufactures pesticides and herbicides, released a cloud of dioxin into the environment. The company kept the incident a secret for eight days. Although no known deaths were reported, at least 200 people suffered from the consequences. Pregnant women at Seveso feared their fetuses would be harmed by the poisonous and carcinogenous chemical, and had abortions [Samb04]. As word spread, nations throughout Europe were alarmed by what happened at Seveso, because major disasters are transboundary, and could affect the land and water in many countries. They joined forces in 1982, calling themselves SEVESO in honor of the besieged village. Yet, in spite of joint efforts to decrease the impact of manufacturing disasters, more tragic industrial incidents have occurred.

The world can hardly forget the industrial disaster in Bhopal, India, on the evening of December 2, 1984, when a series of mechanical and human failures [Tris01] caused the release of about 40

tons of methyl isocyanate (MIC) gas from an underground storage tank. This accident took the lives of more than 10.000 persons, and the cost of damages was over 1 billion US$.

Another example of dangerous industrial processes was the explosion at the nuclear power plant in Chernobyl, Ukraine, on April 7, 1986. Again, the real proportion of the catastrophe remains unknown, because the former USSR refused to release information to the public. The actual death toll is unknown. Apart from the 31 people who died in the explosion that day, 134 people fell ill shortly afterward because of acute radiation, and 29 died within three months. Nearly 8.400.000 people were exposed to radiation according to the United Nations [UN04]. Thyroid cancer afflicted 628 children during the following years; that is 614 more than probabilistically expected[1]. The economic consequences of Chernobyl are estimated at 300 billion US$.

Regretfully, Oppau, Seveso, Bhopal and Chernobyl are not isolated cases. The end of 20th century has seen many of these industrial accidents [Rena01]:

- **November 1984 (Mexico):** Toxic gases in a storage center caused about 500 fatalities and 7.000 injured persons.
- **July 1990, Houston (USA):** The third of a series of explosions in chemical plants in the region that killed 40 people in nine months.
- **November 1992, Berre (France):** An explosion at a Total refinery killed six people.
- **March 1993, Frankfurt (Germany):** One dead and one gravely injured person were the result of an explosion in a Hoechst chemical plant. Within six weeks, seven more incidents took place in Hoechst locations.
- **June 1995, Zemun (Serbia):** Explosion of the Grmec chemical product factory left a record of ten fatalities and many injuries.
- **September 2001, Toulouse (France):** Explosion at TotalFinaElf Grande Paroisse factory killed 29 people and injured 600.

History has demonstrated that industrial development carries hazards that cannot be ignored, and many are preventable. Various strategies have been developed, in an attempt to reduce the number of incidents, or decrease the risk of their occurrence. The advent of computers has enhanced the ability to combine human knowledge about materials used and hazardous conditions that can occur during manufacturing processes. One of these strategies is the monitoring of the operation conditions in such industrial processes.

Close condition monitoring prevents many industrial accidents, and therefore reduces the number of on-the-job injuries, decreases the severity of hazardous-conditions and economic

---

[1] Statistical information provided by "The Tschernobyl Initiative" [MaWo00]. The authors remark the difficulties by assessing the impact of this catastrophe in the long term, mainly because of its sociopolitical implications.

losses due to equipment failure, and helps avoid the violation of government regulations. Accurate monitoring of process operating conditions makes possible the early recognition of faulty behaviors, improves safety and reliability in industrial processes, preserves human beings, and protects the environment and manufacturing installations from harm.

Process condition monitoring, similar to many other industrial processes such as simulation, operation planning, advanced control and fault diagnosis, usually relies on models that describe the technical installation and its behavior. Working on models is always cheaper, faster, easier and safer than working with real processes. Besides, sometimes testing on a running real process is not an option.

The importance of how precisely the model describes the technical installation and its behavior is in process monitoring indisputable. Substitution of the real process with a model is only possible when the model reproduces functions and characteristics of the original system with precision that is equal to, or greater than, the minimal precision required by the application. Precision of a model, i.e. how well this model reproduces system behavior or characteristics, is a requirement established by the intended application.

However, modeling real processes is not always straightforward. Many systems are too complex to be described, meaningfully and wholly, with precise deterministic models. Additionally, only approximate, inaccurate and unwritten descriptions of a system are, most of the time, available. Modeling becomes even more difficult if alternative process behaviors must be taken into account. The dilemma can be resolved by creating a model that focuses upon the system aspects to be analyzed, enhanced or supervised. Such models utilize the ability of humans to describe complex processes with expert knowledge about the system and common sense, via curves, rules or tables.

The quality and applicability of a model-supported monitoring solution depends strongly on characteristics of the model and the modeling method. Qualitative models, fuzzy sets and probabilistic reasoning systems are some of the methods used to represent the expert knowledge in a model suitable for industrial applications.

1.2 Vague and uncertain information in industrial processes

In this research work, the term vagueness comprises approximate, inaccurate and sometimes unwritten process information; uncertainty represents unpredictable, nondeterministic behavior due to unknown parameters in a real system. Vagueness and uncertainty are significant obstacles in the modeling of real systems, as it is illustrated using a typical adsorption[2] process.

---

[2] The word adsorption derivates from adherence, and must not to be confused with absorption, where a substance is absorbed by (taken into) a porous body, as it happens in a sponge.

Adsorption is a filtration process that consists of several layers of adsorbents, such as granules of activated carbon or aluminum oxides, to remove unwanted substances from gases or liquids. As the substance is forced, by pressure, through adsorbent layers, molecules of unwanted components are attracted to the adsorbent particles. Depending upon the amount of impurities, as time passes, the adsorbed molecules eventually saturate the adsorbent particles until the filtration process is no longer effective. The adsorption particles must be regenerated before the purification procedure continues to work effectively and efficiently.

Safety plays an important role in adsorption processes, because many of the processed chemicals are flammable fuels or highly toxic substances. The risk of reaching undesirable or dangerous situations increases during the actual processing. In order to protect employees, the factories, public dwellings near the factory, and the environment, potential hazardous situations are often modeled. The problem is that the described phenomena cannot be observed by the human eye, because it occurs on a microscopic scale. At most, consequences are observed and interpreted. For example, temperature profiles in the adsorption layers can be continuously observed and compared with curves or tables that "approximately" describe the unit behavior. Even this temperature profile has a limited precision because it is acquired through a finite number of temperature sensors distributed in the adsorption unit. Consequently, only temperature intervals and discrete temperature points can be analyzed. Although the information about the process is vague, it is helpful. Yet, uncertainty exists as well, because so many variables are involved in the process (such as size, form and location of each adsorbent particle) that it is impossible to predict exactly when the temperature will change during the process, or to what degree the temperature change will occur.

Adsorbent regeneration can also be controlled, based on the before-mentioned curves and tables that describe typical absorbent behavior. Medium saturation time can be probabilistically characterized to define sub-optimal washing cycles. Although it is impossible to know, beforehand, exactly when the absorber will be saturated, it is possible to keep the process operation within specific parameters and bounds. Here handling uncertainty can help solve the problem.

The example can be further analyzed by considering how precisely a process variable must be controlled. Assuring that this variable remains exactly in a desired value, if at all possible, requires expensive equipment and complicated control strategies. A real process contains infinite information about pressures, temperatures, cross sections, weather influence, material compositions, etc, but often the data cannot be measured. Furthermore, a process value is only as accurate as the instrument used to measure it. In consequence, gathered process information is always incomplete.

In praxis, control strategies rely on operation bounds rather than on exact set points. This is the case of many temperature and composition control problems, where the process is required to

remain between two predefined values. As a result, the process is constrained to operate according to a group of curves or intervals that vaguely describe its behavior. This predicament, illustrated with the help of a control strategy, may be present in cases as dissimilar as designing a process after specifications and quality assurance, as well as in simulation, forecasting and process monitoring applications.

The intrinsic imprecision of real processes requires consideration of alternative behaviors in sensitive nonlinear processes such as pH-neutralization, where small variations in process parameters and variables can cause significant changes in the process behavior. Another problem is that many industrial processes such as adsorbers and catalytic reactors are too complex for rigorous modeling. In both cases, the process is determined, among other features, by the size, geometry and position of each particle, so a rigorous model should calculate the effect of each of millions or billions of particles, separately, over the whole system. Actually, these processes are characterized, modeled or even specified based on probabilistic descriptions of its features and behavior.

The fact is that catalysators and adsorbents are sold and used even though these vagueness and uncertainty problems exist. Usually there is no way to avoid working with processes that cannot be precisely described. A better approach is to develop suitable ways of modeling and handling their intrinsic vagueness and uncertainty, as in the absorber's example. Vagueness and uncertainty are not merely problems or imperfections. They also offer the opportunity to model complex systems, approximately, facilitating its handling. This approach has made it possible for today's software to deal with such complex features as computer-enhanced decision-making in safety-related situations. These applications are typically developed to imitate the human ability to reason, based on experience and common sense.

1.3 Objective and goals of the research work

The objective of the present research work is to develop a method for the online monitoring of complex processes, based on a convenient representation and management of vagueness and uncertainty in situation-based qualitative models. Vagueness and uncertainty must not be eliminated during the modeling; instead, the information hidden in them must be used to enrich the model. Goals associated with this objective are, consequently:

- Taking advantage of vague and uncertain information by the modeling of technical processes, as a way of increasing the available information, allowing a more precise representation of reality.
- Integrating this concept in existing Situation-based Qualitative Modeling and Analysis (SQMA) approach.
- Representing this vagueness and uncertainty without increasing the model complexity.

- Applying this enhanced SQMA model to online monitoring of complex technical systems.

In this formulation, vagueness and uncertainty management is considered first as a mean to enrich the descriptive power of qualitative models, and second as the solution for a problem of imperfect information [Pars96].

1.4 Organization of this document

Chapter 2 analyzes the theoretical framework of this research work. It begins with the description of the problem of monitoring complex industrial processes and continues with the exact definition of what this "representation and management of vagueness and uncertainty" means. Process modeling and monitoring techniques, which are based on the described representation methods, are also discussed.

Chapter 3 describes Situation-based Qualitative Modeling and Analysis [Lauf96],[Fröh97] in detail. SQMA's capability of handling vague and uncertain information will be analyzed, to support the formulation of a set of requirements and a procedural framework for the aimed solution concept.

A concept for representing vague information in SQMA models during the modeling of complex processes is introduced in *chapter 4*. The approach relies on the modeling of process variables with Rough Intervals [Rebo01a], which integrate elements of the Rough Set Theory in SQMA's interval-based representation of process variables. Integrating Rough Intervals in SQMA demands a new definition of key concepts such as intervals and situations, as well as a deep analysis of new capabilities, advantages and limitations.

The application of Rough Set Theory's principles by the modeling of process variables introduces uncertainty in the model. This uncertainty derives from the resolution in the representation framework. The reduction of this uncertainty is addressed in *chapter 5*. A stochastic model of process dynamics, based on Qualitative Stochastic Automata [Lunz95], is also proposed in this chapter, to represent and handle the uncertainty associated with nondeterminisms in complex dynamic systems.

A new approach for monitoring complex systems is developed in *chapter 6*. This monitoring approach uses Rough Intervals in an SQMA model to manage vague and uncertain information for analysis of the current situation. Transition forecasting is enhanced, based on the Qualitative Stochastic Automata technique and distributed probabilistic transition matrixes. The result is a remarkable improvement of model precision, without demeaning SQMA's capability of dealing with complex industrial processes.

The solution concept for situation-based qualitative monitoring of complex systems considering vague and uncertain information is evaluated in *chapter 7*. Based on its application on a Three-

Tank System, the main stages of the monitoring application are carefully studied. These stages are: modeling the technical system; observation and analysis of the current process state; and forecasting reachable situations. Solutions with conventional SQMA models and with the new concept based on Rough Intervals are compared to verify the fulfillment of the formulated requirements.

Conclusions and a summary of the most important considerations and results of this dissertation are presented in *chapter 8*. Additionally insights into possible future works in this research area are described.

2 State-of-the-art of the related technologies

The objective of this research work was formulated in section 1.3 as “the development of a method for the online monitoring of complex processes, based on a convenient representation and management of vagueness and uncertainty in situation-based qualitative models”. In this goal formulation, two dissimilar knowledge areas can be identified: the field of the problem (the process monitoring in complex industrial automation systems) and the field of the methods supporting its solution (representation and handling of vague and uncertain information). This chapter describes both knowledge areas.

Various vagueness and uncertainty modeling techniques from the industrial automation domain are subsequently introduced. Finally, these techniques are thoroughly evaluated from the point of view of the project goals. The conclusions of this chapter will support the development of the solution concept in the further chapters.

2.1 Complex industrial automation systems

Monitoring a running process is a key feature in any technical system. However, this function becomes essential in complex industrial processes, which are difficult to describe and interpret. Complexity can be divided into structural and functional complexity ([HDJ92] cited by [Fröh96]). Structurally complex systems consist of a large number of strongly coupled variables. Functional complexity prevails if the relationships between the system variables and parameters are partially known, are difficult to describe or demand difficult calculations. This functional complexity is strongly related to the information at hand by working with industrial processes. As a rule, the more complex a process, the lower the quantity and quality of available information. Process information is the raw material for modeling and further monitoring of a process. A useful monitoring application is difficult to develop if the information quantity and quality are inadequate for the intended application. However, discarding this information is not always possible.

This chapter is devoted to the precise definition of process monitoring in complex technical systems and to the analysis of the above-described problem. As a result, a set of requirements on the solution concept will be formulated.

2.1.1 Technical systems

The target application area for this research subject is automation of technical processes, specifically framed within the area of safety of industrial automation systems. A technical process is a process during which either material, energy or information is altered. This

alteration does imply the transition from an initial state to a final state [LaGö99a]. As shown in Figure 2.1, a technical process regards the change of state of (at least) a section of a technical system in terms of matter, energy or information. This process can imply the transformation, transportation or storage of elements in any of these dimensions.

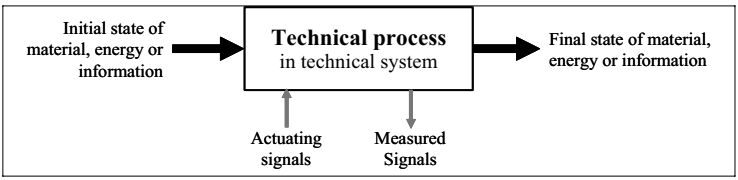

Figure 2.1: Flow of material, energy or information

The German standard about definitions in computing systems for processes [DIN66201] establishes a common reference framework through the illustration of the difference between a technical process and a process in the general sense:

> *"A process is the entirety of all interacting processes within a system that transforms and stores material, energy or information. A technical process is a process in which its physical parameters are recorded and influenced by technical means"*

The definition above introduces in the concept of technical process a new element: the gathering of information from the process (measured signals) and the influencing of physical parameters in the process by technical means (actuating signals).

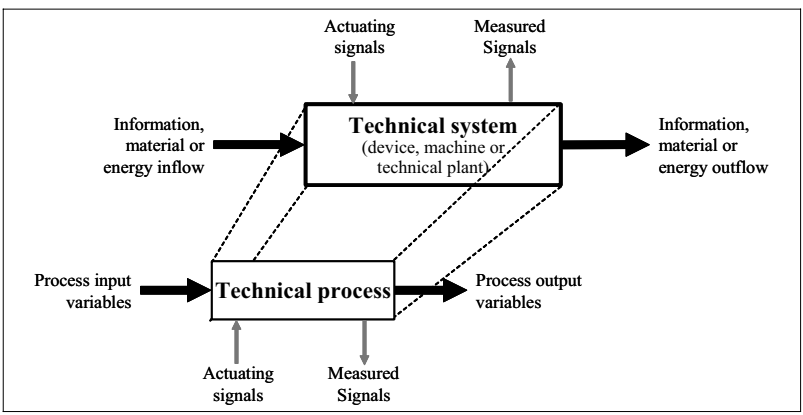

Figure 2.2: Technical process of a technical system

An important feature about technical processes is that they can vary from single products to highly complex systems such as power and production plants. Whatever the case, identifying a

set of smaller or simpler technical processes within a technical process is usually possible. This hierarchical structure makes possible the treatment of a system independently of the system's size. Technical system is therefore defined as the uppermost abstraction level (s. Figure 2.2), which summarizes the whole set of technical processes in an overall technical process. Depending on the chosen abstraction level, technical process and technical system can refer the same entity, for example, when a technical system comprises only one technical process.

Closing this section, dynamic processes are systems that exhibit a time-dependent behavior. Hence, all dimensions in a dynamic technical system must be considered as temporal functions. This includes actuating and measuring signals.

2.1.2 Safety-related automation applications

An *Industrial Automation System* consists of three parts: first the technical system, second the automation (control, safety, process information, etc.) hardware and software and third the humans who operate technical processes in the technical system [LaGö99a]. The technical processes define the boundary conditions for the automation of the system, establishing the requirements on the automation hardware and software.

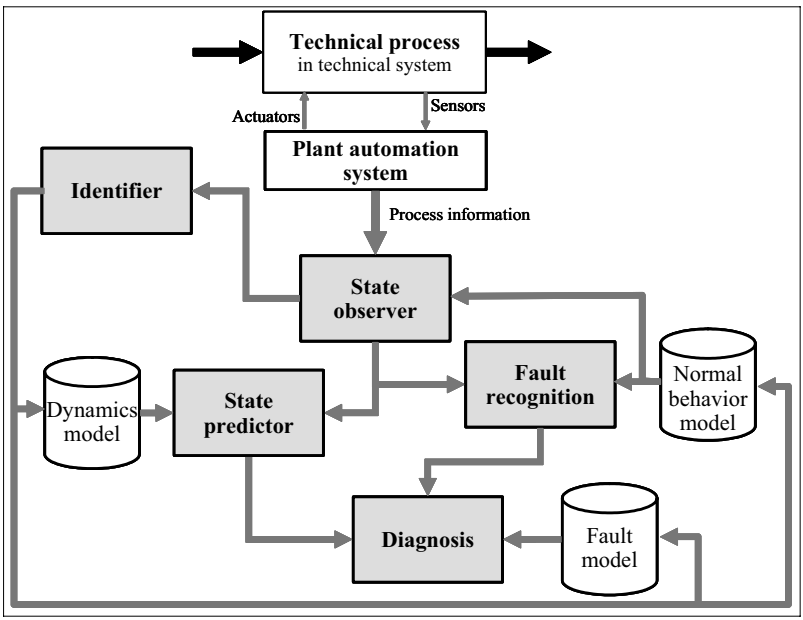

Figure 2.3: Typical configuration of safety-related automation applications

Safety-critical automation systems execute functions with safety responsibility. The failure of such systems could have dangerous effects. In complex industrial systems, like those handling hazardous chemicals or flammable materials, a number of safety-related applications must assure the reliability of the processes, preserve the installations and guarantee no harm to human beings or the environment. Figure 2.3 depicts a typical configuration of safety-related applications in industrial automation systems, signaling the flow of information between them.

The *state observer* is a safety automation application that can be found direct at the process, running online with it and following its evolution. A state observer acquires in real time the values of variables and system conditions to produce information manageable by other automation applications and by humans who are responsible for the process operation. The information gathered by the state observer can be directly used with a model of the correct process behavior to recognize irregular conditions in the process run. *Fault recognition,* may include primary level safety responses such as whether to shut off a plant or trigger fire extinguishers.

Only in exceptional cases does an opportune warning alarm, based on precise fault recognition, fulfill the safety requirements of a technical process. *Diagnosis* of the irregular process is also required, i.e. the isolation and characterization of the fault, the prediction of its consequences, the identification of possible causes and sometimes the description of a remediation procedure. Models of the faulty system achieve these tasks. The term online diagnosis is usually employed to remark the absence of diagnostic aspects such as fault isolation and fault recovery, which can function off-line, and therefore, does not require real-time capability. This definition of online diagnosis matches the above-described fault recognition.

The prediction of the states that the process can reach in order to identify possible future problems (or what is the same: current risks), is called state prediction. The *state predictor* follows the process behavior through the observer and, based on this information and a process dynamics model, determines which states can be reached from the current one. State prediction is rarely used alone as an automation application. Sometimes, a predictor is presented in the form of a single-variable simulator. Model simulation empowers safety analysis, for example, but also many other offline studies. On the other hand, with a correct process model, the online observation is not required. It is also possible to find diagnostic applications, which make use of simulators to evaluate fail recovery paths. The term “simulator” will be used along this project to signify these offline, single-variable, noncritical applications. The term “predictor” will be used instead in the context of risk identification applications.

All the described applications are based on the evaluation of the process information against a model, which, together with its corresponding modeling procedure, is the core of the corresponding safety application. However, these models are not always available. In many cases, an *identifier* must be use to provide it. An identifier uses data from an observer to produce

a system structure with a suitable set of parameters, which is capable of reproducing the observed behavior. It is not important how much the identified model reassembles the system, but the reproduction of its behavior must be guaranteed.

As it can be seen, it is not always easy to distinguish the borders between the different automation applications. For example, different levels of diagnosis were shown, from a simple fault recognition application (online diagnosis) to diagnostic applications, which are capable of postulating and evaluating several recovery paths.

2.1.3 Online process monitoring

Based on the above-presented classification of safety-related automation applications, the *online process monitoring* can be described (s. Figure 2.4). It encompasses the observation of the current system state, the early fault recognition and the determination of the current risks in the process (prediction of potential failures). For this application, a real time capability is as important as its dependability and precision. This online process monitoring application, or simply *"process monitoring"* as it will be further handled in this document, is the goal application of the present work.

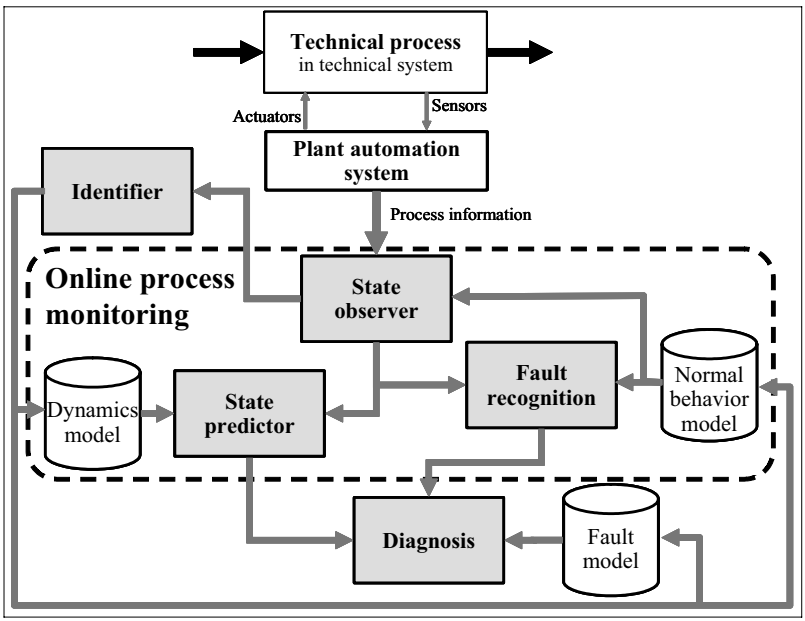

Figure 2.4: Typical configuration of a process monitoring application

The simplest approach for process monitoring does not consider the prediction of the process state and, therefore, does not require a model of the system dynamics. In this case the model of the system normal behavior consist of a series of limit values, which are used to determine if the observed system behavior is acceptable or not. Further upgrades of this signal oriented monitoring approach may use of a wrapping function for each monitored variable. This envelope defines operation zones that characterize the system behavior. This basic monitoring approach can be used to supervise signals acquired from the process, the rate of change of these signals or a combination of both types of variables [LaGo99].

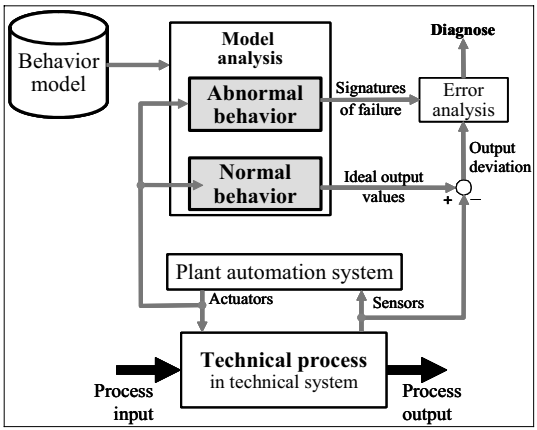

Figure 2.5: Output simulator approach for process monitoring

The described online process monitoring approach can be generalized by the output simulator approach. In this monitoring approach, the behavior of the technical process is directly compared with the process model behavior, as shown in Figure 2.5. The process model generates ideal output values from the measured process inputs. A faulty behavior is detected if the difference between the ideal and real output values exceeds a given tolerance. Then the error analysis is activated, based on a model of the abnormal process behavior, which definitively diagnoses the faulty behavior.

However, comparing process inputs and outputs may not be sufficient for an exact fault diagnosis. Accurate diagnostic applications require the monitoring of the process states, which are not always available for the automation system. In these cases, the state observer approach can be used. Based on the acquired process variables, state identification algorithms estimate variables and parameters that cannot be directly measured. These values are then analyzed against process state models, as shown in Figure 2.6, to allow the accurate fault detection and diagnosis.

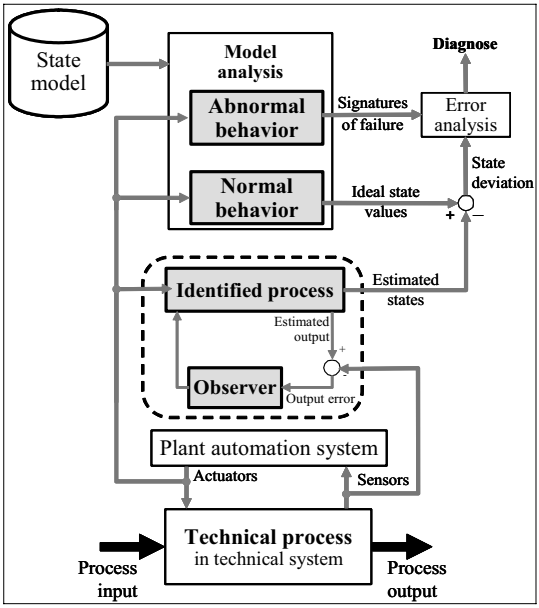

Figure 2.6: State observer approach for process monitoring

2.1.4 Imperfect information in industrial automation systems

The problem of representing and handling vague and uncertain information was briefly described in the previous chapter (s. 1.2), using the adsorption scenario[3]. Selecting a suitable technique for representing and managing vagueness and uncertainty is usually difficult, because making a clear separation between these and similar concepts is difficult. Prior to classification and separation, the data must be first delimitated from other kinds of information.

Imperfect information [Pars96] encompasses all the problems that can be present in the information gathered, stored or managed about a real system. Imperfect information in industrial automation systems results from functional complexity, which limits the quantity and quality of the information that can be handled. However, sometimes, imperfect information is expressly introduced in the process description as a way of dealing with its structural complexity. Handling imperfect information is a problem of representing this information in a model, operating with it, and reasoning upon it.

No definitive, unified, or universal theory or classification of imperfect information exists, even though the topic is discussed by both the database community and Artificial Intelligence

[3] The problem of representing and handling vague and uncertain information has been considered by philosophers, logicians and computer scientists, particularly those interested in AI and information management.

researchers [Pars96]. In this work, a classification schema is proposed which is consistent with characteristics and requirements of industrial automation systems, which are not always compatible with those of the database and AI worlds. This classification schema supports the delimitation vagueness and uncertainty in industrial automation systems. This classification schema consists of the following information imperfections:

2.1.4.1 Vagueness and imprecision

The formulation of *vagueness* is based on the notion that a vague concept cannot be delimited exactly; its truth or untruthfulness cannot be established beyond doubt. Vagueness is a general quality of verbal modes of expression, and thus of human perception and description of real phenomena. The boundary-line principle ([Freg03] cited by [Pawl92]) states that a concept must have a sharp boundary-line; consequently, a vague concept does not have a sharp boundary-line. This concept is then considered vague. Imprecision is handled as a case of vagueness applied to calculated or measured values, not to concepts.

Vagueness and imprecision in most technical systems are present in difficult-to-measure variables or parameters, and in the system rules, which formulation involves the human cognition and linguistic expression of the relations that take place in the system. Expressions such as *"the tank is almost full"* or *"the temperature of the reactor is between 50 and 60 degrees Celsius"* are frequent descriptions of a technical process state. Vagueness arises from the granularity in the knowledge about the universe or in the description of the space where the information is measured.

Imprecision can be expressed with intervals (value ranges) or sets of discrete values, whereas vague data is characterized by weighting values, which express the confidence on its information [Perm90]. A technique that handles vagueness and imprecision must have algorithms capable of dealing with these intervals and weighting values.

2.1.4.2 Uncertainty

Uncertainty in industrial automation systems is the lack of information about the actual state of the technical system and the world around it. Specifications in technical systems (Mean Time Between Failures, quality requirements, etc.) are usually expressed using uncertainty measures. Some systems, such as those including catalytic or multiphase processes, only admit an uncertain description of their behavior in experimental curves or tables.

Dealing with uncertainty in a technical system serves the management of uncertain situations, by ranking their possibility or likelihood among a group of situations, based on the available (incomplete) information. This way of dealing with imperfect information is called uncertain inference [Neap92]. Only one situation is true, while the others are false, but no information is provided about which one of them is the real solution. Uncertainty management in not about

deciding whether a situation is true or not, but about representing its tendency to be true in comparison with the other possibilities. Since there is no general agreement in the definition of uncertainty in the literature, the definition above presented, which is widely used in the industrial automation world, is adopted for this research study.

2.1.4.3 Ignorance

Ignorance is the total lack of knowledge about a particular situation in the technical process or about the value of one or more parameters. Ignorance particularly applies when these parameters or situations evade their approximate (vague) determination or the estimation of degrees or tendencies among the usually infinite possibilities (uncertain representation).

Ignorance is often found in industrial automation as Black-box (ignorance about the system structure and their parameters) and Gray-box (e.g. ignorance about system parameters) problems. Even though the modeling and effective handling of ignorance is considered an important task in information management and artificial intelligence (AI), operating in absence of appropriate information is avoided in industrial automation. This imposes the identification or estimation of the required parameters. In extreme cases, arbitrary values are used and the introduced error is either characterized or constrained by other means.

2.1.4.4 Inconsistency

Inconsistency is more related with the source of the information that with the information itself. Inconsistency is produced by the existence of conflicting information, such as in the verification of a nonzero water level in a tank and the observation of no flow through an uninhibited pipe at its bottom. Untrustworthy information can be handled as a case of inconsistency from the point of view of the single information elements.

In spite of being frequent in automation systems, inconsistency is not as well studied and characterized as vagueness, uncertainty and ignorance. Paradoxically, inconsistency is usually introduced in technical systems by redundant information sources such as double sensors, pairs of direct/indirect measurements and state observers and simulators, which may deliver contradictory information. Data reconciliation, based on the confidence in single information sources, helps eliminate inconsistency in conflicting or redundant information in large production complexes [Rebo95].

2.2 Vagueness and uncertainty representation in industrial automation

A variety of modeling techniques has been developed to manage vagueness and uncertainty in industrial automation applications. Managing vagueness and uncertainty in the modeling of

technical systems consist on selecting a suitable method for the representation and further handling of vague and uncertain information. This section provides insights about the fundamental and operational differences among the most important of these methods: probability, possibility, interval-based discretization, fuzzy sets and rough sets.

2.2.1 Probability

Probability represents the likelihood that an uncertain event or experiment will have a particular outcome. The probability of an outcome is represented by a number between zero and one (or 0% and 100%), with probability zero indicating certainty that an event will not occur and probability one indicating certainty that it will occur. The simplest problems are concerned with the probability of a previously specified "favorable" result of an event that has a finite number of equally likely outcomes. If an uncertain claim A has a true probability P(A), which can be approximated by the times it appears n(A) in a population of n tests. Problems may be more complicate than that, like when the various possible outcomes are not equally likely or when the events may have infinite outcomes.

Probabilities are frequently used for the description of uncertain dynamics in industrial automation because of its capability of representing and handling chained uncertain events and dynamic processes. This is the base of many simulation, prediction and filtering approaches. Furthermore, probabilities are also used for the description and analysis of the behavior and characteristics of complex processes.

Probability is a large and dense knowledge field, so a precise use of its basic terms is required. The following are a few fundamental concepts related to probability, which will be regularly used along this document.

- *Compound Probability* is the probability of all outcomes of a certain set occurring jointly. This must not be confused with Total Probability.
- *Total Probability* is the probability that at least one of a certain set of outcomes will occur.
- Two outcomes of an event are *Mutually Exclusive* if the probability of their joint occurrence is equal to zero; *i.e.* the occurrence of one precludes the occurrence of the other.
- Two outcomes are *Independent* if the probability of their joint occurrence is given as the product of the probability of their separate occurrences, *i.e.* occurrence or nonoccurrence of one does not alter the probability that the other will or will not occur.

These concepts may seem irrelevant outside the context of a probabilistic problem, but most of them are already part of the human way of understanding, conditioning or comparing events.

Another important definition is *Conditional Probability*, which is the probability of an outcome when it is known that some other outcome has occurred or will occur. The conditional probability P(A|B) is the true probability of the event A given the verification of the event B. Bayes' theorem enables handling simultaneous and conditioned events through probabilities. According to this theorem, P(A|B) satisfies the relationships in (2.1) and can be determined, therefore, using (2.2).

$$P(A|B) = P(A \wedge B)/P(B) \quad \rightarrow \quad P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A) \tag{2.1}$$

$$P(A|B) = P(B|A)P(A)/P(B) \tag{2.2}$$

With the introduction of state diagrams, Markov chains provided a new way of working with uncertainty to model dynamic systems. The information about the state transitions is enhanced by adding to each particular transition its individual probability. Now, standing in any particular state in the diagram, each permissible next states is signaled by the corresponding transition probability, *i.e.* the history of the transitions in the system is unnecessary. Processes that exhibit this behavior are called Markov Processes.

Stochastic models are another probabilistic way of handling uncertain information. Assumptions are the probabilistic independence of process variables and parameters and that the deviations in its behavior follow a certain function of probability distribution. These conditions in a stochastic system apply between two different variables, as well as between different instances (corresponding to different times) of the same variable, *i.e.* a process is stochastic if it deals with independent random variables that also in time remain independent and random.

2.2.2 Interval based qualitative representation

In mathematics, a container that totally includes or excludes any given element is called a *classic set*. According to the Classic Set Theory, an object (element) can either be part of a given set or be excluded by it; i.e. each element of the *universe of discourse* is either in the set or out of it. This set can represent any idea or concept, as in the example in Figure 2.7, where the set C should contain all the big objects of this universe of six elements.

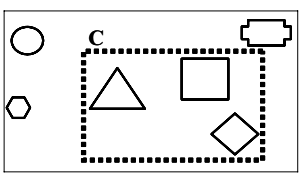

Figure 2.7: Classic set C of big objects

Even though in industrial automation problems it is not usual to find crisp borders, classic sets are usually employed to approximate them, as it is shown in the figure. It can be considered, for instance, that the object in the upper-right corner is also big. Alternatively, it is perhaps not big enough. What is about the diamond? Classic set only resolves approximately this problem by adopting a trade-off position, such as that of the set C.

This classic set notion can be adapted to continuous variables, instead of objects, just by delimiting value ranges in the dimension where these variables are defined. Value ranges are frequently used in industrial automation for the depiction of vague concepts (like the case of "big" in the previous example) and imprecise values. They are usually associated with specific process characteristics or behaviors allowing the qualitative modeling of the process. Value ranges can be used to represent discrete and logic values as well, which offers a common framework for the qualitative modeling of real process variables and parameters.

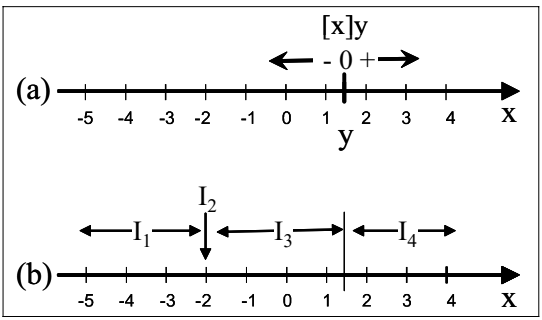

Figure 2.8: Prefix and interval representation

A possible form of value range notation is the prefix sign representation relative to a reference point [dKBr84]. In this notation system, a qualitative value is derived from the prefix of the quantitative value in the reference system determined by the given referential point. Figure 2.8a shows this kind of qualitative representation. [x]y is a qualitative variable with the values {"+", "0", "–"}; where x is a variable and y is a parameter that serves as reference point of the qualitative variable [x]y. For x < y, x = y and x > y, becomes the qualitative variable [x]y the qualitative values "–", "0" and "+" respectively. It is valid therefore that [x]0 = Sign(x). Based on this principle, many reference points may be defined at will, in order to generalize the interval bordering. The result of the superposition of these qualitative variables is an interval representation.

The interval [a b] (with a < b) describes the area between the reference points a and b (for example I3 = [-2 1,4] in Figure 2.8b). Following this notation, intervals of the form [a a] describe the reference point exactly like the corresponding real number a (for example I2 = [-2 -2] = -2 in Figure 2.8b). In any case, the borders of these intervals are crisp (from here

the name of crisp interval); there is no transition or common point between adjacent intervals. Finally, there are also unbounded intervals of the form (-∞ a] and [b +∞). They describe the area from -∞ to the smallest point a (I1 = (-∞ -2] in Figure 2.8b) and that defined between the biggest reference point b and +∞ (I4 = [1,4 ∞) in Figure 2.8b). The entire dimension must not always be portrayed; it often suffices representing excerpts of it.

2.2.3 Fuzzy sets and possibility

The term "classic" in classic sets obeys to the need of making a clear differentiation with a more modern notion of sets: the *fuzzy sets*. A fuzzy set is a set without a clearly limited boundary, capable of partially containing elements and capable of flowing into the boundaries of other sets. A fuzzy subset **F** of a set **U** can be defined from the basis of the classic subset theory as a set of ordered pairs, each with a first element that is also an element of **U**, and a second element which represent the *degree of membership*. A value zero is used to represent complete nonmembership, the value one is used to represent complete membership, and values in the interval (0 1) are used to represent intermediate membership degrees.

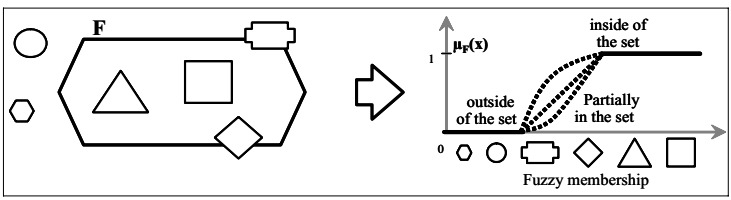

Figure 2.9: Fuzzy subset F of big objects

In the six-objects-universe example, each one of the objects is represented in the fuzzy subset **F** (Figure 2.9, left) with its corresponding membership value, which is determined through the membership function shown in Figure 2.9 (right). Even though the terms "fuzzy subset" and "Fuzzy Subset Theory" are mathematically correct, "fuzzy set" and "Fuzzy Set Theory" are employed instead in the most of the specialized literature. To facilitate the understanding and comparison of this principle with the others studied in this document, the expressions "fuzzy set" and "Fuzzy Set Theory" will be further used, unless a precise mathematical definition is intended, as is the case of the following paragraph. The set **U** is defined as the universe of discourse for the fuzzy subset **F**. The mapping in the interval [0 1] would be then the membership function of **F**. Based on this definition, the degree to which the statement: *"x is in* **F***" is* true, is determined by finding the ordered pair whose first element is x defined in **U.** The degree of truth of the statement is the second element of this ordered pair defined in **F**, and is normally denoted as $\mu_F(x)$. In the fuzzy subset theory, there are no further restrictions for the membership functions. A membership function can be as simple as unitary steps with a gracious

slope in the change of levels, but this is not the rule; they can be much more complicate than that, like Gaussian functions or limited conic curves.

The same strong relationship between the Classic Set Theory and Boolean logic exists between the Fuzzy Set Theory and fuzzy logic. Fuzzy logic is the logic of what is intrinsically vague, not a vague logic. Memberships are treated as *true degrees* or *grades of truth*, so a membership function becomes a truth function. Logic operations are then defined on the true degrees of the fuzzy variables and implemented via a TRUTH operand. The more important fuzzy logic operations are AND, OR and NOT with basically the same meaning as in classic logic.

Zadeh also introduced the concept of possibility [Zade78]. It collects related theories and concepts that emerge from the notion of fuzzy sets, but beyond them and similar to probabilities, possibility represents uncertainty, not vagueness. The goal of the *Possibility Theory* is to manage the feasibility of that a particular event comes into being. Mathematically, possibility is defined as the fuzzy degree of truth of the statement *"x is* **A***"*, where **A** is a fuzzy set. So the fuzzy linguistic expression *"x is in certain degree* **A***"* is translated into the expression: *"x is possibly* **A***"*, and from here is the possibility value numerically defined to be equal to the correspondent μ for the original fuzzy statement. *Possibility* is a fuzzily elastic constraint for the values that can be assigned to a variable. From here on, the possibility theory follows its path disassociated from fuzzy sets and fuzzy logic, and evolves independently as theory.

In his paper, Zadeh defined the measure of possibility as a scalar index that evaluates the consistency of a fuzzy proposition with respect to a state of knowledge expressed by means of a fuzzy restriction. The notion of fuzzy restriction corresponds to a radical change in the semantics of the membership function. A fuzzy restriction is a fuzzy set of possible values, and its membership function is thus called a possibility distribution. Trying to formalize this concept Dupois and Prade [DuPr94] can be quoted:

> *"Let x be a variable taking its values in a set U. A possibility distribution Π_x attached to x describes a state of knowledge about the value of x. This value, although unknown, is supposed to be unique. Π_x is a mapping from U to the unit interval, such that Π_x (u) = 1 for at least one value u"*

To complete the definition above, $\Pi_x(u) = 0$ determines the impossibility of x = u, whereas $\Pi_x(u) = 1$ indicates that x = u is completely allowed. Possibility valuations play a role similar to probabilities in logic. Instead of considering a probability distribution on a set of possible interpretations, a possibility distribution, which expresses a sort of order of preference among its elements, can be considered. The result is possibilistic logic.

2.2.4 Rough sets

The Rough Set Theory was proposed by Pawlak in 1982 [Pawl82] as a method for the joint management of vagueness and uncertainty. It is based on the Classic Set Theory, but is inspired by Zadeh's fuzzy sets. According to Worboys [Worb98]:

> *"The starting point of the theory of rough sets is that entities can only be perceived by making observations about them, and that the observations provide information at differing degrees of precision and accuracy. Any particular observation is made at some granularity or resolution, in which collections of elements are indiscernible from each other. The higher the resolution (or lower the granularity), the better we discern differences between elements."*

Reasoning in the Rough Set Theory is a matter of classification, not of degrees of truth or likelihood. A vague concept, expressed in the form of a vague classification criterion, is always susceptible of being decomposed in two precise concepts (here, two classification criteria) that can be further handled independently. For a given vague concept it can be always formulated a lower-approximation containing all the elements that without doubt are included in the vague concept, and an upper-approximation containing those elements that cannot be excluded from the vague concept.

In the Rough Set Theory, it is assumed that some information (knowledge) about the elements of the universe of discourse is available. This information enables the element classification according to predefined criteria. Therefore, if the same information can be associated with several elements of this universe, these elements are *indiscernible*; i.e. they cannot be distinguished from each other based on the handled knowledge. This *indiscernibility*, in fact, can be broken if enough discriminating information is added to enrich the knowledge about the process, but this is not always feasible. Equations (2.3) and (2.4) define the lower and the upper approximation of a set X as a function of this indiscernibility:

$$\text{Lower approximation: } B_*(X) = \{z \in U : B(z) \subseteq X\}, \tag{2.3}$$

$$\text{Upper approximation: } B^*(X) = \{z \in U : B(z) \cap X \neq 0\}, \tag{2.4}$$

B(z) represents in these definitions the set of all the elements of U, which are indiscernible from z based on the information in B. Whereas similarity is assumed to be a reflexive and symmetric equality relation, indiscernibility is in addition transitive, *i.e.* similarity is a tolerance relation and indiscernibility an equivalence relation.

The partition of the problem space in an upper approximation $B^*(X)$, a lower approximation $B_*(X)$ and a boundary region $BN_B(X) = B^*(X) - B_*(X)$, resolves the vagueness. If the boundary region of X is an empty set, the concept represented by the set X is precise and the rough set converges to a classic set; otherwise it is named rough (vague). Thus rough sets can be viewed

as a mathematical model of vague concepts, and can be characterized numerically by the following coefficient:

$$\alpha_B(X) = \frac{|B_*(X)|}{|B^*(X)|}, \qquad \text{resulting } 0 \le \alpha_B(X) \le 1 \tag{2.5}$$

If $\alpha_B(X) = 1$ then X is precise, and if $\alpha_B(X) < 1$, X is rough because its elements are indiscernible respect to B. This definition of vagueness is consistent with the boundary-line principle introduced in 2.1.4.1.

The above-described classification-based concept is illustrated in Figure 2.10. The left side shows the elements $z \in U = \{1,2,3,4,5,6\}$ to be classified according to their membership to a vague concept X (e.g. representing the feature "big") and the available information B. The right side of the figure represents the vague concept X following rough set's principles. The inner set B_* (lower approximation) comprises elements {1,2} that are unequivocally included in X. The outer set consists of elements {5,6} that definitively are not part of X (e.g. they are small objects). The classification of {3,4} is uncertain, e.g. it is not possible to decide whether they are big or small. Elements {1,2,3,4} define together the upper approximation of X.

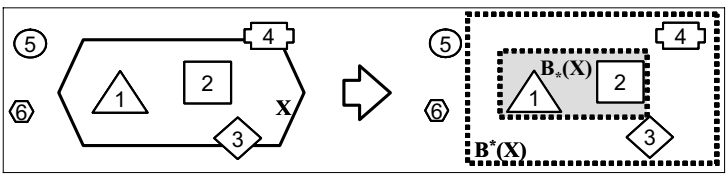

Figure 2.10: Vague information with rough sets

Now, if a vague concept is a case of boundary-line, uncertainty is related to the membership of the different elements of the universe to the encircled set (i.e. the concept). This uncertainty is concentrated in the boundary region ($BN_B(X)$) of the concept X based on the information B. The boundary region confines the uncertainty in the concept. Here no solid conclusion can be reached about the membership or not of an element to the considered concept according to the available information. An uncertainty-based membership function, related to the rough set concept (rough membership function), is therefore necessary to attack uncertainty problem. It can be easily defined employing the indiscernibility as shown in (2.6):

$$\rho_X^B(z) = \frac{|X \cap B(z)|}{|B(z)|} \tag{2.6}$$

The function | ... | represents the size of the set, i.e. the number of elements in it. The Symbol ρ is used to represent rough membership instead of the generally used μ, with the intention of marking an explicit differentiation with the fuzzy membership, where the symbol μ is also used.

Rough membership can be expressed in percentage (%) to remark its probabilistic nature; nonetheless, its expression as a value between zero and one is also frequent.

Table 2.1 shows the more important properties of rough membership functions (s. [Pawl94]). As it can be observed, these include the definition of the fuzzy membership function as a special case. Beyond these properties, no explicit definition is made about the form of a rough membership function, mainly because this theory was formulated over sets, not on a continuous numeric space like was the case of the fuzzy membership functions.

Table 2.1: Properties of rough membership

$\rho^B_X(z) = 1$	if $z \in B_*(X)$
$\rho^B_X(z) = 0$	if $z \notin B^*(X)$
$0 < \rho^B_X(z) < 1$	if $z \in BN_B(X)$
$\rho^B_{U-X}(z) = 1 - \rho^B_X(z)$	for any $z \in U$
$\rho^B_{X \cup Y}(z) \geq \max(\rho^B_X(z), \rho^B_Y(z))$	for any $z \in U$
$\rho^B_{X \cap Y}(z) \leq \min(\rho^B_X(z), \rho^B_Y(z))$	for any $z \in U$

There is a relation of precedence and consequence, an order among elements, in a numeric space. This relation of precedence and consequence governs the interval definition in real variables. The application of this notion to the properties of a rough membership function would determine the necessity of using monotonic growing and decreasing functions in the boundary region, between the limit values of z where $\rho^B_X(z) = 0$ and $\rho^B_X(z) = 1$. It means that, the nearer a number (an element) to the interval, the more probably this number is part of the represented concept.

2.2.5 Representing vagueness and uncertainty

In this chapter, several methods for the depiction of vagueness and uncertainty were studied. In general, it could be observed the suitability of sets and intervals for modeling vagueness. Classic, fuzzy and rough set approaches deal with the representation of vagueness and imprecision in its own way. Figure 2.11 compares the application of these three kinds of sets for the description of the vague concept “big object”. Note that for the classic set (Figure 2.11a) a clear delimiting line was defined to approximate the classification. This crisp border may not represent exactly the size relation among the different elements of the universe of discourse admitting further reinterpretations, but at least represents the problem in a very simple way, particularly when compared with fuzzy sets (Figure 2.11b) and rough sets (Figure 2.11c), which are clearly more precise.

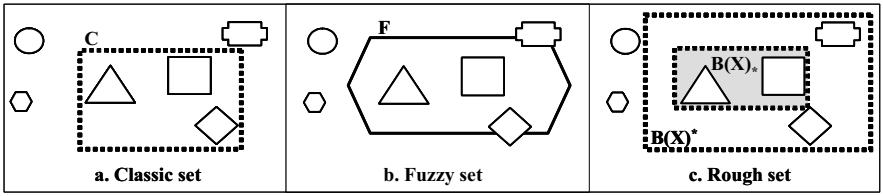

Figure 2.11: Various set representations of vagueness

Fuzzy sets could be used for representing uncertain events as well [Zade83], but this approach is not recommendable for uncertainty management [Bezd93]. Fuzzy sets serve better the treatment of vagueness than uncertainty. Uncertainty is better and easily represented through probabilities and possibilities. Rough sets, on the contrary, can be seen as a solution serving simultaneously vagueness and uncertainty. There is a strict relationship between vagueness and uncertainty according to the Rough Set Theory. To some elements of the universe of discourse, the same information may be associated. This causes that these elements could be considered part of one or another vague concept. This uncertainty is represented in the Rough Set Theory separately with probabilities. Thus, the rough set approach establishes a clear connection between the two concepts: vagueness is related to sets (concepts), whereas uncertainty is related to the elements of the sets, i.e. vagueness is defined in terms of uncertainty.

This definition of vagueness is compatible with the one formulated in 2.1.4.1. The original formulation of vagueness in this research project was based in the notion that a vague concept cannot be delimited exactly; its truth or untruthfulness cannot be established beyond doubt. Vagueness was defined a general quality of verbal modes of expression, and thus of human perception and description of real phenomena. The concept that serves as fundament for the Rough Set Theory is compliant with the one enunciated. The difficulties by the exact definition or evaluation (truth or not truth) of the concepts in the rough set concept of vagueness would result from the absence of a boundary line separating them.

Both definitions of uncertainty are also compatible. Representing the uncertainty in a technical system was defined as serving the management of uncertain situations, i.e. the uncertain inference as defined in [Neap92]. This coincides with the intention of the uncertainty in rough sets of expressing how likely a given element can be found to be a member of the analyzed concept or to exhibit the studied feature. This can be analyzed from the point of view of incomplete information as well, since a more precise attribute set (*i.e.* more information) could easily break the indiscernibility that causes the uncertainty in a rough set.

2.3 Vagueness and uncertainty modeling for process monitoring

This section describes a representative set of approaches for the construction of approximate models of the process behavior, which are used in process monitoring applications. Figure 2.12 shows the three first levels of a classification scheme for process models in industrial automation proposed in [LaGö99b]. Based on this classification scheme, techniques capable of handling vague and uncertain information (dark boxes at the bottom) are analyzed in detail, comparing their theoretical foundations and applications. This section also describes hybrid approaches.

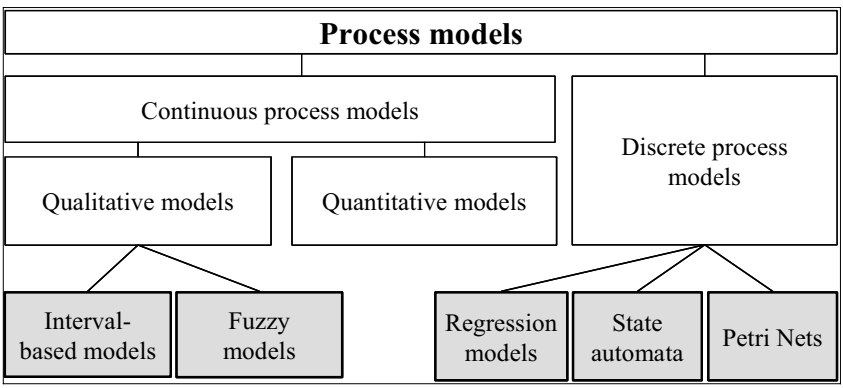

Figure 2.12: Classification of process models

It is not the intention of this section to make judgments about the techniques, only describes their characteristics and the identification of relations among them. A detailed analysis of the state of the art may be found in the next section.

2.3.1 Stochastic regression models

Stochastic regression models [Schn98] are frequently used in the estimation of a variable with incomplete information. Assumptions in this analysis are that the variables are probabilistic independent and that deviations in the variable's behavior follow a certain function of probability distribution. These conditions in a stochastic system apply between two different variables, between different instances (corresponding to different times) of the same variable, *i.e.* a process is stochastic if it deals with independent random variables that also remain independent and random in time.

Inside the stochastic models can be distinguished the auto-regressive, moving average models with external inputs (ARMAX), which are most frequently used in engineering and scientific applications. This general model includes among others the ARX, ARMA and multiple linear regression models. Its discrete usual form is shown in equation (2.7), where Y, X, U and E are vectors that contain respectively outputs, states, inputs and an error component for its nondeterministic nature. In ARMAX models, the recent history of each variable must also be included. For this reason, coefficients **A**, **B**, **C** and **D** are actually matrixes of polynomial equations in the *z*-plane or *w*-plane [Kuo80].

$$\mathbf{A}Y + \mathbf{B}X + \mathbf{C}U + \mathbf{D}E = 0 \tag{2.7}$$

An important concept in ARMAX models is the *brown error* (for the vector E). According to this principle, any real error can be represented as the superposition of several temporal instances of *white noise*, where white noise is an ideal error, which is an independent random variable, independent even from itself in time.

Besides the stochastic behavior of the system, two assumptions are made with ARMAX models:

- Any present output depends upon the current and previous values of input and (supposed brown) errors, as well as previous values of the output variable itself.
- Any estimation can be as precise as a white error will allow, which is by nature unpredictable.

Many principles have been devised for the determination of matrixes **A**, **B**, **C** and **D** in an ARMAX model, which are the coefficient of the model and have meaning not only according to the variable, but also according to time. Most of these techniques follow the same principle of minimizing the square-error (as in linear multiple regression [Schn98]) or the variance (such as the minimal variance method [GoSi84]). These techniques demand intensive probabilistic and statistical calculations, especially if they are used together with hypothesis verification theorems.

The nondeterminism in ARMAX models is introduced by means of the considered error factor. Without this factor, ARMAX models are equally suitable to be employed in deterministic systems. With X, Y and U as state, output and input vectors, respectively and **F** and **G** as mapping functions between the involved sets, nondeterministic stochastic systems can be generalized as:

$$X(k+1) \in \mathbf{F}(X(k),U(k)) \tag{2.8}$$

$$Y(k) = \mathbf{G}(X(k),U(k)) \tag{2.9}$$

F is not a specific mapping function but a relation between the sets. The symbol $\in$ in this case embodies the uncertainty that in equation (2.7) was represented by the error factor. In this case,

the nature of **F** is not the same as that of **G**, however in order to simplify the generalization of this notion, **F** and **G** will be assumed as equivalent. These relationships represent dynamic systems as discrete processes, using equations of differences instead of the usual differential equations. In this representation, the framework is not the time but the specific sequence of values that are important. These sequences of input and output values can be (but do not have to be) indexed and limited, i.e. these values could be restricted to sets of individual elements, where a connection with qualitative modeling methods can be developed.

ARMAX models are used most often in advanced control applications, in constructing predictors and observers for otherwise invisible system states. ARMAX models with incomplete information. This structure allows the modeling of linear processes corrupted by an additive noise, but its main problem is that it is a recurrent model, which makes the estimation of these parameters difficult.

2.3.2 Stochastic Petri nets

Since their introduction by C. Petri in 1960, Petri nets have been widely used in modeling and analysis of computer systems in application areas such as automatic control, communication networks, manufacturing and distributed computing. Petri nets are used to model dynamic behavior of discrete systems. The basic elements of Petri net models are “places”, “transitions” (represented by bars or boxes), “directed arcs” and “tokens”.

The Petri net in Figure 2.13 illustrates how these elements connect and interplay with each other. If a directed arc connects a place to a transition, the place is described as the input place to the transition so that P1 represents the input place to transition T1. Similarly, if there is a directed arc connecting a transition to a place, the place is an output place for that transition. A single place can be connected to a single transition with more than one weighted arc. Conforming to these basic rules, multiple places and transitions can be connected to form a complex net.

Petri net models consist of two parts: (1) the net structure (Figure 2.13, top) represents the static part of the system, and (2) a marking (Figure 2.13, a to d) represents system’s current state. Marking is determined by the token distribution among the places of a Petri net. A place is marked when one or more tokens reside in it, otherwise the place is unmarked. The number of tokens at a place represents the local state of the place, thus marking of the net represents the global state of the system. Figure 2.13a captures the initial marking of the net, which represents the initial system state. The flow of tokens and the firing of transitions model the dynamic system behavior.

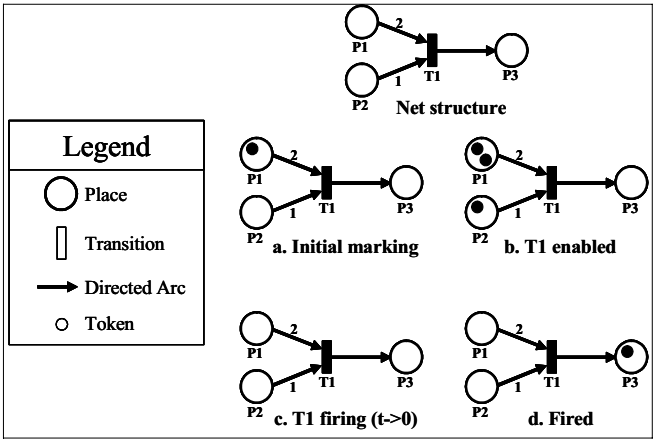

Figure 2.13: Petri net elements and firing sequence (a to d)

Transition firing means that tokens in the input places have apparently moved to the output places. Transition firing involves the following steps:

(1) In the initial state there is only one token in P1. Transition T1 is disabled (Figure 2.13a).

(2) A transition is enabled if each input place has at least as many tokens as the weight of the arc connecting them (Figure 2.13b).

(3) An enabled transition is fired by removing from each input place the number of tokens equal to the weight of the arc connecting them (Figure 2.13c).

(4) When the transition is fired, tokens are added to the output places connected to the transition. The number of tokens added to each output place is equal to the weight of the arc joining them (Figure 2.13d).

The above-described mechanism is called “rule firing” or informally, “token game”. While this token game governs the dynamic behavior of Petri net models, the meaning of this process is determined by the net interpretation. When applied to different domains, elements may represent different things. Normally, in the manufacturing context, a place represents the state of a resource or its availability. A token in a place would then indicate the resource is in that state or that it is available according the case. Events and activities are then modeled by transitions. Firing a transition triggers an action or event, or indicates the start or termination of an activity. Assuming infinite input and output buffer size and parts supply, the net is never starved.

To study the performance and dependability of a dynamic system it is necessary to include time and probability in a model. This is usually done by associating delays, probabilities, or a combination of both, with transitions in the net. For step (b), enabled transitions are never forced

to be fired. In practical modeling, transitions can be related to external conditions that determine whether they should be fired when enabled. In Petri net models with no temporal feature, firing occurs instantly (Figure 2.13c).

Stochastic Petri Nets (SPN) [Natk80] were defined by associating an exponentially distributed firing time with each transition. An SPN can be analyzed by considering all possible markings and solving the resulting reachability graph as a Markov chain. Generalized Stochastic Petri Nets (GSPN) [MBC84] allow immediate and exponentially distributed timed transition firing; immediate transitions are drawn as thin bars, while timed transitions are thicker. GSPN are solved as Markov chains as well. A third type, Extended Stochastic Petri Nets [BTGN84], allows transition times to be generally distributed. This last Petri net can be solved in some cases as Markov chains or semi-Markov processes; otherwise, they must be simulated [CiBe02].

Petri nets are characterized by their ability to handle operation sequences, concurrency, conflicts and mutual exclusion in dynamic systems. These features make them a promising tool for describing and analyzing concurrent and real-time systems. In addition to their modeling power, Petri nets are both a graphical and a mathematical tool. As a graphical tool, they provide a visual medium for the modeler to describe a complex system and present it to users. As a mathematical tool, a Petri net model can be represented by linear algebraic equations, which opens a possibility for formal analysis of the model [ZuZh94].

2.3.3 Nondeterministic finite automata

An *automaton* or *finite state machine* consists of a collection of states interconnected by transitions that are activated by events. A system moves from its current state to the next via a transition. Transitions are triggered by events, which are typically received asynchronously by the state machine as inputs. The next system's state is a function of the current one and an event. Traditional state machines require that the system resides fully in only one of its states at any given time.

Automata are generally associated with the graphic depiction of states of a given system and its corresponding transitions. This however is not a rule. In general, an automaton cans result of the re-expression in discrete form, based on differences, of system state equations; in particular when these states are considered as a group of discrete elements. Beyond this, it is irrelevant if these states are numeric, qualitative or hybrid. In general, an automaton follows the relation in (2.10), where X(k) is the vector of modeled states and V(k) the system inputs (events) for the same instant k. In a recursive relation like this, in order to describe the system trajectory, knowing the initial value X(0) is mandatory. It is assumed that there is no input before k = 0.

$$X(k+1) = f(X(k),V(k)) \tag{2.10}$$

Deterministic finite automata are a special kind of finite automata characterized by two features: there is one specific starting state and from every state there is exactly one transition for each possible input symbol. Systems described by such automata are completely predictable. Given an input string, the sequence of states that are visited is completely determined. Nondeterministic finite automata, on the contrary, do not necessarily go to a unique next state. All possible next states in this kind of automata are called reachable states, in order to emphasize that a given state of this set may or may not follow the current one. A Nondeterministic Finite Automaton may or may not change its state when reading an input symbol, but if it does, it should select one of its reachable states.

2.3.3.1 Qualitative and stochastic automata

Qualitative models can be developed if sets of qualities are defined in a finite automaton as inputs and states of the system. However, since qualitative models are nondeterministic [Lunz95], qualitative automata are also nondeterministic. All possible state changes are modeled to ensure the new state will be one of those marked as reachable, even though it cannot be said which one of them.

Lunze also presents a sufficient transitional representation (automaton) for the dynamics of a qualitative automaton:

$$S(k+1) = \mathbf{T}*S(k) \tag{2.11}$$

This equation follows the general relationship expressed in (2.10). S(k) is a vector representing admissible states for the instant k and **T** the adjacency matrix that summarizes all possible state transitions. All elements of **T** are either one for possible transitions or zero for those that are not allowed. Based on this matrix, and knowing the state at $t = 0$, a complete state sequence (and thus the current state at any time) can be calculated. This starting point is usually known; that is why the vector S(0) normally contains only one element with a value of one. As expected, uncertainty increases with the index k, and with it the number of ones in S(0) and **T**.

Another way of representing an automaton is considering the transitions independently of triggering events by handling its occurrence probability. For example, if a system goes from state Z_i into state Z_k only after triggered by events V or W, and assuming these events are unknown or unmeasured but with occurrence probability P(V) and P(W) respectively, the probability of the transition ($Z_i \rightarrow Z_k$) will be the combined probability P(V) + P(W). In general, a matrix can be built with these transition probabilities as elements. This transition matrix, with the probabilities of each particular transition instead just one for possible or zero for impossible, is the core of a stochastic automaton. Stochastic automata are also nondeterministic, since are unable to deliver a unique value but a set of candidates that includes the real solution. Hence stochastic automata, like qualitative automata, follow the relation in (2.11), but with a probabilistic transition matrix **T**. Nonetheless the delivered information exceeds that of others

nondeterministic automata (including qualitative automata), since the provided candidates are ordered and weighted in the set according to their occurrence probability. Both definitions, qualitative automata and stochastic automata, are not mutually exclusive. Moreover, they can be combined in stochastic qualitative automata by handling qualitative states with a probabilistic transition matrix.

2.3.3.2 Fuzzy automata

A fuzzy automaton [KlYu95], also called a fuzzy state machine is an extension of a traditional state machine where the system can partially reside in one or more states. A fuzzy automaton is the result of assigning membership degrees as weights to the states of a machine, weights on transitions between states, and then a composition rule to calculate new degrees for future states. States themselves are crisp as in deterministic automata, but the system can exist in a given state to varying degrees. The degree of being of a state is generally represented by the symbol ß to differentiate it from membership function defined for system variables, which represented by μ. Fuzzy Automaton is different from other automata because the system can partially reside in a state (ß < 1) and the system can simultaneously reside in more than one state (nondeterminism) as long as system "being" is conserved (Σß = 1).

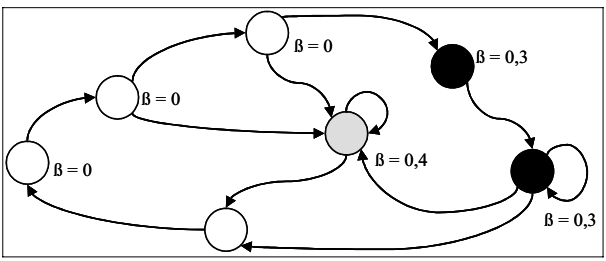

Figure 2.14: Fuzzy dynamics in state space.

Figure 2.14 shows two partially activated danger states (black circles) and at the same time the partial activation of a warning state (gray circle in the center of the diagram), each one with a different activation degree (ß), each one suggesting independent evolution paths. Accordingly, a variable inside a system component may exhibit at the same time a "Normal" and a "Too_high" value, although not necessarily with the same strength. If this is the case, the diagram can show the system state approximately, relative to the defined states.

Brubaker [Bru93] makes an ample, but at the same time sharp, analysis of the events in a Fuzzy Automaton. He describes two types of events, Trigger Events with an implicit TRUTH-value of 1, and Conditional Events, which TRUTH-value is defined for a fuzzy logic expression or a fuzzy rule. Triggers are crisp and instantaneous, for them a "partial occurrence" is not allowed. When used in a fuzzy automaton, they cause complete transitions in the system. The entire

degree of being in the current state is transferred to the destination-state. In this sense, fuzzy automaton behaves like a traditional deterministic automaton.

Conditional events, on the other hand, can be either crisp or fuzzy. The event's degree of fuzziness is the degree to which the associated condition is true (membership), and is represented by the symbol μ. Fuzzy events can occur over several data sample intervals, resulting in smooth transitions. Stronger events cause faster inter-state transitions, while weaker events cause slower inter-state transitions. For each executed transition, the degree of being of the source-state is reduced by the product of the source state's degree of being and the fuzziness of the event. Because the overall "being" of the system is conserved, the degree of being of the destination-state is the sum of its current value with this same product. This feature allows the modeling of sequences with Fuzzy Automata.

2.3.4 Interval-based qualitative models for process monitoring

Qualitative models are simplified models of real systems, based on "linguistic" descriptions of physical situations and relationships. *Qualitative modeling* is the first of two processes normally included in qualitative reasoning. The second process is *qualitative simulation*, which uses previously determined qualitative models to predict possible behaviors of the modeled system. In order to achieve this, a qualitative simulator generates an expected output from the real system's inputs and compares the simulated and actual behavior. This procedure can be used to monitor the correct functioning of a technical process.

The main characteristic of *qualitative reasoning* methods is that they aim to capture fundamental system behavior in a computer model, while suppressing much of the detail. In these models, vague and uncertain expressions simplify the approach for online monitoring and diagnosis of complex systems. Qualitative Reasoning is an area of AI, which creates representations for continuous aspects, such as space and time, to support reasoning with very little information. These methods try to approximate human thinking and reasoning. Qualitative reasoning is a promising tool for research activities in monitoring and diagnosis of complex dynamic systems. Important representatives of qualitative reasoning are the methods introduced by de Kleer and Brown [dKBr84], Kuipers [Kuip94] and Forbus [Forb90]:

- De Kleer's approach employs a special *physics based on Confluences* for qualitative system modeling. Confluences are the qualitative depiction of differential equations, where only the symbols [-], [0], [+] and [?] (for negative, zero, positive and undetermined) are accepted as arguments. Confluences are managed with sign arithmetic represented in a tabular form. This arithmetic includes the derivative function $\delta[x] = \text{signum}(dx/dt)$ to represent system dynamics.

- Kuipers' Qualitative Simulation (QSim) is one of the most cited methods of qualitative reasoning. The QSim qualitative model uses an interval-based representation of physical variables and *qualitative differential equations*. Intervals and qualitative differential equations are operated based on the sign (+/-) of the variables and threshold values associated with qualitative system changes.

- The qualitative process theory introduced by Forbus regards the modeled system as a collection of static but capable of interacting *objects*, *quantities* describe the world and their objects based on inequalities and *processes*, which are capable of modifying the objects, their relationships and their properties.

Many successful monitoring and diagnosis applications, such as GDE+ [StDr89] and Mimic [Dvor92], have been developed based on these initial approaches. Each technique has strengths and weaknesses. In common, they have their capability of handling incomplete information by modeling process variables or small processes. Nevertheless, these models are usually too large to be used for the description of a whole complex system.

Also worthy of being mentioned is the Situation-based Qualitative Modeling and Analysis approach (SQMA) [Fröh97]. It models a system using qualitative descriptions of situations that can take place. These situations, gathered in a table of valid situations, correspond in technical systems to a qualitative description of the finite set of states where the process can reside. Each situation consists of a particular combination of intervals in the process variables. The validity of these situations is proved against a set of physical rules, qualitatively expressed and handled.

2.3.5 Fuzzy techniques for process monitoring

After the formulation of the Fuzzy Set Theory in 1965, there has been an astonishing development in the application of fuzzy inference systems in industrial automation in the form of fuzzy controls and fuzzy expert systems [Rebo02]. From these applications, FIS have been further developed and applied to cover other industrial automation applications; among these is the online monitoring of technical processes. However, in the last decade there have been a number of mathematical configurations (such as the in section 2.3.3.2 described fuzzy automata), which exploits other features of the fuzzy sets for process monitoring. This is a very ample field, therefore only a selection of these techniques will be presented in this section.

2.3.5.1 Fuzzy inference systems as process monitors

One of the most cited fuzzy modeling techniques was introduced in 1993 by Sugeno and Yasukawa from the Tokyo Institute of Technology in Japan ([SuYa93]). This approach is based on the notion of qualitative models as fuzzy models (models based on fuzzy sets or fuzzy numbers) of the linguistic description of the system behavior. Moreover, they define Qualitative

Reasoning as the reasoning based on these qualitative (fuzzy) models. This is a very general method for modeling under ignorance about the structure or relations in a real system that consist of a set of *"If-Then-Else"* structures that encode the dynamic behavior of a system in a Fuzzy Inference System (FIS).

The main contribution of this approach is the clear separation of the stages (Table 2.2) that comprises a model identification, which is considered only one phase in most of the conventional methods. Their definitions are so general and so precisely formulated that each can be applied independently of the others, as autonomous parts of other identification techniques. At the end of this procedure, the fuzzy model is converted in a qualitative model by using a linguistic approximation method that makes meaningful rules of the abstract partitioning delivered by the fuzzy identification technique. This is performed by matching the rules in the fuzzy model to a set of predefined words and phrases, employing hedges for the adjustment of the membership functions. The final model can be then employed to follow the dynamic behavior of a real system.

Table 2.2: FIS identification stages.

Stage	**Deliver**
Structure identification type Ia	Input candidates
Structure identification type Ib	Input variables
Structure identification type IIa	Number of rules
Structure identification type IIb	Partition of the input space
Parameter identification	→ FUZZY MODEL
Linguistic approximation	→ QUALITATIVE MODEL

The main problem of a FIS based approach, such as the one proposed by Sugeno and Yasukawa, is the large number of rules that are usually required to represent a real technical system. The technique Dynamic Fuzzy Systems (DFS) [KrSc98],[KrSc00], developed at the University of Karlsruhe, is an alternative solution to this problem. In DFS, system dynamics is represented with simple dynamical rules and extrapolated as needed. Expert knowledge is not mandatory to formulate these initial rules; common sense or user/operator knowledge level is usually enough.

These rules are part of a modified FIS inside of an ARMAX structure (s. Figure 2.15). The FIS forms new fuzzy rules as the method advances, by projecting the original rules based on different temporal instances of the system input (u_k) and the estimated output (y^*_k). This produces a fuzzy output that is fed back (without defuzzification) in the ARMAX structure. Existing knowledge about the process can be introduced as trends in the initial set of rules, working in the FIS as an attractor. The result is an important reduction in the dispersion the solution space, hence less spurious solutions are produced. A limitation of this technique is that it can only handle triangular membership functions because of its inference engine.

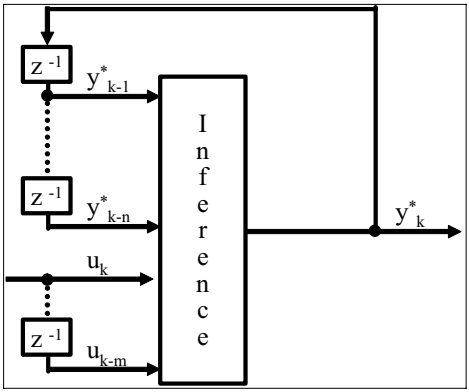

Figure 2.15: DFS main process

2.3.5.2 Fuzzy universal approximators

A universal approximator results from the superposition of nonlinear functions and is capable of reproducing (at least theoretically) any behavior, independently of its complexity. Neural networks, (∞-degree) polynomial structures, Taylor and Fourier series are the most known and used universal approximators. Some fuzzy system configurations also exhibit this property [BSBH98],[WaMe92],[RiRü98]. Recognizing the opportunities in this feature, researches have developed fuzzy universal approximators for the identification and further monitoring of complex process dynamics. As illustration of this approach, the Fuzzy Function Networks are introduced.

The Fuzzy Function Networks can be also found in the literature under the names Dynamic Fuzzy Basis Function Network and Functional Fuzzy Model. These names refer in principle the same approach, saved a few differences. Fuzzy Function Networks [KrRe95],[Krol94] were developed at the Gerhard-Mercator University of Duisburg (Germany) based in turn on the Relational Fuzzy Models [Pedr84], but introduces a crisp analytic function as conclusion of the fuzzy rules, instead of the conventional fuzzy set. This approach uses Sugeno-Yasukawa's method [SuYa93] for the identification of the system structure. An ARX structure is employed for the conclusion part of the rules; consequently, the defuzzified output (the estimated variable) follows the form:

$$y_i(k) = c_i - \sum_{j=1}^{n} a_{ji} y(k-j) + \sum_{f=1}^{m} b_{fi} u(k-f) \tag{2.12}$$

Where a_{ji} and b_{fi} are the parameters of the model that must be identified, while c_i encompasses all the known constants and parameters of the system and correspond to the center of the ith identified cluster. This function can be re-expressed (after some mathematical handling) as:

$$Y_i(k+1) = \left[c_i \theta_i^T\right] \bullet \begin{bmatrix} 1 \\ x(k) \end{bmatrix} \quad \text{..., where} \quad \begin{aligned} \theta_i^T &= \left[a_{1i}, a_{2i}, ..., a_{ni}, b_{1i}, b_{2i}, ..., b_{mi}\right], \\ x^T(k) &= \left[y_{(k-1)}, ..., y_{(k-n)}, u_{(k-1)}, ..., u_{k-m}\right] \end{aligned} \tag{2.13}$$

According to their developers, this technique produces a set of rules, which is less transparent than the rules produced by conventional FIS. In turn, it delivers a model with a good to excellent prediction capability, a reasonable number of parameters and few rules even for very complex systems. The required knowledge level is rather low, common sense or an insight about the physical relations normally suffices.

2.3.5.3 Hybrid fuzzy qualitative approaches

Many joint applications of fuzzy and interval-based methods have been successful in the modeling of the real world's vagueness and uncertainty; particularly in the area of industrial automation. These techniques develop an effective balance between both representation principles, taking advantage of strengths of the one to cover weaknesses of the other. Two representative techniques in this hybrid world are Qualitative Simulator (QuaSi) and Fuzzy Simulator (FuSim).

QuaSi [BoBo94] was developed at the Artificial Intelligence and Robotics project of the Milan Polytechnics Institute in Italy. This approach presents a family of applications, which is able to simulate algebraic-differential models and fuzzy rules-based models. In QuaSi, fuzzy numbers describe the process initial conditions and internal parameters. It produces a hyperspace that encloses the behavior of the modeled system. An alpha-cut operation is applied over the fuzzy dynamic model to produce a set of interval-based system equations. The "alpha-cut Method" projects fuzzy calculus in interval mathematics, based on the alpha-cut discretization of fuzzy arguments. Nguyen [Nguy78] proved that by computing the value of a fuzzy mapping, a generic h-level α-cut of the resulting set only depends on the h-level α-cuts of the arguments:

$$y_h = \varphi(x_1, x_2, ..., x_n)_h = \varphi\,(x_{1h}, x_{2h}, ..., x_{nh}) \tag{2.14}$$

Where an h-level alpha-cut of a fuzzy set A is the crisp set $A_h = \{x \in A,\ \mu A(x) \geq h\}$, with $0 \leq h \leq 1$. So, the Fuzzy algebraic computation may be decomposed into several interval computations, each one having as arguments the alpha-cut intervals of the fuzzy arguments at the same level. These resulting equations are handled with Moore's interval arithmetic [Moor66] in a system simulation based on Taylor series; again, a universal approximation approach is employed.

The second approach, Fuzzy qualitative simulation [ShLe93], was devised at the Heriot-Watt University in the UK. FuSim is an enhancement of the QSim in the direction of the fuzzy systems, which is even older than the most generally cited reference of QSim [Kuip94]. It has provided a starting point to other fuzzy/qualitative simulation principles like Fuzzy Region Simulation (FRenSi) [KeLe99] and Interval Identification [ScKe99].

FuSim uses a common partitioning of the fuzzy variable space for all the variables of all the system's components. An extensive fuzzy interval arithmetic was defined to handle the four-term notation of FuSim's Fuzzy Intervals. To optimize the dynamic simulation, FuSim uses a set of qualitative filters. This reduces the space of the possible next states, resulting of the initial application of the system rules. These filters include analytical (physical, chemical, etc.) relations and temporal constrains.

2.3.5.4 Fuzzy inductive reasoning

Inductive Reasoning [Cell91] was developed at the University of Arizona based on the General System Problem Solver (GSPS) [Klir85]. This technique was later enhanced with fuzzy techniques to create the Fuzzy Inductive Reasoning (FIR) [Nebo94] with the collaboration of researchers of the Technical University of Catalonia in Spain. It is based on an inference engine, which applies the Five-Nearest-Neighbor algorithm, where the prediction is done by looking for similarities between the input and the recorded experience, functioning as case base in Case Based Reasoning (CBR)[4].

FIR follows the conventional structure of a fuzzy system: fuzzification, fuzzy processing and defuzzification. Fuzzy information processing is distributed in two phases. First, an *Optimal Mask Matrix* is identified. The existing knowledge is used for the initial definition of the Mask candidates and general structure. This mask establishes which variables are decisive in the system and how many back-steps in the periodic temporal regression must be considered.

The second phase is the fuzzy forecast, where the mask is applied over recorded input/output data in the form of an episodic matrix that summarizes the experience base of the inference system. The mask defines the variables to be correlated by the inference engine. The information in the episodic matrix is stored as fuzzy triplets: *Class* (numeric), *Membership* and *Side* (respect to the top of the set). A condition for these fuzzy sets is that they must reach its maximal value (membership = 1) only at one point, otherwise the interpretation of the triplet will not deliver a unique value in the defuzzification.

2.4 Analysis of the state-of-the-art

After defining the basic concepts about technological developments for integration of vague and uncertain information in modeling complex industrial automation systems, (s. 2.1 and 2.2), frequently-used representative modeling approaches for process monitoring were described (s.

---

[4] Case Based Reasoning systems [Bart87] make use of past experiences during problem solving. CBR systems deploy case bases of previously solved problems to resolve new problems. When a new problem is found, it is used as reference to retrieve a similar problem case from the case base. The found solution may be applied or adapted to serve the requirements of the original problem. CBR systems can learn by storing the results of a successful problem-solving episode in its case base for future use. The applicability of CBR to process diagnose is evident [Heid96],[ALTH92].

2.3). Based on methods described in 2.2, these techniques cope with the vagueness and uncertainty in industrial automation systems. This section analyzes and compares these modeling techniques, their conceptual foundations and applications. Instead of making judgments about the techniques in this section, they are characterized regarding the research goals. This analysis identifies theoretical resources that are valuable in the management of vagueness and uncertainty, and can be adapted and integrated afterwards in the solution concept.

Regarding qualitative and fuzzy modeling, there is no universal convention about their definition. Authors like Sugeno [SuYa93] sustain that a qualitative model is a special case (a further step) of fuzzy model, while recognized researchers (Kuipers, de Kleer) defend an independent area for the qualitative reasoning. A third group (usually of fuzzy researchers) uses the terms fuzzy and qualitative indistinctly. Other authors conceive them as separate realms and try to develop the common area. They recognize the autonomy and importance of each, as well as similarities and the potentiality of combining them. This notion, with *qualitative* related to Qualitative Reasoning (based on symbols or intervals) and *fuzzy* associated with fuzzy systems (based on fuzzy sets or fuzzy numbers), is the one adopted in this research.

After resolving this inconsistency, the cases listed in Table 2.3 were identified among the analyzed techniques. No application of rough sets to monitoring, simulation or prediction in a technical system could be found. This could be because rough sets are not capable of handling continuous variables (s. section 2.2.4). Just a couple of rough sets applications [STQS00],[WLZW02] where detected, which are related to fault diagnosis, but in these cases rough sets were utilized to analyze previously acquired and discretized information, supporting the decision-making.

Table 2.3: Classification of analyzed techniques

Probability		**Interval-based**		**Fuzzy**
Stochastic Automata	Stochastic/ Qualitative Automata	Qualitative Automata	QuaSi	Fuzzy Automata
Kalman filter / Regression models		Process Theory	FuSim	Dynamic Fuzzy System
Bayesian nets		Confluences		Fuzzy Function Networks
Markov chains		QSim		
Stochastic Petri nets		SQMA		

FIS, DFS and Fuzzy automata are pure fuzzy methods, in the sense that they neither support nor demand interval-based management. On the other hand, techniques such as SQMA and QSim completely lack fuzzy handling. Several tendencies were observed between these two ends, as is

the case of QuaSi and FuSim. QuaSi transforms fuzzily expressed problems into interval-based ones by using α-cut operations, and then applies interval arithmetic to process analytic models based on qualitative differential equations. From this point on, the modeling lies completely in the realm of the interval-based qualitative methods. FuSim is a quite original and complete approach. It adapts QSim's powerful interval arithmetic to the fuzzy world. In comparison, a method like the Dynamical Fuzzy System only offers a partial solution to the modeling problem, the management of the rules in a FIS.

Stochastic Petri nets, regression models and qualitative stochastic automata are suitable for modeling and further handling of process dynamics in technical systems pervaded with uncertainty. However, SPN only recently began to spread from research to industry applications. One reason may be the fact that these techniques deliver complex and difficult to understand models, demanding the participation of experts by the modeling of real technical processes [Buch99].

Several general-purpose and some application-oriented methods have been developed for the identification and simulation of nonlinear processes; the suitability of fuzzy and interval based systems to undertake these automation functions cannot be objected. However, only SQMA, among the studied techniques, addressed the specific interest area of this research: the comprehensive safety assurance in complex technical processes. Safety-related automation remains an interesting future application field for the rest of the studied techniques, which, besides, are only applicable to single variables or small systems. For these reasons, SQMA is adopted as base concept for the proposed solution. However, its performance by handling vagueness and uncertainty is still susceptible of being improved, which requires the careful analysis of SQMA's suitability for the solution of the addressed monitoring problem. This is the purpose of the next chapter.

3 Analysis of the situation-based qualitative modeling approach

The first step for the monitoring a complex technical system is modeling its generally complex processes. Complex processes, be they economic, ecological, social or political, share several features [WuMa02] that are common in most industrial automation systems. The most important of these features are:

- They exchange energy, mass and/or information with their environment.
- They are often composed of a large number of diverse, interacting components.
- Relationships between system components are usually nonlinear and difficult to describe.
- Complex systems are highly heterogeneous, in both time and space.

A comprehensive concept of complex system must consider the role of the observer as well (goals, demanded precision, employed sensors, etc.). The observation determines the flow of information between the system and its model. Hence, complex systems are frequently associated with low quantity and quality of process information. Since process information is the raw material for modeling and further monitoring of a system, deficient information makes the development of a useful monitoring application very complicated. The tragedy of Bhopal (s. section 1.1) is an example of how dangerous the inadequate use of information by the monitoring of complex real systems may be.

All the techniques described in the previous chapter offer mechanisms for modeling vague and uncertain information, but only SQMA is susceptible of being employed for the monitoring of a complex industrial plant such as the one at Bhopal. SQMA was specifically conceived to model complex industrial systems as a whole in safety critical applications. For that reason, the situation-based qualitative solution concept will be introduced in this chapter. A detailed analysis of SQMA will determine a solid procedural and representation framework and will allow the evaluation of how well this modeling technique fulfills the goals of this research. The following analysis will identify the elements in SQMA, which can be improved for convenient usage of vague and uncertain information.

3.1 Situation-based qualitative modeling and analysis

The focus of the Situation-based Qualitative Modeling and Analysis (SQMA), developed at the Institute for Industrial Automation and Software engineering (IAS) at Universität Stuttgart [Fröh97], [Lauf96], is on safety-related applications in technical systems. SQMA is a qualitative

modeling technique, where system components are modeled using qualitative descriptions of situations that can take place on it. Process variables are modeled in SQMA using intervals and characteristic values (represented as one-value-intervals) that can be defined optimally regarding the application. SQMA models are defined upon these situations (collected in a *situation table)*, which describe normal component behavior.

In SQMA, all possible situations are modeled using Moore's interval arithmetic [Moor66], taking into consideration the system structure to get a complete system situation table and a transition matrix representing the qualitative state-space model. Using interval analysis, the description of a variable can just be fitted to the available knowledge and characteristic values or value ranges. The advantage of qualitative modeling methods using intervals is that the internal physical relations do not need to be represented exactly. The models represent only situations in which something "happens" and the model should be able to distinguish between such situations. From this feature arises the denomination of situation-based methods.

The SQMA concept is based on five principles:

(1) "Situations" are a visual and intuitive method of representing the system states derived during the process.

(2) SQMA is component based. The system is divided into small parts that can be easily described in a qualitative model. This also allows the reuse of model sections.

(3) SQMA models have a hierarchic structure, which makes system modeling possible, independent from its size.

(4) Many dynamic systems are susceptible of being represented as electric circuits. This allows using Kirchhof's electrical laws to decompose complex systems as a network of simpler components.

(5) Intervals are used for qualitative (approximate) representation of real dimensions.

These five principles make SQMA suitable for modeling, monitoring and hazards analysis in large and complex systems. SQMA models are multivariable by nature and provide complex process dynamics a high abstraction degree. Other features of SQMA are its suitability for computer-aided design and implementation and for safety-related applications in technical processes. The following sections describe the main elements of this modeling approach, which set the basis for the developed solution concept.

3.1.1 Interval arithmetic

In higher mathematics, an interval is a subset of an ordered set. Intervals are the part of real numbers that start with one number and end with another. The interval {a b} is a collection of points on an imaginary line of real values, starting with "a" (lowest endpoint) and ending with

"b" (highest endpoint). The symbols "{" and "}" stand for parentheses and brackets that indicate whether endpoints are included (closed interval) or excluded (open interval).

- Open interval, enclosed in parentheses as (a b), does not include its endpoints. Therefore, interval (5 15) does not include 5 and 15.
- Closed interval, enclosed in brackets as [a b], includes its endpoints, so 5 and 15 are part of the interval [5 15].

A variety of hybrid combinations are also possible:

- An interval can be half-open, if only one of its endpoints is excluded, such as the left-half-open interval (a b], and the right-half-open interval [a b).
- Intervals can be unbounded, as in (-∞ b] and (a +∞).
- Intervals that can be described with only one number, as in [a a], are called a singleton.

The interval arithmetic, postulated in [More66], is an expansion of the real arithmetic to work with intervals. The subset of Moore's interval arithmetic used with SQMA implements operations defined according to the formula $\{a\ b\}*\{c\ d\} = \{x*y \mid a \le x \le b,\ c \le y \le d\}$; where "*" substitutes one of the symbols "+", "-" or "∩" and a, b, c and d are real numbers known as interval limits. These operations are explicitly enunciated as follows:

- Addition: $\{a\ b\} + \{c\ d\} = \{a+c\ \ b+d\}$
- Negation: $-\{c\ d\} = \{-d\ \ -c\}$
- Subtraction: $\{a\ b\} - \{c\ d\} = \{a\ b\} + \{-d\ -c\} = \{a-d\ \ b-c\}$
- Intersection: $\{a\ b\} \cap \{c\ d\} = \begin{cases} \{a\ b\} & \text{for } b > c \wedge a \ge c \wedge b \le d \\ \{c\ b\} & \text{for } b > c \wedge a \le c \wedge b \le d \\ \{a\ d\} & \text{for } b > c \wedge a > c \wedge b > d \\ \{c\ d\} & \text{for } b > c \wedge a < c \wedge b > d \\ \{b\ b\} & \text{for } b = c \\ \text{undefined} & \text{for } b < c \end{cases}$

Again the symbols "{" and "}" represent parentheses and brackets in the definitions above, depending of the open-ness or closed-ness of the respective intervals. The open-ness or closed-ness of the limits in the intervals that results from each operation must be defined as well. In general, each endpoint preserves its limit type (open/closed) after applying the negation operator. Since in addition and subtraction operations, the symbols "(" and ")" dominate before "[" and "]", the following rules are explicitly established:

- ( ] +/- [ ] = ( ]
- [ ) +/- [ ] = [ )

- ( ] +/- ( ] = ( ]
- [ ) +/- [ ) = [ )
- ( ) +/- [ ] = ( )

For the intersection, the order among the interval borders a, b, c and d must be considered together with their corresponding interval limit types (open or closed). In general this order can be analyzed using the relational operations "<", "=" and ">" or a combination of them. Specifically in the case of interval arithmetic and contrary to the "classic" single-value arithmetic, a given relation between two intervals is satisfied, if there is at least one value, common to both intervals, that satisfies this relation. Relational operators work as it is expressed in the following rules:

- $(a\ b) < (c\ d)$ for $a < d$
- $(a\ b) > (c\ d)$ for $b > c$
- $(a\ b) = (c\ d)$ for $(a\ b) \cap (c\ d) \neq \varnothing$
- $[a\ b) < (a\ b)$
- $(a\ b] > (a\ b)$

Unlike single-value relational operations, these relations are not mutually exclusive. Pairs are satisfied concurrently, as in $a < c$ and $b > d$; therefore, relational operations lose their ordering capability by comparing overlapping intervals.

The interval arithmetic was extended in [Fröh97] to describe the dynamic behavior of the system. To describe changes in qualitative variables, i.e. to perform temporal comparisons, Fröhlich introduced the differential operator "δ" to represent the derivation in time of a variable and thus its direction of change. The following declaration defines the differential operator. Let x1 and x2 be two consecutive values of a variable x, with x1 preceding x2 in time, then:

- $\delta x > 0$ for $x2 > x1$
- $\delta x = 0$ for $x2 = x1$
- $\delta x < 0$ for $x2 < x1$

Only the sign is required as the result of the differentiation. It indicates the direction of change in the variable. The speed of change, contrary to what happens with quantitative functions, has no meaning in qualitative operations. The result can be then represented as the intervals $(-\infty\ 0)$, $[0\ 0]$ or $(0\ +\infty)$ representing that the variable decreases, remains unchanged or increases respectively.

Because of the above discussed nondeterminism of the relations "<", ">" and "=", outcome of the differential operator can be nondeterministic as well. For instance, $\delta x = [0\ +\infty)$ may represent two different cases: either $x2 < x1$ or $x2 = x1$. Nondeterminism in interval arithmetic corresponds to the natural ambiguity of the qualitative concepts represented by the intervals.

The SQMA interval arithmetic also defines a special dynamic for components capable of storing energy. An energy-storing component cannot go instantaneously from one situation to another; it must transit all intermediate intervals. This characteristic of energy-storing components, called *steadiness,* satisfy the relationship case $\delta x \neq \pm\infty$ and is marked using the symbol "$".

3.1.2 Modeling procedure

The SQMA modeling procedure is illustrated in Figure 3.1. The modeler represents the whole system hierarchically and decomposes the innermost level (#1 in Figure 3.1) into components. Variables in the components are modeled after that using intervals and characteristic values represented as one-value-intervals (#2 in Figure 3.1). The description of a component is completed with the declaration of the physical rules that will be used for situation verification, using the in previous section presented interval arithmetic.

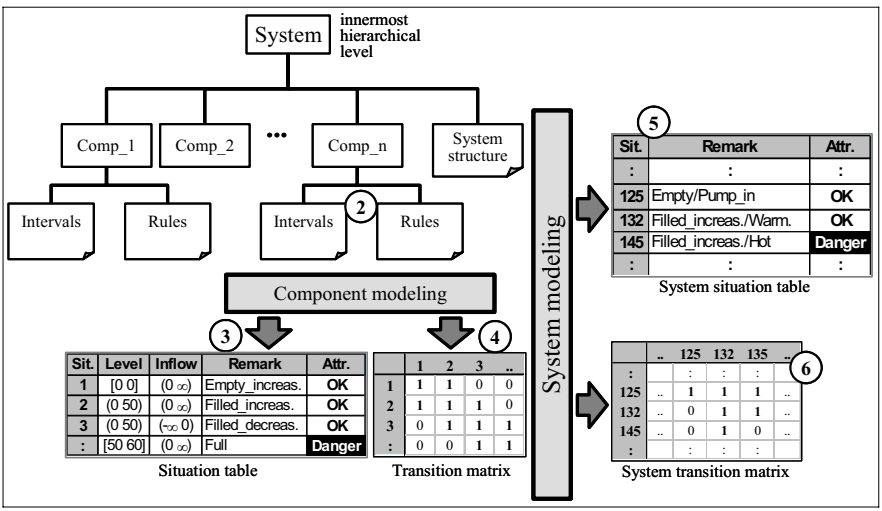

Figure 3.1: Situation-based qualitative modeling

During the computer-aided modeling of a component, every situation candidate is verified. A given component situation is valid if it comply with all the rules declared for the corresponding component. The intersection operation is used instead of the relational operator "=" to check the fulfillment of a rule. This leads to no contradiction with the equality operation (s. 3.1.1). Qualitative equations are used with operands and operators on the left side of the equation equaled to zero (homogeneous form). Rules are satisfied by the existence of intersection between the arithmetic result and the interval [0 0], or what is the same, by the existence of the value zero inside of the in left side resulting interval. Impossible situations (those that do not

satisfy all the component rules) are detected and discarded from the component situations set. An SQMA component model is defined upon these situations collected in a component situation table (#3 in Figure 3.1). It describes the normal behavior of the component.

The relationships ruling the system dynamics, which were described using the extended interval arithmetic as well, are now used to validate the transitions between component situations. The transition rule testing procedure is similar to the implemented for the situation validation and is realized with the help of component modeling software module too. Valid transitions are marked with the number one, while invalid transitions remain unmarked (zero). This information is represented in a quadratic transition matrix (#4 in Figure 3.1) with situations in rows and columns. A mark indicates the possible transition from the situation in the field's row to that in the corresponding column. The transition matrix completes the component model. This model can be reused (in case of similar components in the system) and further combined at system level. Moreover, complete component models can be stored in an SQMA model library for future use.

To model a complete system, the situation tables of all the components at the currently modeled level are combined. Equations assembled by a network modeling software module, following the structure of the system, describe the relationships between components. These equations or system rules express material and energy balances in a way, which is similar to the Kirchhof's laws in electric circuits. These system rules are used to eliminate invalid coincidences of component situations from the system situation table. A computer program (system modeling software module) executes the system rule validation, following the same rule-checking procedure employed by the verification of situations and transitions in the components. From this computer-aided validation results a situation table, with all the possible behaviors of the entire system (#5 in Figure 3.1). The corresponding system transition matrix (#6 in Figure 3.1) is built by verifying the component transition matrixes after the following rule: *A transition between two system situations* (in the system transition matrix) *is only possible, and therefore marked, if the corresponding transition* (in the component transition matrix) *is also possible for each single component*. With that is the situation-based qualitative model of the hierarchical level complete. The structure of this model is identical to the structure of the component models, and can be reused, further combined and stored in an SQMA model library. If the system has only one hierarchical level (see examples in [Lauf96], [Fröh97], [Manz98] and [Manz99]), The entire system model is complete.

Large manufacturing facilities, such as a petrochemical plant or a refinery, may require resolving several hierarchical levels to model the whole system. Pumps, valves, heaters and vessels can be modeled as single components and combined to model evaporators, reactors or distillation columns which are then combined following different configurations to form a number of processing units that can be then combined to model process plants and finally the entire petrochemical or refinery complex. These production complexes may comprise a couple

of plants, tens of processing units and hundreds of single-component types, for example. In these cases, the described procedure is repeated for each hierarchical level until the completion of the system model.

The SQMA modeling approach relies on combinatorial analysis. SQMA models represent exhaustively the solution space; all possible solutions are analyzed and, if accepted, included in the system model. However, the realization of such combinatorial models imposes the use of computer based modeling aids. The SQMA Modeling Software set consists of three modules: `Net2Equ` to process the system structure, `MkModel` for the component modeling and `Analyse` for the system modeling. Two additional modules of this software set, `Evaluate` and `EvalTran`, support the offline situation and transition analysis. They are, therefore, not required for online process monitoring, which is the final goal for the models developed in the present study.

3.1.3 Situation-based process monitoring

SQMA was conceived for the development of safety critical automation applications such as hazards analysis [Lauf96],[Bieg03] and process monitoring in industrial technical systems [Fröh97],[Manz01a]. Situation-based process monitoring, i.e. process monitoring based on SQMA models, proceeds as shown in the example in Figure 3.2.:

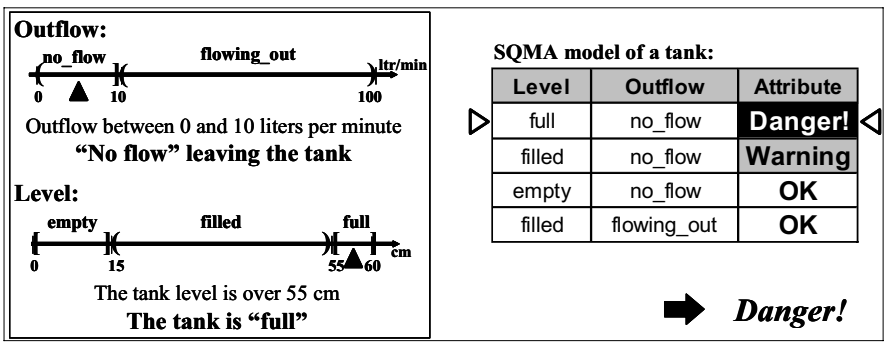

Figure 3.2: Process monitoring with SQMA models

The monitoring application works in a hierarchical level over the process control system. It reads the pertinent measured variables and control commands from the running process through its automation hardware and software. These values are then compared against the defined intervals to determine the matching qualitative value (right in the figure). The data is then validated and interpreted against the situation table of the SQMA model (left in the figure) to determine the current process state. The validated situation is afterwards displayed with its corresponding label ("empty increasing", "full/warm", "overrun" …) and attribute (for example

“Danger”, “Warning” or “Ok”). If no valid situation is found for the observed values, a notice is issued instead, signaling the observed behavior as abnormal or faulty. An invalid (and therefore not modeled) situation can be, for instance, the result of defective sensors or not modeled structural changes (leaks, obstructions, etc.) in the process.

In this example, the control system reads an outflow of 5 lt./min and a level of 57 cm from the tank with the help of the installed sensors. These values correspond to “no flow” leaving the tank and a condition of “full” in the tank level. These qualitative values are effectively represented in the situation table of the SQMA model. The resulting situation corresponds to a dangerous process state.

The information about the current process state may be used to determine, based on the corresponding transition matrix, which situations can be reached from the current process state. Coming situations are determined at any time by reading the row that corresponds to the current situation. The actual coming situation should be one of those represented in the columns, which fields are marked with one. The central idea is that instantaneous (e.g. within the current time period) transitions between nonadjacent situations (unmarked transitions) are not possible.

Based on the component-oriented nature of SQMA models, Fröhlich [Fröh97] defined two basic and alternative structures for the application of the SQMA procedure to process monitoring (Figure 3.3). These structures consider the distribution of tasks and the available processing resources; these are:

A) Evaluation of the system models in runtime based on SQMA component models.

B) Use of previously consolidated SQMA system models.

Because of the hierarchical structure in SQMA models, the components mentioned in A indicate in fact the subsystems in the hierarchical layer immediately below of the system layer, which can be in some cases very complex. An analysis at this level evaluates SQMA component models (Figure 3.3, top) in connection with the available on-line information. These are e.g. measured values from the technical process and control actions and commands from the control system. The analysis is afterwards completed for the whole system. The advantage of this approach is that some monitoring functions can be accomplished in the components. This allows the distribution of safety-critical tasks. Nevertheless, the entire model analysis must be completed in real-time. This usually requires substantial computing power and processing time, which are not always available for the on-line automation of complex systems.

The second structure (B) consists on monitoring the process based on an already consolidated system situation table and system transition matrix (Figure 3.3- bottom). The behavior of the real process is compared with the system situation table using the existing information about measured values and control commands. The processing requirements and implementation cost by the supervision of the desired process behavior are appreciably smaller than by distributed

structures as the one in the first case, but at the cost of less flexibility and a more modeling expenses. In particular, if a modern, object-oriented automation concept is used, structure A offers substantial structural advantages and supplies better results. If the computing resources of the monitoring application do not suffice, a good result can be achieved also with structure B as well.

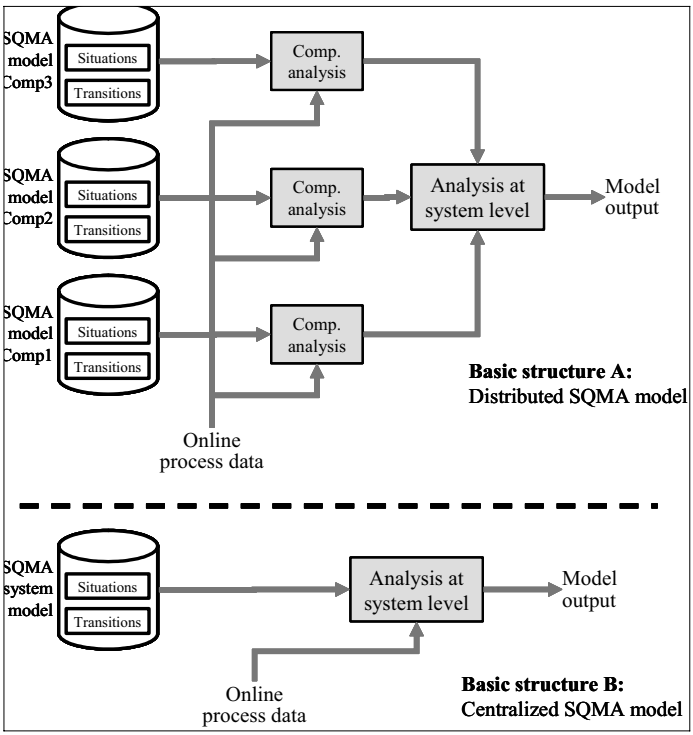

Figure 3.3: Basic SQMA process monitoring structures

The described monitoring approach, based on SQMA models, was extended in the Situation-based Qualitative Monitoring and Diagnosis (SQMD) [Manz01a], which was developed at the Universität Stuttgart (IAS) as well. A SQMD model comprises both an SQMA model and a set of simplified numeric models of the dynamic behavior in energy storing components, which have a significant time dependency. The new numeric counterpart enhances the dynamic performance of SQMA models, achieving a drastic improvement of the short-term forecasting precision, and thus the reduction of the space of possibly reachable system situations. SQMD calculates periodically all possible trajectories for a defined time slot to reduce the solution space. This reduced qualitative solution space contains all reachable system situations for the

predefined time slot and can be examined for possible process deviations and process faults. Situations that cannot be reached from the current state within the next temporal window are this way eliminated. Since SQMD is in fact an SQMA model with a numerical counterpart, the following analysis on SQMA pertains SQMD as well. If a part of this analysis corresponds specifically to SQMD, this will be explicitly indicated

3.2 Complexity management in SQMA

While the term “complexity” has become a buzzword across many fields in science, it has various meanings. Since “complexity” is difficult to define and the many of the attempted definitions are incomplete in one respect or another, rather than discuss these definitions, the reader is referred to [Heyl97]. However, two strongly related aspects of complexity will be considered: *structural complexity* that reflects the compositional diversity and configurational intricacy of a system; and the *functional complexity*, which emphasizes the variety of function and behavior, the heterogeneity and nonlinearity in system dynamics

3.2.1 Structural complexity

Structural complexity results in a system from a large number of coupled variables. The structure of a system may be represented through a minimal unique description of the system under consideration, such as the algorithmic information for messages of infinite length. In this sense, structural complexity tends not to be related to the size of a system, but rather to the number of hierarchical subcomponents and their placement regularity (or irregularity). There is a close relationship between the maximal structural complexity that can be modeled and the granularity, and thus the precision, demanded for the model. A building can be described as composed of a number of bricks in a certain order, but this does not consider the molecular structure of the bricks, which are structurally complex themselves.

Representing structural complex systems like the petrochemical plant at Bhopal requires managing multiple hierarchical levels and identifying components and connections in each level. The hierarchical structuring of a system and its decomposition in components (blocks, objects or subsystems) are two widely proven approaches to overcome structural complexity. Both approaches are used in SQMA (Figure 3.4), which makes it suitable for the description of structurally complex processes. At least theoretically, the combination of both mechanisms allows the modeling of processes, which structure is infinitely complex. The physical availability of resources for the representation, storage and further handling of the model set the limits for the method applicability.

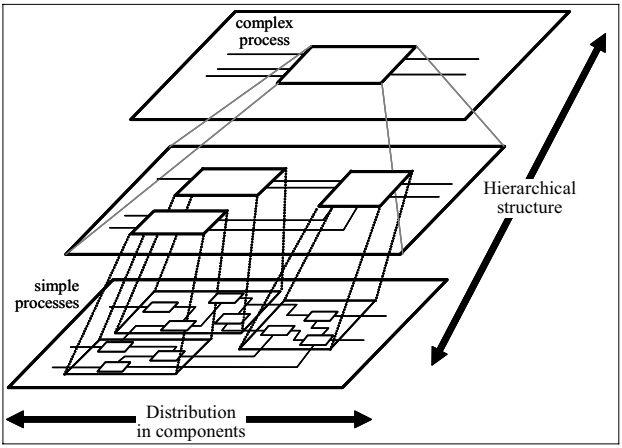

Figure 3.4: Modeling of structurally complex processes in SQMA

3.2.2 Functional complexity

Structural complexity pertains the static part of a system, which depends on the elements composing the system and on the position and connections of these elements inside of the system. Functional complexity is related to how a system performs, i.e. how the system's processes are perceived from the point of view of the functions they are capable of accomplishing. A system is considered functionally complex if the relationships between their components are difficult to describe (like when they are partially unknown) or demand difficult calculations. Consequently, the functional complexity by the modeling of a system is related (but not restricted) to the availability of adequate information, which is usually insufficient by working with industrial processes.

The state of the art by the domination of functional complexity is not as well developed as by the structural complexity control. There exist many approaches, which are suitable for different abstraction levels and precision demands. Qualitative modeling is one of these approaches. Qualitative models have proved to be effective in the representation of complex functions [BrLi98], where the application of deterministic models is completely unthinkable. Qualitative modeling techniques such as SQMA represent only what is considered important in the system, and rely on intervals for the representation of physical values. This reduces noticeably the modeling and processing effort making possible the modeling of (functionally) complex systems.

Functionality also refers to the interactions between the components and with the environment. Since these interactions occur in many forms, there are different types of functionality. Their combination and interdependency will also fall within the scope of functional complexity. The

Bhopal tragedy, for instance, resulted of the combination of several factors, although they did not take place at the same time. Water leaking from the refrigeration system into a tank, which was already too full with MIC, caused the accident. The reaction of the MIC with the water caused the toxic gases, but it was the amount of these reactants, which determined the magnitude of the catastrophe. The reserve tank was also full; therefore, the tank could not be emptied. It was also stated that several safety devices were temporally switched off. In total 40 tons of this toxic gas were released because of the coincidence of all these conditions, in spite of the existence of instruments capable of detecting them separately.

In the Bhopal accident, the particular deviations and problems were detected in time, but they could not be regarded as a whole. The monitoring of individual variables is not sufficient, in order to ensure the safety of industrial plants. The combination of factors, i.e. the interactions that take place inside of the system, must be considered too. SQMA is suitable for the management of this type of complexity. It does not model the single process variables but all the possible combinations of factors that can be observed in the system (the situations). With that are also all possible interactions modeled, independent of their (functional) complexity.

3.2.3 SQMA model size and precision

The precision of a model can be defined in as the maximal lecture error, or what is the same, the minimal extension that can be exactly measured. This precision, related to the roughness (granularity) in partitioning the appraised variable, is established by the intended application, not by the process itself. In SQMA, model precision corresponds to the maximal number of situations that can be perfectly differentiated from one another. This coincides with the initial situation space (before the validation of the situations), and therefore with the combination (multiplication) of the number of intervals defined in all the variables of the system.

A less technical feature related with the precision of an SQMA model regards the readability and descriptive power of the information that this model may deliver. SQMA delivers little information about the current system state; only one set of qualitative values (a situation), without indication about the real state of the system referred to the intervals that define these values. This makes impossible the required recognition of situation transitions. Usually many intervals need to be defined on the process variables in order to achieve an SQMA model with a precision, which is suitable for industrial automation applications. This may cause the geometrical explosion of SQMA models, which eventually limits the applicability of the method.

Poor information about the current state also encourages the interpretation of possible future situations as current risks to complement the information about the present. However, because of the nondeterminisms of the transition forecasting in SQMA, the space of future situations (or

current risks, depending on how it is interpreted) tends to be very large. The size of this space makes the required assessment in many cases even more difficult.

The main threat by the process modeling with SQMA is the geometrical explosion. This problem is present in all noninteracting qualitative modeling methods [BoBo94][Kele99]. An SQMA model grows in geometrical progression with the number of intervals defined for the process variables, which in turn is determined by the demanded precision, and by the number of process variables (system complexity). This limits seriously the construction of precise qualitative models when working with complex automation systems.

Contrasting SQMA's tendency to geometrically explode, SQMD delivers a solution space that has the minimal physical size possible; yet, the cost of this feature is impressive. The effort demanded by the dynamic modeling, which is just a part of the total modeling work, is comparable with the required by the construction of a simplified quantitative model. The term "simplified" depends on the demanded precision. What is worse, this work is not reduced with the reuse of already developed components, because quantitative models of process dynamics depend strongly on the structure of the system. Although the cost of this accuracy is for some applications payable, it is excessive for most of the typical applications of the qualitative modeling. Regarding the size of the model, SQMD eliminates the need of transition matrix and with that delivers a model (a system situation table and a set of dynamic equations), which is significantly smaller than the corresponding SQMA model, but at the same time, more complex.

3.3 Vagueness and uncertainty representation in SQMA

The adoption of a situation-based approach for the modeling of complex systems determines the framework for the information representation in the model. This framework should be enhanced through a convenient representation of vague and uncertain information, which requires the detailed analysis and comparison of different approaches for the integration of vague and uncertain information in the model.

3.3.1 Interval-based modeling of vague and uncertain information

Several problems are summarized under the modeling of uncertainty. One of them is the management of incomplete knowledge in the sense of a partial or total ignorance about the process to be modeled. The term "incomplete knowledge" is usually cited in association with SQMA, so a detailed analysis of this feature is at this point required. In SQMA, the expression "incomplete knowledge" signifies that by the modeling the information about the system is simplified before using it. This is achieved with the partition and characterization of variable spaces in intervals associated with observable and meaningful situations.

Incomplete knowledge does not mean that the information about the process is incomplete in the sense of its availability (ignorance, uncertainty) or in any way insufficient. In fact, SQMA requires a deep yet not analytically represented knowledge about the system. This is necessary to identify components, an adequate (interval) partitioning of the variable spaces, system rules and sampling period for a monitoring application. SQMA, similar to most modeling methods, discards or approximates partially unknown or ill-defined parameters and variables. Only the most probable parameter values or scenarios are taken into account during the definition of the component intervals and rules. These factors, in the case of a monitoring application, are intrinsically associated with the perception and management of the time in the model. Therefore, in SQMA incomplete information cannot be associated with uncertainty or ignorance about the process.

The modeling of vague concepts such as “too full”, “almost empty” or “somewhere between Full and Overflow” is indeed possible, yet, it requires the definition of a new interval for each one of these cases (s. Figure 3.5). This would mean scaling the model size, since this grows in a geometrical progression with the number of intervals in the variables as progression rate (s. previous section). It is necessary to achieve a fair representation of this information in the model without augmenting its size so drastically.

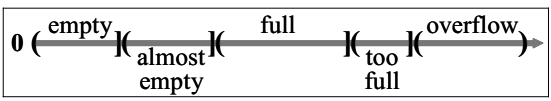

Figure 3.5: Interval-based representation of vague concepts

The management of uncertainty regards the nondeterministic nature of transition models. While monitoring a process, it is necessary to decide in presence of more than one possible solution. This is especially critical when the typically huge tables of the possible future situations are considered. Here is a problem of multiple possibilities the one that introduces the uncertainty. SQMA does not offer indexes or measures such as probabilities, which could support the analysis and further use of the information it delivers.

3.3.2 SQMA extended with Fuzzy Intervals

Permantier [Perm90] remarked, in his dissertation on knowledge representation for industrial automation systems, the importance of handling soft limits between intervals for the information modeling during the development of knowledge-based industrial automation systems. This should allow the analysis in transition regions. Intervals are fundamental in SQMA, and the analysis of these transition regions is very important in any process monitoring initiative. Furthermore, Fröhlich [Fröh97] recommended the description of the intervals based on the Fuzzy Set Theory and the management of event probabilities in the SQMA models as a way of

improving the diagnosis capability of SQMA. The first approach in this sense was the Fuzzy Interval concept (FI) [Fröh94], which represents the gradual approaching to the interval.

Fuzzy Interval and Fuzzy Interval Arithmetic (FIA) are concepts formulated as extensions of the crisp intervals employed in SQMA and the previously described Moore's interval arithmetic. Moore's interval arithmetic only handles yes/no decisions, i.e., values lie within an interval or not. The fusion of the concepts of fuzzy sets and qualitative crisp intervals offers the possibility of reshaping these immediate transitions into gradual changes. With the definition of FI it was suggested [Manz01b] that the degree in which a number is within an interval (membership, μ), can be handled as a probability factor. This factor, transferred to the system situations, can result in the definition of the probability of occurrence of the corresponding state.

Fuzzy Intervals can only use trapezoidal membership functions. This restriction makes possible to simplify the notation. An FI can be described by means of two intervals separated by a colon. The part preceding the colon represents the base of the trapezoid, while the other interval (after the colon) encloses the region where the membership is total. Figure 3.6 shows the FI [2 5]:[3 4], where the typical trapezoidal function can be observed.

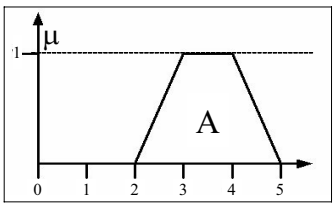

Figure 3.6: Example of Fuzzy Interval representation

An extension of Moore's interval arithmetic is Fuzzy Interval Arithmetic (FIA), which defines the same three basic operations: Addition, Subtraction, Negation and Intersection. FIA reproduces these operations applying their original definitions simultaneously and correspondingly in both intervals of the FI notation. For example, with the fuzzy intervals A = [2 5]:[3 4] and B = [1 3]:[2 2], the following operations can be performed:

Addition: A + B = [2 5]:[3 4] + [1 3]:[2 2] = [2+1 5+3]:[3+2 4+2] = [3 8]:[5 6]

Subtraction: B - A = [1 3]:[2 2] - [2 5]:[3 4] = [1 3]:[2 2] + [-5 -2]:[-4 -3] = [-4 1]:[-2 -1]

Negation: -A = -[2 5]:[3 4] = [-5 -2]:[-4 -3]

However, the interval intersection is no longer applicable in FIA, therefore it was redefined based on the intersection of fuzzy sets.

Although fuzzy interval proved to be a suitable concept for handling vagueness in SQMA [Rebo02], a few observations and conceptual inconsistencies have remained from its

formulation. Fuzzy Interval is a representation principle between both, traditional SQMA intervals and fuzzy sets. Fuzzy Intervals have their own definition and notation. They are not completely fuzzy (in the sense of the Fuzzy Set Theory) and at the same time, they are more than merely modified SQMA Intervals.

Comparing fuzzy sets and fuzzy intervals it can be found that only triangles and trapezoids can be used as membership functions, otherwise, it would be necessary the complete redefinition of the FI notation and the corresponding FIA. Besides, the negation in FIA matches neither the fuzzy set complement nor the NOT operation in fuzzy logic. A further inconsistency is the interpretation of membership in FI according probabilities. A fundamental concept in Fuzzy Set Theory is that a given object is not limited to belong only to one class (in SQMA, one interval), instead any element may be represented with a specific membership degree (including zero as possible value) in each class.

Nonetheless, Fuzzy Intervals go away from the conventional SQMA intervals as well. An FI is an interval with a valuation function as described in [Fröh97]. For example, the overlapping intervals is not allowed in SQMA, whereas it is mandatory with Fuzzy Intervals [Rebo02], and the question about fuzzy interval closed-ness or open-ness has no consequence, even though it is fundamental in the interval definition.

The concepts of FI and FIA were never completely developed and integrated in SQMA. Not only because of the described inconsistencies and limitations, but also because handling FIs duplicates the modeling effort and the model's storage requirements, when compare with the SQMA method using crisp intervals.

3.3.3 Transition modeling

SQMA represents the dynamic behavior of a component based on a transition matrix. This transition matrix is an adjacency matrix (one to indicate adjacent situations, zero everywhere else) that indicates which transitions are possible inside of a time period. This time period is nowhere defined nor declared, the modeler selects it based on its expertise. This representation in essence coincides with Lunze's definition of Qualitative Automata (s. 2.3.3.1).

Lunze described the nondeterministic nature of Qualitative Automata, which is also applicable in the case of SQMA. The prediction of reachable situations based on a transition matrix delivers a number of possible solutions (nondeterminism). It is sure that the actual next situation is among those listed, however no information is provided that may help by the evaluation or classification of the possible solutions. This information involves a high degree of uncertainty, which grows, if predictions several steps in the future are required. If a deterministic dynamical model is employed instead, as in SQMD, the solution space is minimized by means of the elimination of those situations that cannot be physically reached in the following time window.

This reduces appreciably the nondeterminism, and with that the uncertainty by the assessment of the results.

As a way of reducing the nondeterminism in Qualitative Automata, Lunze proposed to enrich the automata with stochastic information. Stochastic Qualitative Automata introduce probabilistic information in the transition matrix. This helps by the representation of the uncertainty in the model and the further evaluation of the information that this model may deliver. Nevertheless, the idea of integrating probabilities in the SQMA transition matrix have never been explored prior to this research work.

The representation of uncertainty with possibilities can be also considered. Because of the affinity between fuzzy and possibilistic models it is thinkable the development of a synergetic approach based on FIs, as it was suggested in [Manz01b]. However, the availability of models of the process dynamics in the form of fuzzy or possibilistic automata are considerably less frequent than the handiness of probabilistic models.

3.4 Requirements on a situation-based solution concept

Functional and structural complexity can be controlled with different abstraction mechanisms. Functional abstraction is reached in SQMA by the representation of process states as qualitative (interval-based) situations, whereas structural complexity is controlled via hierarchical and component oriented decomposition. SQMA facilitates the approximate modeling of complex systems, which would make it suitable for safety-critical industrial automation applications, where the modeling of complex relationships in a complete system is a requirement. These SQMA features are summarized in Table 3.1.

Table 3.1: Mechanisms for complexity management in SQMA

SQMA feature	**For the treatment of …**
Hierarchic structure	Structural complexity
Component oriented	Structural complexity
Situation-based	Functional complexity
Interval-based	Functional complexity

Regarding the handling of vague process knowledge can be said that the management of vagueness and uncertainty is still susceptible of being improved. In SQMA, there is a problem of vagueness (modeling of vague concepts) as well as of uncertainty (elucidation of the actual situation among the group of possible results), so both problems need to be considered. Representing and handling vagueness effectively requires the definition of a notation system, which uses weighting values to express the confidence on the information elements, as well as the development of a set of algorithms, which can deal with such weighting values.

Managing uncertainty, on the other hand, must support the output assessment. Within this context, uncertainty management is not necessarily related with the representation of information in the model, but with the way this information is used. Combined, both approaches may enable the recognition of transitions, which is currently impossible in SQMA.

Another important feature is the readability of the information delivered by an SQMA model by process monitoring. Having only one situation for the current state in SQMA is definitively easy to read, but its information content is very poor. If the size of the predicted situation table is also considered, the information content is enhanced but the result is more difficult to read. However, the main drawback of SQMA is the tendency to geometrical explosion of its solution space, particularly in the case of the transition matrix. The size of a model is strongly related with the complexity of the system, but also with the precision requested from its model. Besides, in complex industrial processes, the development of the deterministic models required by SQMD is not always an option.

For that reason, SQMA solves the problem of the modeling of complex systems only partially. A new approach for information representation in SQMA is therefore necessary. This approach should improve the relationship between the process complexity, the size of the resulting models, and the demanded precision and readability. This requires reducing the model's speed of growth in relation with its precision by the modeling of complex systems. This can be achieved with the incorporation of the information hidden in vagueness and uncertainty through a more convenient representation principle, which should be capable of enhancing the relation size/utility and the descriptive power of the model. The description of this new representation principle is the purpose of the next chapter.

Based on the previous analysis, the requirements regarding to the online process monitoring and the accompanying qualitative model of the technical system can be formulated as in the following sections.

3.4.1 State observation and transition recognition

An application based on the monitoring concept must be able of running with the process, acquiring the data available to the automation platform and comparing it with the situations in the model in real time. The observer must be capable of indicating the actual situation of the technical process according to the model. It is also important to recognize the transitions between the process states. Therefore, the online monitor should be capable of determining when the process is about to change to a new state from the current one.

3.4.2 Identification of impossible or hazardous states

Although impossible situations are not included in a well-developed system model, the particular combination of process values associated with these situations can be observed because of faulty instrumentation. An application based on the monitoring concept must be able to identify these impossible states and to present them accordingly. Since the final goal of the monitoring application is to safeguard the technical process and humans from danger, each dangerous or potentially dangerous situation must be recognized as early as possible. Rules defining undesired operation states shall be evaluated for each observed situation, and their results accordingly displayed.

In most real applications, however, it is not enough the characterization of the current process state to assure its safe operation. The identification of hazardous states also includes the early detection of risky operation conditions. In this context, it is understood under current operation risks those faulty states that can be reached from the current state of the process. With the intention of supporting fail recovery, the system should additionally offer information about the nearest reachable “safe” situations.

3.4.3 Vagueness and uncertainty management

As it was already stated in section 1.3, the goal of this research work is the development of a process monitoring concept, which is capable of using vague and uncertain information as they were defined in section 2.1.4.

On the one hand, representing and handling vagueness requires the definition of a notation, which weights values to express the confidence on the information elements, as well as develops algorithms that can deal with such weighting or confidence values. The outcome of this process must also express the certainty of the information it delivers.

On the other hand, modeling a technical system under conditions of uncertainty must serve the decision taking in uncertain situations, i.e. the uncertain inference. Uncertainty management must support, therefore, the output assessment. Within this context, uncertainty management is not necessarily related with the representation of information in the model, but with the way this information may be used.

3.4.4 Model size reduction in relation to demanded precision

A common problem with the quantitative and the qualitative modeling methods is that complexity grows rapidly with the size of the system and demanded precision. Model precision determines its applicability. There is a practical low limit for model precision, under which the information delivered by the model is insufficient. For the definition of model precision, the notion of measurement precision (maximal reading error, minimal extension that can be exactly

measured) is employed. This precision is determined by the roughness of the employed measuring method.

Another important characteristic of a process model is its size. There is a practical size limit. If a process model grows too fast, eventually it cannot be completed or further used. When this limit is reached, it is said that the model has numerically exploded. The size of a model is strongly related with the above-described precision and the complexity of the modeled technical system.

A compact modeling procedure should represent complex processes without loosing precision. This demands to handle the triple relationship between the application's precision requirements, the complexity of the modeled technical system and the resulting size of the model.

3.4.5 Compatibility with existent SQMA models

SQMA is a modeling approach, which has been continuously developed since 1992. Since an important feature of SQMA is the reusability of models in complex structures, the existing SQMA models, including qualitative models in SQMD, should be portable to a platform based on the new concept.

SQMA also includes mature modeling software tools. Since a further strength of SQMA is the high degree of automation of the modeling task, the reliability of the available tools is important. A manual validation of the resulting model is generally impossible because of its size and complex relationships. Therefore a minimal intrusion into the internal code of the existing software tools is desirable, in order to avoid unnecessarily jeopardizing the software's reliability. Any change must be traceable.

4 Vagueness representation with Rough Intervals

The objective of this research work is the development of a model-based monitoring method suitable for complex technical processes. SQMA was chosen as base modeling method, because of its capability of modeling large and complex systems. Yet, as it was discussed in the previous chapter, SQMA alone is not enough to fulfill the formulated requirements. The way process information is modeled and processed in SQMA must be enhanced through a better handling of vagueness and uncertainty to improve the triple relationship between process complexity, model size and the precision demanded by the monitoring application.

Process information in SQMA is represented and processed based on intervals. These intervals represent the qualitative values that a process variable can assume. Such qualitative values are usually coupled with vague concepts such as “soon”, “warm”, “full” or “fast”, which are not easily represented with intervals. On the other hand, uncertain information in SQMA regards the nondeterministic nature of transition models. SQMA does not offer indexes or measures to support the analysis and further use of the information in the generally huge tables of reachable situations. They are two different problems that require a separate treatment: modeling vague concepts (vagueness) and handling nondeterminism (uncertainty). This vague and uncertain information (left side in Figure 4.1) need to be integrated in the SQMA process model (Figure 4.1, right).

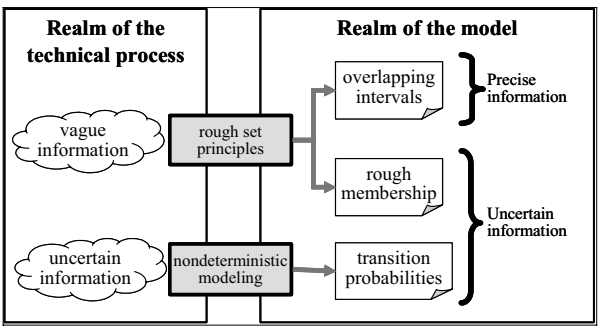

Figure 4.1: Representing vague and uncertain information in SQMA models

The first part of the solution concept aims the modeling of vague information in SQMA. It consist on the analysis of qualitatively or linguistically expressed process knowledge based on principles of the Rough Set Theory, which allows extracting from vague concepts, precise information (s. Figure 4.1, right-top) capable of being further processed in SQMA. This method is the Rough Interval concept. This chapter will define and describe the Rough Interval concept

precisely. It will analyze the result of using Rough Intervals instead of conventional crisp intervals in SQMA as well.

The solution concept for the modeling of the technical process is completed in the next chapter. It will analyze the consequence of introducing the rough membership function in SQMA to reduce all the vagueness of the original concepts to uncertainty (Figure 4.1, right-center) in the border regions (s. 2.2.4, and 2.2.5). Next chapter will also address the modeling of the inherent process uncertainty (nondeterminism, Figure 4.1, right-bottom).

4.1 Rough Interval concept

The Rough Interval (RI) concept is the core of the representation of vagueness in SQMA models. Rough Intervals provide a framework for the interpretation of crisp intervals based on indiscernibility. Indiscernibility (s. 2.2.4) is a property defined inside of the Rough Set Theory, which allows the extraction of precise information from vague concepts, by comparing these vague concepts with the available process knowledge. The required process knowledge may be provided by sensors in the technical process and technical specifications, upon which it can be decided, when a physical (measured) value clearly belongs to the represented concept, and when this membership is uncertain.

Rough Intervals were introduced because rough sets were conceived only to handle discrete objects [STQS00]; they cannot be used to represent continuous values such as temperature, flow or level in SQMA. A Rough Interval is a particular case of a rough set. They fulfill all the rough set's properties and core concepts, including the upper and lower approximation definitions. Inside of the upper approximation interval (UAI), the variable could take the represented qualitative value (a vague concept in rough sets), or what is the same, it is clear that outside this interval, the variable cannot take it. The second element of a rough set, the lower approximation, can be also redefined on this basis: In the lower approximation interval (LAI), it is sure that the variable takes the represented qualitative value.

The Rough Interval concept also satisfies the mathematical definition of rough set's upper and lower approximation in (2.3) and (2.4). If a particular qualitative value C must be represented over a variable, the two enveloping intervals, I_* and I^*, can be defined. The implications (4.1) and (4.2) represent the relationship between the variable $x \in U$ (where the universe U can be any complete set of continuous or discrete values), the qualitative value I (defined on x) and the intervals I_* and I^*.

$$x \in I_* \Rightarrow x \in I \tag{4.1}$$

$$x \notin I^* \Rightarrow x \notin I \tag{4.2}$$

According to Rough Set Theory, the imprecision of a concept is reduced to uncertainty areas about their borders. Within these uncertainty areas, in Rough Intervals as in rough sets, no definitive conclusion about the problem is possible. Therefore, a rough membership function must be defined.

For practical reasons, a straight linear function is adopted as the base membership function. It not only satisfies all the conditions listed in Table 2.1 and [Pawl94] offering a trade-off among the infinite possibilities, but also implies the lowest calculation and representation effort, which at the end signifies a more efficient management of time, which is critical in real time applications such as process monitoring. This linear representation of the probabilistic rough membership function corresponds to the assumption of a uniform probability distribution, which is usually adopted as a worse case scenario for the representation of systems under uncertainty, i.e. when no information about the probabilistic distribution of the modeled events is available.

Table 4.1 summarizes the main characteristics and properties of Rough Intervals and compares them with that of crisp intervals and the equivalent application of the membership of continuous variables to fuzzy subsets for their qualitative representation.

Table 4.1: Comparison of qualitative representation principles

	Fuzzy subset	**Rough Interval**	**Crisp interval**
Representation of	vagueness	vagueness and uncertainty	imprecision
Membership functions	trapezoid, Gaussian, sigmoid, …	straight lines defined in boundary regions	crisp Yes/No decisions
Overlaps	no restrictions or conditions	mandatory in boundary region, not allowed in Lower Approximation	not allowed
Cross-dependency	each set is defined independently	probabilistic complementarity	no overlaps and no empty spaces allowed
Total membership	$\Sigma\mu>1$ allowed (Fuzziness)	$\Sigma\rho = 1$ at any point (Probability)	$\Sigma m = 1$ at any point ($m \in \{1,0\}$)
Interpretation	possibility, membership or degrees of truth	probability of being member of the interval	inside or outside of the interval

Rough Intervals, like fuzzy intervals (s. 3.3.2), may be represented as two superposed intervals. The first part indicates the base of the resulting trapezoid, i.e. the “upper approximation” to the represented qualitative value. The second interval delimits the “lower approximation”, the region where the concept certainty is total. Attention must be paid to the fact that the terms “upper” and “lower” in Rough Intervals (and rough sets in general) are not used regarding their graphic representation. An upper approximation interval in a Rough Interval would correspond to the “base” of a Fuzzy Interval; whereas the lower approximation interval corresponds to its “top” (compare Figure 3.6).

This double-interval notation is inconvenient when working with SQMA, since it would require twice more storage space than using crisp intervals, making SQMA models larger than they already are. However, this notation is not necessary by working with RIs. Figure 4.2 illustrates the partitioning of a temperature variable with RIs. Both RI elements are separately depicted, together with the corresponding rough membership function. Observe in this figure how the overlaps of the upper approximation intervals coincide exactly with the empty zones between contiguous lower approximation intervals and with the regions where membership functions are defined. These three representations (upper approximation, lower approximation and membership function) are equivalent; they contain the same information. That means two of these representations can be reproduced from the third.

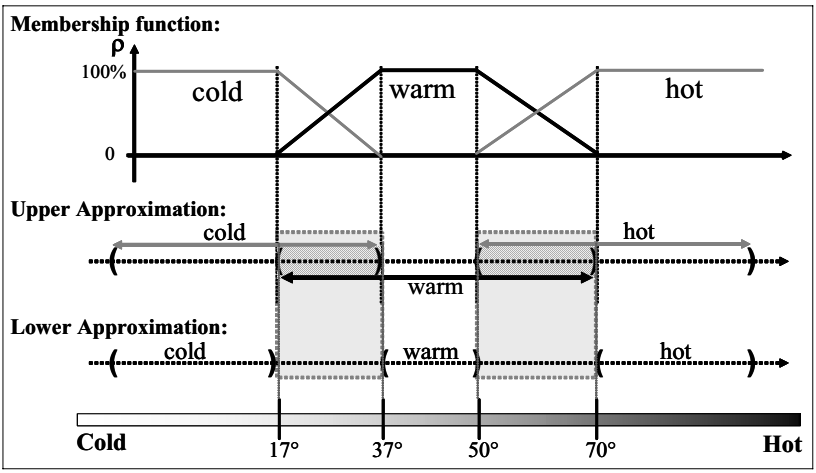

Figure 4.2: Upper and lower approximation intervals

This equivalence in the represented information makes possible the representation of only the overlapping upper approximation intervals to describe the entire RI partition. The employment of this crisp-interval-like representation is crucial for the integration of Rough Intervals in SQMA. So represented, there is no difference between Rough Intervals and overlapping crisp intervals. This representation assures a minimal impact with the introduction of the Rough Interval concept in the SQMA modeling procedure. SQMA models with crisp intervals and RIs are identical. Only the interval overlaps, which are not permitted in the conventional SQMA, indicate the use of RIs.

RI overlaps are transient areas between two clearly typified regions. In the overlap, a variable changes gradually from one quality to another (like from “warm” to “cold” in Figure 4.3). Vague intermediate qualitative values such as “between warm and cold”, “almost cold” or “not warm enough” can be used to describe these uncertainty regions. The superposition of two

intervals can represent this information. By analyzing a particular value inside of an overlap, the qualitative values corresponding to both intervals must be considered together with the complementary rough memberships. This enhances the precision and descriptive power of the model in the transition areas, where a higher degree of detail is required.

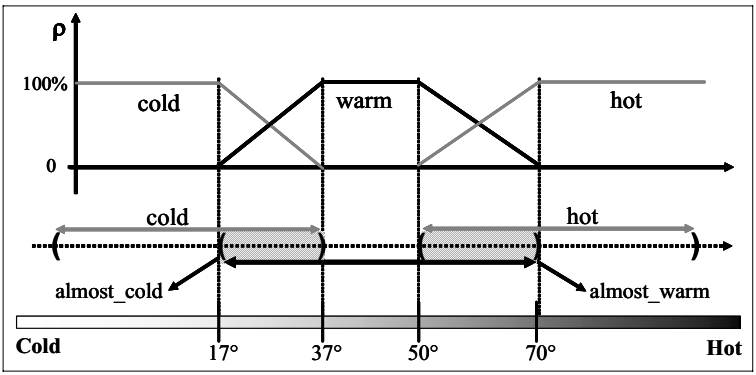

Figure 4.3: Single interval representation of Rough Intervals

The described representation and interpretation framework, based on the UAIs, could have been developed using the lower approximation intervals instead. However, the superposition of qualitative features (overlaps) is a notion that is closer to the human understanding than representing featureless (empty) areas, with qualitative descriptors assigned ad hoc. The chosen approach, the representation of the upper approximation intervals, allows handling Rough Intervals independently from their overlaps. Intervals are characterized, combined and verified during the modeling. Overlaps emerge from the interpretation of the model using real values from the technical process. Where real values hit lower approximation intervals, no overlap will emerge. SQMA handles this case exactly as the case where crisp intervals are used.

4.2 Interval arithmetic with Rough Intervals

In an SQMA model, RIs are sufficiently described through their upper approximation intervals (or, given the case, through their lower approximation intervals). This is possible because the model includes all the RIs defined in a variable, making the overlaps evident. Nevertheless, during the evaluation of component and system rules, RIs that are used isolated from other RIs in the same process variable, because the situation to be checked comprises only one qualitative value (i.e. just one RI) for each process variable. For this reason, each RI in a single-interval notation (as it is found in the model) must be changed to its equivalent double-interval notation before being used for the evaluation of an SQMA rule. This double-interval notation is similar to the notation developed for the Fuzzy Intervals. A given RI (for example A) is represented by

its upper approximation interval (A^*) as the base of the trapezoid, followed by its lower approximation interval (A_*) as the top of the trapezoid, both separated by colon (A^*: A_*).

Once this transformation is made, Rough Intervals are compatible with the Fuzzy Intervals Arithmetic (s. 3.3.2). Rough Interval Arithmetic (RIA) can be analyzed considering two groups of operations: those directly related to the interval treatment according to Moore's interval arithmetic and the group of set and logic operations associated with the "set" nature of Rough Intervals.

4.2.1 Interval operations

As it was the case with FIA, RIA defines three basic operations: Addition, Negation and Intersection. Based on the double-interval notation, RIA manages upper approximation and lower approximation intervals separately with Moore's interval arithmetic. No attention must be paid to membership functions during these operations. For example, with the Rough Interval A = [2 5]:[3 4] and B = [1 3]:[2 2], the following operations can be performed:

Addition: A + B = [2 5]:[3 4] + [1 3]:[2 2] = [2+1 5+3]:[3+2 4+2] = [3 8]:[5 6]

Subtraction: B - A = [1 3]:[2 2] - [2 5]:[3 4] = [1 3]:[2 2] + [-5 -2]:[-4 -3] = [-4 1]:[-2 -1]

Negation: -A = -[2 5]:[3 4] = [-5 -2]:[-4 -3]

These are the same operations defined for the FIA (compare section 3.3.2). These functions satisfy the core concepts of upper and lower approximation of rough sets. Consider the last example, which encompasses the other two basic operations:

$C = B - A = (B^*:B_*) - (A^*:A_*) = (B^* - A^*):(B_* - A_*)$ ➔

$C = (1\ 3):(2\ 2)-(2\ 5):(3\ 4) = ((1\ 3) + (-5\ -2)):((2\ 2) + (-4\ -3)) = (-4\ 1):(-2\ -1)$

C not only satisfies the rough set properties described in section 2.2.4, but also is conceptually consistent. The lower approximation C_* contains all the elements that without doubt results from subtracting a number certainly contained in B (i.e. B_*) from a number certainly contained in A (A_*), while the upper approximation C^* contains all the elements that cannot be excluded after subtracting a number possibly contained in B (B^*) from a number possibly contained in A (A^*).

Fröhlich's extensions of the SQMA interval arithmetic, the differential operator and the steadiness condition (s. 3.1.1), can be also employed with RIs. The differential operator, however, deliver the same crisp intervals as in interval arithmetic (in double-interval notation): $(-\infty\ 0):(-\infty\ 0)$, $[0\ 0]:[0\ 0]$ and $(0\ \infty):(0\ \infty)$. The integration of these operators in RIA allow the modeling of dynamic relationships in SQMA using RIs.

4.2.2 Set and logic operations

Set and logic operations exploit the "set" nature of the Rough Intervals. The set operations *unification*, *intersection* and *complement* are defined for the RIA. For the unification of Rough Intervals no changes were introduced, since this operation coincides with that defined for rough sets and its equivalent operation in Fuzzy Interval Arithmetic. By the application of the intersection, however, particular conditions may arise that will be detailed in the next section.

The complement of a Rough Interval is clearly disjoined from the sign inversion operation (Negation), which regards the interval (not the set) nature of the Rough Intervals. To avoid notational ambiguity, as symbol for RI complement, its logical equivalent "¬" (NOT) is used instead of the traditional "-", which remains consequent with its use in SQMA and Moore's interval arithmetic. The complement operation results in the set of all the elements in the universe of discourse that are not contained in the RI introduced as an argument for this operation.

Although it was not originally considered in SQMA, the complement operator allows the simplification of rules that require the exclusion of a characteristic that otherwise must be represented by joining all possible qualitative values of the variable but the one negated. This is illustrated with the following example, which shows in both cases the same rule; omitting the complement operator in the right side:

$$\text{Soft} \cap \neg\text{Very\_Hot} = \{\text{Soft} \cap \text{Hot}\} \cup \{\text{Soft} \cap \text{Warm}\} \cup \{\text{Soft} \cap \text{Cold}\} \ldots.$$

RI logic is defined, based on the above-described set operations and the uncertainty represented for the membership value. These basic operations, namely AND, OR and NOT, correspond to the set operations intersection, unification and complement respectively, but handle elements of the set (in this case the interval), not the set as a whole. The result of a logical operation with Rough Intervals depends on the value where it is evaluated. In RIA, the following logical operations are defined:

- $(A \text{ AND } B)_z = \rho A \cap B(z) = \rho A(z) * \rho B(z)$
- $(A \text{ OR } B)_z = \rho A \cup B(z) = \rho A(z) + \rho B(z)$
- $(\text{NOT } A)_z = \rho_{\neg} A(z) = 1 - \rho A(z)$.

… with $z \in U$ and $A, B \subseteq U$.

These operations comply with the last three properties of the rough membership functions in Table 2.1. The formulation specifically obeys to the need of handling the membership of a value to a given RI based on probabilities.

- The compound probability of two independent facts, for example the probability of two process variables assuming the qualitative values represented by the Rough Intervals A and

B respectively, corresponds to the product of the single probabilities of occurrence, which is represented by their membership values.

- The total probability of two independent facts, for example the probability of a process variable taking the qualitative value represented by the Rough Interval A or another one taking the qualitative value B, corresponds to the addition of the single membership values.
- The probability of a given value being part of a Rough Interval A is equal to one minus the probability of not being part of it.

These three probabilistic rules coincide with the defined logic operations and establish the basis for the further probabilistic management of rough membership. The first rule is the very fundament for the definition of a situation in SQMA and the aggregation of membership values in superior hierarchical levels.

A last operation included in RIA is the alpha-cut operation. This operation, adapted from the Fuzzy Set Theory, allows generating a crisp interval out of a Rough Interval. The application of an h-level alpha-cut operation with ($0 \leq h \leq 1$) to a Rough Interval would result in the unique interval of all the elements that belong to the original RI with a certainty (i.e. rough membership value) greater than, or at least equal to, h (s. Figure 4.4). Applying a 0,5-level alpha-cut to a process variable partitioned with RIs delivers the equivalent Rough Interval. These crisp intervals have borders exactly at the midpoints of the RI overlaps.

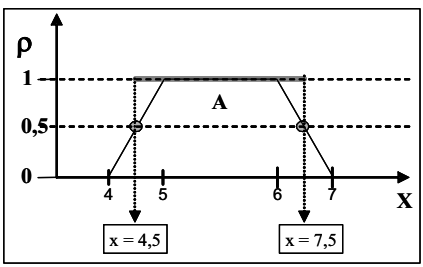

Figure 4.4: Rough Interval A = [4 9]: [5 8] and its 0,5-level alpha-cut

4.2.3 Rough Interval intersection

A special case among the operations defined in RIA is the intersection operator. Let two Rough Intervals be defined in the double-interval notation: A = [4 9]:[5 8] and C = [5 10]:[6 9]. The first interval in each RI refers to the UAIs (from 4 to 9 for the interval A) while the second interval refers to the LAI (from 5 to 8, in A). These intervals are shown in Figure 4.5. The product of intersecting A and C is a new Rough Interval (Figure 4.6), which contains elements

that are surely contained in A and in C (LAI), and those that cannot be excluded from the common area between them (UAI).

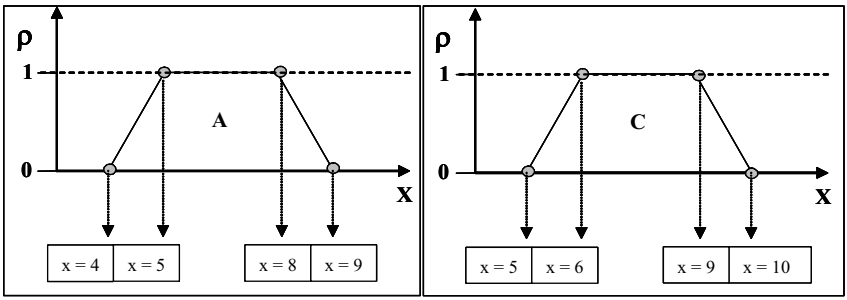

Figure 4.5: Rough Intervals A and C

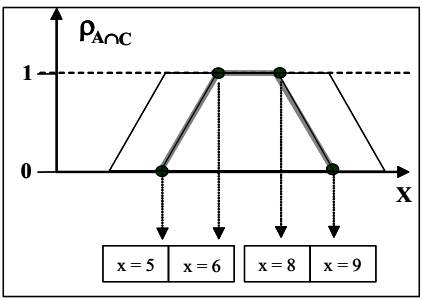

Figure 4.6: Intersection of Rough Intervals A and C

By definition, an arithmetic operation over one or two RIs must result in an RI; and this is the case here. However, the result of an RI intersection may not fulfill the defined notation conventions (compare FIA intersection in [Fröh94]), as in the following case:

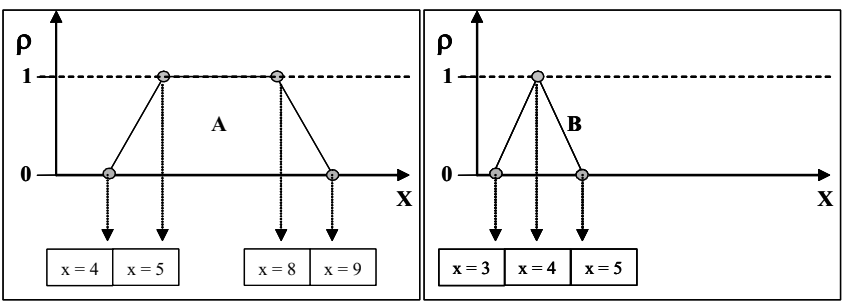

Figure 4.7: Rough Intervals A and B

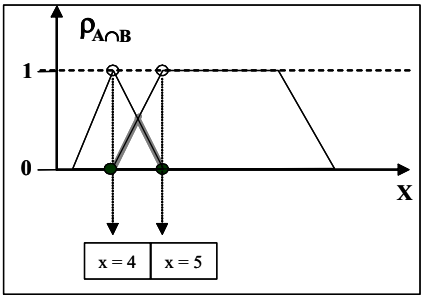

Figure 4.8: Intersection of Rough Intervals A and B

The intersection of RIs may result in a new RI with a maximum rough membership lower than one. In this case, the last point in the resulting lower approximation interval precedes its initial point, i.e. the LAI is defined by an invalid interval that may be forced to be consistent with the established RI definition. It would denote that there is no certainty region about this concept. Let A and B, defined as A = [4 9]:[5 8] and B = [3 5]:[4 4] (s. Figure 4.7), be considered for this second case. The set produced by the intersection of A and B is a new RI ([4 5]:[5 4], Figure 4.8), where the end of the LAI is smaller than its starting point. This kind of RI denotes a case without 100% of certainty area, but that cannot be completely discarded. The maximum membership of this RI is 0,5.

However, it is impossible to assure that in all the cases where the RI intersection is realized, the result is another RI according to the previously defined notation. The counterexample in Figure 4.10, displaying the intersection of Rough Intervals D and C (s. Figure 4.9), demonstrates this third solution case in RI intersection.

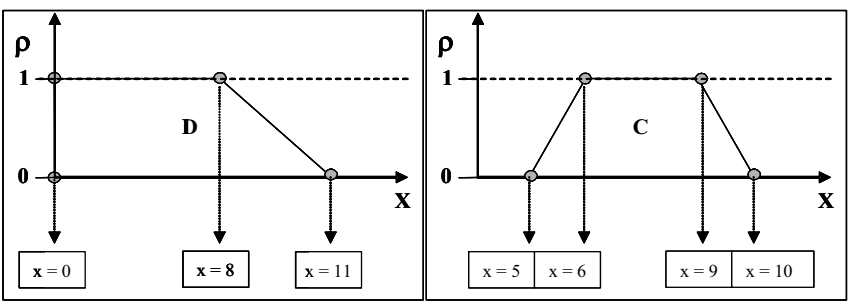

Figure 4.9: Rough Intervals D and C

In Figure 4.10 five points delimit the intersection area. This cannot be represented using the four points of the double-interval notation and, therefore, cannot be used in further operations. For example, the operation (D ∩ C) + A cannot be performed with the mentioned values, while the

equation $(D + C) \cap A$ is computable with any value of A, C and D because the intersection operator is evaluated at the end of the equation. This characteristic limits the use of RI intersection to the last computing place in a rule, as it is the case by the use of homogeneous equations (equations equalized to zero).

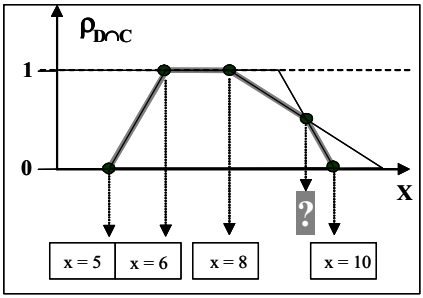

Figure 4.10: Intersection of D and C

However, if RI intersection is still required in the middle of a rule, introducing a new convention may resolve the problem. It is very important to consider that in the core of the Rough Interval concept is the definition of the upper and lower approximation intervals and not the definition of the membership functions as is the case in fuzzy sets and fuzzy intervals. Observing Figure 4.10, the regions can be verified where $\rho = 1$ and $\rho \neq 0$ are respectively consistent with the definitions of UAI and LAI in a Rough Interval. Therefore, the approximation of the membership function to a straight line between the limits of the UAI and the LAI, independently of its real graphical result, may resolve the inconsistency (s. Figure 4.11). So determined, the intersection operation is compatible with the other operations in RIA.

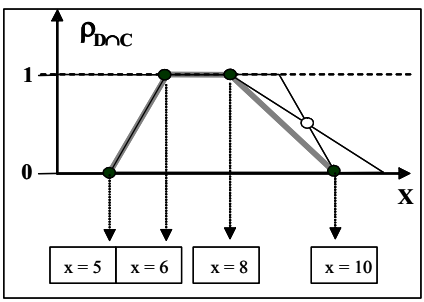

Figure 4.11: Redefined intersection of D and C

4.3 Rough Interval modeling

Determining an adequate partitioning of the process variables is one of the most challenging tasks in building interval-based qualitative models. This is also the case in SQMA. Nonetheless, techniques adapted from other areas such as AI, probabilities and curve analysis can support the determination of Rough Intervals in SQMA. This section introduces some RI modeling approaches.

4.3.1 Heuristic definition of Rough Intervals

Restricting membership functions to trapezoids imposes conditions that help by their definition in a particular variable. Let the upper approximation interval be considered. It comprises the region were the variable would be said to belong to a specific class, to have a specific quality or to exhibit a specific feature for at least a person inside the considered universe. It does not matter if this membership is clear or not. In other words, inside the UAI, *NOBODY can be 100% sure that the variable DOES NOT belong* to this interval or what is the same, it would be for everybody clear that *OUTSIDE the UAI it is 100% sure that the variable DOES NOT belong to the represented class*. The second element of the RI, the lower approximation interval, can also be defined on this basis. In the LAI there is no possible doubt about the membership of the variable to the indicated class. *EVERYBODY can be 100% sure that the variable DOES belong* to this interval or what is the same, *INSIDE of the LAI it is 100% sure that the variable DOES belong to the represented class*. No exceptionally profound knowledge is required to apply these simple definitions as design rules. They do not complement each other; in fact the second one is a subset of the first. The lower approximation interval must be defined inside the upper approximation interval.

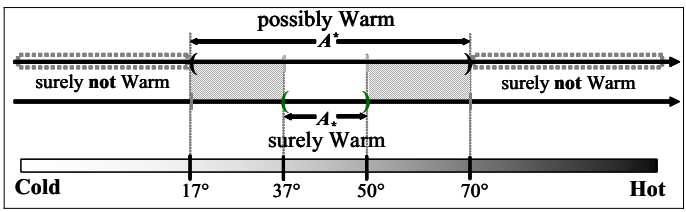

Figure 4.12: Qualitative value “Warm” represented as a Rough Interval

Figure 4.12 shows the example of a class “Warm” in a temperature variable. Common sense and general knowledge can help by the definition of its limits, for example:

- The temperature of the human body is about 37°C, so anything warmer will be never considered “Cold”.

- An average human hand cannot hold an object over 70°C because it is “Hot”.
- When something is colder than the environment (for example 17°C), it cannot be considered “Warm” any more.

A few statistical considerations, together with some sense of symmetry, may assist the modeler by completing the definition of this Rough Interval and its neighbors. This is common praxis, for instance, in interval and fuzzy modeling. In this example, the Rough Interval A resulted for the vague concept “Warm”. The key factor in this example is the use of precise concepts to define a vague one. Verifiable knowledge, statistics or physical laws, which are generally measurable, trustworthy and easier to model than the original vague concept, can support this definition.

4.3.2 Episodic interval identification

The Interval Identification method (s. 2.3.5.3) can be adapted to support the identification of Rough Intervals. Interval Identification is a procedure developed in 1999 at the Technical University of Berlin, with the support of the Heriot-Watt University, for the identification of intervals for the qualitative simulation of continuous variables. Although this method was initially conceived to be used with FRenSi, this is a general-purpose interval identification approach. The experience of this research group is concentrated on the safety of chemical reaction processes; therefore the applicability of this method together with SQMA can be assured.

Interval Identification decomposes an experimental run of the system in qualitative “episodes”, using the sign of the system response and the corresponding first and second derivatives. Additionally the sign of the input variable is registered. The 14 possibilities resulting from combining these four parameters are registered as: A^+, A^-, B^+, B^- … G^+ and G^-, according to Table 4.2. It is important in this simulation that the input variable, the one to be modeled, must monotonically increase or decrease. Those value areas of the variable where there is no change of episode would correspond to the same interval.

Table 4.2: Episodes in the Interval Identification method

Type	A	B	C	D	E	F	G
x							
dv	+	-	-	+	+	-	0
ddv	-	-	+	+	0	0	0

The undefined region between two stable sequences of episodes is marked as a transition interval. These transition intervals can be redefined afterwards as the overlaps of contiguous

RIs. The definition of episodes in each variable would enhance this method for its application in systems with multiple inputs and multiple outputs (MIMO). However, this would increase the modeling effort.

4.3.3 Experimental definition of Rough Intervals

In many applications, the knowledge about the process and process variables is insufficient to allow the complete definition of RIs based on the previously described heuristic definition. Assuming that some structural and behavioral information is available, the still-required knowledge about the process parameters may be gained through experimentation, using an approximate model of the system. A possible modeling procedure is based on the concurrent simulation of the system behavior and monitoring application. Initial values for RIs can be chosen, based on the available information, and then gradually tuned by trial and error. To support this empirical design procedure, a set of Simulink blocks for SQMA (Moore's) interval arithmetic was developed at IAS [Berl02],[Rebo02].

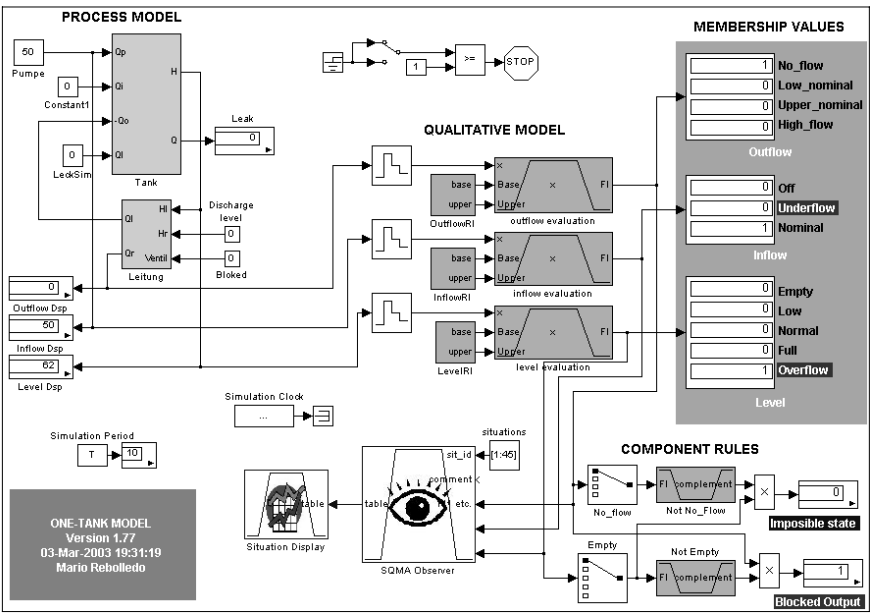

Figure 4.13: Simulink diagram with SQMA blocks

Simulink [TMW99b] is an extension of MATLAB for the simulation of nonlinear, time-continuous systems in a graphical, signal-oriented environment. MATLAB [TMW99a] is an efficient, comprehensive and user-friendly software tool for technical computations. Engineers,

scientists and technicians use Simulink for numeric computation, scientific visualization and system design and testing. The availability of these Simulink blocks for SQMA (s. Figure 4.13) facilitates the development of safety-critical applications parallel to the design of the technical system.

Beyond the determination of RIs, further application parameters, such as component rules, intervals and sampling period, can be determined (or better said: “tuned”) as well. These parameters are strongly related to one another in an SQMA monitoring application. For example, RIs and component rules should be conveniently chosen to minimize the size of the situation tables, yet establishing broad intervals will result in a slowly changing system (from the point of view of the situation monitoring) allowing longer processing periods. For this reason, the combination of this experimental method with other RI modeling approaches is encouraged.

4.3.4 Rough Intervals definition based on fuzzy set identification

Sometimes, the operator’s knowledge is insufficient to develop a prototypical model of the system. This happens especially in cases with many input and output variables and very complex relationships. In these cases, RIs can be determined by analyzing process data, which has been sampled from the real process.

This approach takes advantage of the similarities between fuzzy membership functions and Rough Intervals, keeping in mind that fuzzy membership functions satisfy all the properties required in [PAWL94] for the membership functions in rough sets. Many tools and methods, such as clustering and neuro-fuzzy approaches, have been developed for the identification of a sub-optimal variable space partition in fuzzy set applications. Most of these concepts can be adapted for the definition of Rough Intervals as well. Nevertheless, since a Rough Interval is a particular definition of rough sets, the identification process must assure the fulfillment of the particular features of Rough Intervals.

The analysis of process data offers a promising approach for generating modeling information. The data has to be sampled from the industrial automation process for a representative time period and has to include input and output data sets of all identifiable (but not necessarily explainable) system situations. These data sets are then processed following the schema shown in Figure 4.14. First, fuzzy sets and membership functions are identified. The membership functions of the fuzzy sets resulting from this first step are converted into trapezoid functions. These trapezoid membership functions are finally transformed, in a third step, into Rough Intervals. This transformation assures that the resulting trapezoid covers the entire variable space and complies with the defined characteristics of the RI overlapping (s. Table 4.1). The identified fuzzy rules, together with the membership functions, can support the subsequent formulation of component rules.

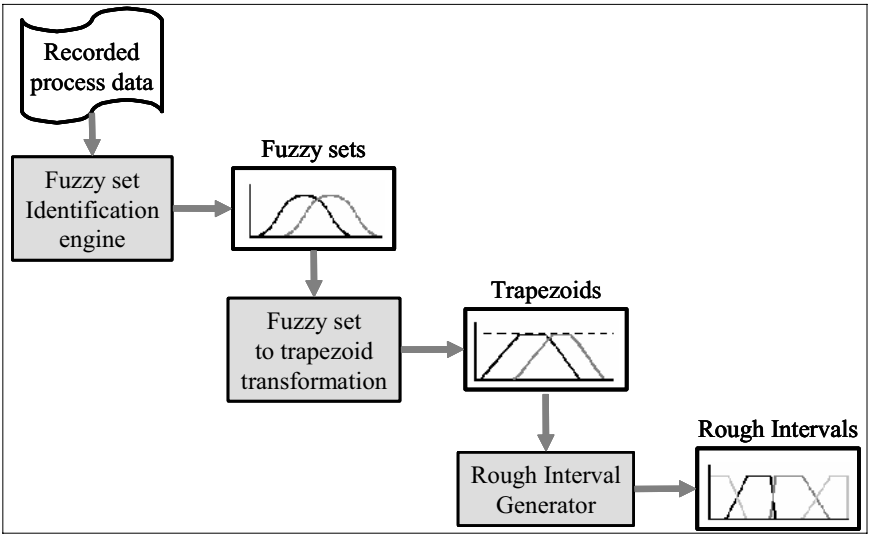

Figure 4.14: Rough Interval modeling with fuzzy identification engine

A Rough Interval Generator was developed in [Will03] as MATLAB command line function. This tool allows the free selection of the fuzzy set identification engine. Several fuzzy modeling techniques, e.g. ANFIS [TMW98], C-Means Clustering [Bezd81], Babûska [Babû98], among others, can be adapted to support the identification of Rough Intervals. The Rough Interval Generator is mostly a transformation algorithm that reshapes trapezoids into Rough Intervals. The preceding step, the input membership function to trapezoid transformation, is unnecessary when working with a fuzzy model identification engine processing trapezoidal input membership functions from the beginning, which is recommended. Whatever the case, computational and theoretical tools, such as the method described in [Tikk02], are made available in case other kinds of fuzzy set have to be processed. A comparison of these trapezoid transformation algorithms is included in [Will03].

4.4 Analysis of vagueness representation

The qualitative modeling approach SQMA, in its original definition, is capable of representing vague knowledge based on intervals. These intervals can be associated with abstract concepts by assigning a qualitative value to each of them. Yet, applying principles from Rough Set Theory, it is possible in SQMA to consider grades or transitions directly in the definition of process variables. With the introduction of RIs in SQMA, it is possible to represent precise abstract concepts such as “black” and “white”, but also all the kinds of “gray” between them.

This gradation is the consequence of the RI overlaps and the definition of rough membership functions on these overlaps. These gray areas are more useful to describe process behavior than perfectly characterized black and white zones. From the point of view of the qualitative descriptors associated with each interval, the overlaps are perceived as independent states of the process variables, i.e. as additional intervals. This results in an important enhancement of the model precision (compare 3.2.3).

The described enhancement of the SQMA's precision and descriptive power by the use of RI does not demand the definition of additional intervals or situations in SQMA. Consequently, it is also possible to simplify the model by keeping the original accuracy. With the areas resulting from the overlapping of RIs, the number of intervals that need to be defined in a process variable to ensure a given model precision is drastically reduced. This means more compact situation tables, since the number of situations grows geometrically with the amount of intervals. For example, (s. Figure 4.15) a component with three variables, each one distributed in five intervals generates a space of 5x5x5 = 125 situations, while the same component defining three RIs generates a space of 3x3x3 = 27 situations with the same degree of detail.

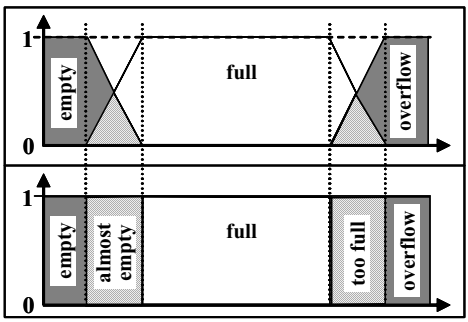

Figure 4.15: Variable space partitioning with crisp intervals and Rough Intervals

The introduction of RIs and transition probabilities in SQMA does not imply the escalation of modeling expenses (costs, time, etc.). The additional effort by modeling RIs (in comparison with defining crisp intervals) is only required to model uncertainty regions, which are not processed in the original SQMA method. The reduction of the model size, the enhanced precision and the minimization of the modeling expenses and processing needs are usually conflicting criteria, which originate different ways of attacking the same problem. In the end, as in most engineering solutions, the problem is reduced to the search for a trade-off between these criteria.

SQMA models with crisp intervals are compatible with the proposed SQMA concept based on RIs. RIs and crisp intervals can be used indistinctly; inclusive to describe the same process variable. The only difference between crisp intervals and the single-interval RI notation is the

overlapping of RIs, which enables the use of the same arithmetic and modeling tools for both kinds of intervals.

5 Uncertainty management in SQMA models

The modeling of process variables using Rough Intervals and the probabilistic representation of the component transition matrixes are the approaches employed to represent vague and uncertain information in SQMA models, for the situation-based monitoring of complex technical processes according to the requirements in 3.4. Chapter 4 introduced the Rough Interval concept in SQMA, as a way of representing vague concepts such as “near”, “soon” or “almost full”, following principles of Pawlak’s Rough Set Theory. This concept divides the space of the vague concepts into precisely defined areas (Lower Approximations) and regions where the represented information is uncertain (boundaries of the vague concepts, s. 2.2.4 and 4.1).

Figure 5.1 shows the evolution of the integration of vague and uncertain information in SQMA models. The uncertainty introduced in the SQMA modeling process by the representation of boundary regions with rough membership functions causes the appearance of spurious situations in the situation table. These spurious situations augment substantially the size of the SQMA model, compromising the SQMA applicability to the modeling of complex systems. This section examines the emergence of these undesirable spurious situations and their subsequent elimination from the SQMA model using the rule confidence value.

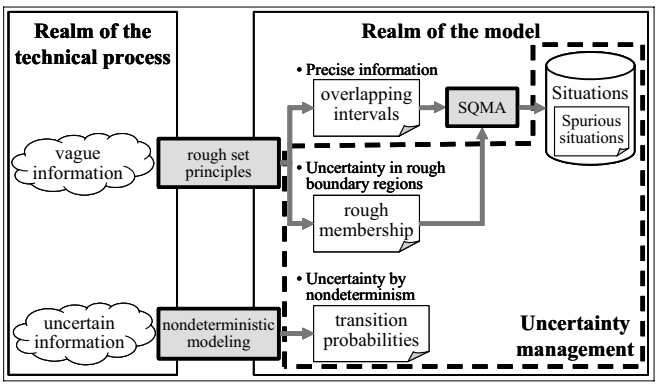

Figure 5.1: Uncertainty management in SQMA

Another important source of uncertainty is the nondeterminism in SQMA’s transition modeling. This nondeterminism comes directly from the technical process, and needs to be represented in the SQMA model conveniently, in order to improve its applicability to process monitoring by enriching the information it can deliver. The Stochastic Qualitative Automata concept (s.

2.3.3.1) is adapted in this section to accomplish this task, based on the already available information in the SQMA model.

Uncertainty management in SQMA consists of both of these elements (Figure 5.1): the treatment of the spurious situations introduced by the uncertainty in the rough boundary regions, and the transformation of SQMA's nondeterministic transition model in a stochastic (probabilistic) model of the system dynamics.

5.1 Uncertainty through Rough Interval overlapping

As introduced above, the integration of the Rough Interval concept in SQMA introduces uncertainty in the model. The assessment of a given process state is precise, as long as all its variables remain inside of the LAIs that were defined in the SQMA process model. Yet, as soon as a variable hits an RI overlap, the uncertainty about the real nature of this variable pervades the whole situation. Therefore, several situations appear that could correspond to the original process state with the probability indicated by their membership values.

The uncertainty introduced by the RI overlapping and the nondeterminism of the relational and the intersection operations in interval arithmetic (s. 3.1.1) come together by the rule evaluation during the situation analysis in the SQMA modeling process. This causes the spurious situations in the SQMA situation tables. This section analyzes the problem of the spurious situations and proposes a mechanism for their elimination

5.1.1 Spurious situations

Spurious situations are component or system situations that are reachable according to the realized computations, but actually cannot take place in the real systems. The adjective spurious is applied in this case to situations that accredit false descriptions to the real process state. Spurious situations appear because SQMA uses the nondeterministic intersection operator to verify rules inside RI overlaps, causing impossible situations to be approved during the modeling process. Analyzing the example of the system rule {T1.h = Valve.dp + T2.h} corresponding to the system Tank-Valve-Tank shown in Figure 5.2, may facilitate the comprehension of this problem.

The rule is tested under two different conditions: Case A with overlapping (with RI) and Case B without overlapping (without RI). The interval partitioning of the system variables is depicted in Table 5.1. The limits of the crisp intervals defined for Case B are exactly in the 50% of the rough membership function of the RI in Case A. The intervals in Case B are the *equivalent crisp intervals* for the RI defined in Case A, and result after applying a 0,5-level alpha-cut (s. 3.1.1) on the corresponding RI. Case A and Case B represent the same variable partitioning with two

different description tools: Rough Intervals and crisp intervals. In fact, the intervals in Case B are the crisp approximation for the RI in Case A, from here the qualifier “equivalent”.

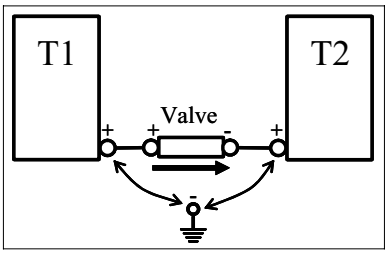

Figure 5.2: Net representation for the system Tank-Valve-Tank

Table 5.1: Cases of the Tank-Valve-Tank system

Case A: with RI			
T1.h	[0 20]	(10 60]	
T2.h	[0 20]	(10 60]	
Valve.dp	(-∞ 0)	[0 0]	(0 ∞)
Case B: without RI			
T1.h	[0 15]	(15 40]	(40 60]
T2.h	[0 15]	(15 40]	(40 60]
Valve.dp	(-∞ 0)	[0 0]	(0 ∞)

The several combinations of these intervals are validated against the rule {T1.h = Valve.dp + T2.h}. Table 5.2 and Table 5.3 show the system situation table for Case A and Case B respectively. In Case A (Table 5.2) no combination of this group is invalid (compare the sequence in the situation numbers), while for Case B there is indeed a reduction of the situation table; situations 4, 6, 7 and 9 have been eliminated. These four situations are impossible, for example situation 6 claims the existence of water flowing from a tank with low level to a filled one, whereas situations 4 and 9 indicate that there is no flow between two tanks with different levels. Yet these four situations satisfy the rule {T1.h = Valve.dp + T2.h} when evaluated with Rough Intervals (Case A).

Since upper approximation intervals represent Rough Intervals in SQMA (single-interval RI notation, s. 4.1), a situation is accepted if the upper approximation intervals defined in this situation satisfy all the component or system rules. Given that UAIs (Case A) are bigger than the equivalent crisp intervals (Case B), the likelihood of interval intersection with RI is greater than with their equivalent crisp intervals, resulting in the acceptance of situations that should be indeed rejected. This problem can be avoided at component level by setting intervals and

reference values in the component rules conveniently. This is impossible with the system rules, because they are determined automatically during the modeling process.

Table 5.2: Situation table for the system Tank-Valve-Tank - Case A

Sit.	T1.h	Valve.dp	T2.h	Description
1	[0 20]	(-∞ 0)	[0 20]	low ← low
2	[0 20]	(-∞ 0)	(10 60]	low ← filled
3	[0 20]	[0 0]	[0 20]	low – low
4	[0 20]	[0 0]	(10 60]	low – filled
5	[0 20]	(0 ∞)	[0 20]	low → low
6	[0 20]	(0 ∞)	(10 60]	low → filled
7	(10 60]	(-∞ 0)	[0 20]	filled ← low
8	(10 60]	(-∞ 0)	(10 60]	filled ← filled
9	(10 60]	[0 0]	[0 20]	filled – low
10	(10 60]	[0 0]	(10 60]	filled – filled
11	(10 60]	(0 ∞)	[0 20]	filled → low
12	(10 60]	(0 ∞)	(10 60]	filled → filled
:	:	:	:	:
:	:	:	:	:

Table 5.3: System Tank-Valve-Tank - Case B

Sit.	T1.h	Valve.dp	T2.h	Description
1	[0 15]	(-∞ 0)	[0 15]	low ← low
2	[0 15]	(-∞ 0)	(15 40]	low ← filled
3	[0 15]	[0 0]	[0 15]	low – low
5	[0 15]	(0 ∞)	[0 15]	low → low
8	(15 40]	(-∞ 0)	(15 40]	filled ← filled
10	(15 40]	[0 0]	(15 40]	filled – filled
11	(15 40]	(0 ∞)	[0 15]	filled → low
12	(15 40]	(0 ∞)	(15 40]	filled → filled
:	:	:	:	:
:	:	:	:	:

However, it is also true that LAIs are smaller than the equivalent crisp intervals. A rule evaluation against LAIs should result in less than or, at most, the same number of situations resulting from evaluating their equivalent crisp intervals. A trade-off between these two extreme cases is developed in the following section, introducing the concept of rule confidence.

5.1.2 Confidence-based reduction of spurious situations

As it was explained in section 4.2.3 there are three possible outcomes by intersecting RI membership functions (RI intersection):

- The RI intersection does not exist.
- The intersection delivers a conventional trapezoidal RI (s. Figure 4.6).
- The intersection delivers a triangular RI with inverted LAI (s. Figure 4.8).

The maximum value reachable by the corresponding membership functions is exactly zero, exactly one or a value in between (one and zero excluded) respectively. This value can express the confidence in the satisfaction of the rule evaluated with the RI intersection:

- Maximum membership value is zero ⇒ the rule is not fulfilled.
- Maximum membership value is one ⇒ the rule is fulfilled.
- Maximum membership value in between ⇒ no decision is possible.

The third case results in a number between zero and one (both limits excluded), which is proportional to the confidence in the fulfillment of the rule by the corresponding situation. Therefore, this value is called rule confidence (C). Rule confidence is hereby defined as *the maximum membership value in the Rough Interval intersection that determines the verification of this rule*. The confidence value of a rule (C) is a value between zero and one, where C = 0 denotes an empty RI (no intersection); C = 1 a trapezoidal RI and C ∈ (0 1) a triangular RI.

Table 5.4: Rule Confidence for Case A

Sit.	T1.h	Valve.dp	T2.h	Description	Conf		Rule evaluation:
1	[0 20]	(-∞ 0)	[0 20]	low ← low	1		
2	[0 20]	(-∞ 0)	(10 60]	low ← filled	1		
3	[0 20]	[0 0]	[0 20]	low – low	1		
4	[0 20]	[0 0]	(10 60]	low – filled	0,5	➔	[0 20]:[0 10] ∩ (10 60]:(20 60]
5	[0 20]	(0 ∞)	[0 20]	low → low	1		
6	[0 20]	(0 ∞)	(10 60]	low → filled	0,5	➔	[0 20]:[0 10] ∩ (10 ∞):(20 ∞)
7	(10 60]	(-∞ 0)	[0 20]	filled ← low	0,5	➔	[10 60]:[20 60] ∩ (-∞ 20]:(-∞ 10]
8	(10 60]	(-∞ 0)	(10 60]	filled ← filled	1		
9	(10 60]	[0 0]	[0 20]	filled – low	0,5	➔	[10 60]:[20 60] ∩ [0 20]:[0 10]
10	(10 60]	[0 0]	(10 60]	filled – filled	1		
11	(10 60]	(0 ∞)	[0 20]	filled → low	1		
12	(10 60]	(0 ∞)	(10 60]	filled → filled	1		
:	:	:	:	:	:		
:	:	:	:	:	:		

Using the rule confidence, it is possible to define thresholds to decide, during the situation validation process, whether a given rule is fulfilled or not (e.g. deciding the acceptance of a rule if it results as in Figure 4.8). Implementing a mechanism to compute the rule confidence by the evaluation of each situation would enable the possibility of reducing the spurious situations introduced by the rough membership function in the SQMA model.

Situation validation in SQMA checks every rule, for each situation. Resuming the example of the Tank-Valve-Tank system, [10 60]:[20 60] ∩ [0 20]:[0 10] = [10 20]:[20 10] results from the rule evaluation with situation 9 (s. Table 5.4). This is a triangular RI with a maximum value of 0,5 (observe the inverted LAI [20 10]). The rule confidence for situation 9 is, in consequence, C = 0,5. Establishing C > 0,4 as acceptance criterion for the rule validation would determine its acceptation. If a higher confidence is demanded (e.g. C > 0,8), this situation must be eliminated from the model. Establishing the rejection of any situation with a confidence under 0,5 would result in a system situation table similar to Case B. The same analysis is also applicable for situations 4, 6 and 7. Eliminating spurious situations based on confidence does not endanger the exhaustiveness of SQMA models. Because of the RI complementarity, there must always exist at least a complementary situation that represents the current process state better than the spurious one.

Table 5.5, Table 5.6 and Table 5.7 display the descriptions of different process states based on the SQMA model for Case A. The values of the process variables are chosen to cover the whole extension of the RI overlaps in T1.h and T2.h. The first column includes the process values for three test points around this area. The second column (header "Intervals") displays the corresponding intervals and their rough memberships. The last column comprises the system situations resulting from combining the intervals identified for each test point. The system situation membership value is also shown. Situations 4, 6, 7 and 9 are highlighted (shaded).

Table 5.5: Membership behavior in lower overlap limit

Process value	Intervals (ρ)		System Situations (ρ)		
T1 = 11	T1 = [0 20]	(0,9)	5	low → low	(0,9)
	(10 60]	(0,1)	6	low → filled	(0)
T2 = 10	T2 = [0 20]	(1)	11	filled → low	(0,1)
	(10 60]	(0)	12	filled → filled	(0)
Valve = right	Valve = (0 ∞)	(1)			
T1 = 10	T1 = [0 20]	(1)	3	low – low	(1)
	(10 60]	(0)	4	low – filled	(0)
T2 = 10	T2 = [0 20]	(1)	9	filled – low	(0)
	(10 60]	(0)	10	filled – filled	(0)
Valve = 0	Valve = [0 0]	(1)			
T1 = 10	T1 = [0 20]	(1)	1	low ← low	(0,9)
	(10 60]	(0)	2	low ← filled	(0,1)
T2 = 11	T2 = [0 20]	(0,9)	7	filled ← low	(0)
	(10 60]	(0,1)	8	filled ← filled	(0)
Valve = left	Valve = (-∞ 0)	(1)			

Table 5.5 shows the behavior of the rough membership in the lower limit of this overlap, while Table 5.6 regards the upper limit. These cases complement one another. The membership value of the situations with C = 0,5 (compare shaded fields in the third column with Table 5.4) reaches

their minimum (ρ = 0) in these extremes. All other situations have membership values ranging between zero and one. Table 5.7 completes the review of the overlap area with the evaluation of three points around the center. The membership values for situations 4, 5, 6 and 9 reach in this area their maximum (ρ = 0,25).

Table 5.6: Membership behavior in upper overlap limit

Process value	Intervals (ρ)		System Situation (ρ)		
T1 = 19	T1 = [0 20]	(0,1)	1	low ← low	(0)
	(10 60]	(0,9)	2	low ← filled	(0,1)
T2 = 20	T2 = [0 20]	(0)	7	filled ← low	(0)
	(10 60]	(1)	8	filled ← filled	(0,9)
Valve = left	Valve = (-∞ 0)	(1)			
T1 = 20	T1= [0 20]	(0)	3	low – low	(0)
	(10 60]	(1)	4	low – filled	(0)
T2 = 20	T2 = [0 20]	(0)	9	filled – low	(0)
	(10 60]	(1)	10	filled – filled	(1)
Valve = 0	Valve = [0 0]	(1)			
T1 = 20	T1 = [0 20]	(0)	5	low → low	(0)
	(10 60]	(1)	6	low → filled	(0)
T2 = 19	T2 = [0 20]	(0,1)	11	filled → low	(0,1)
	(10 60]	(0,9)	12	filled → filled	(0,9)
Valve = right	Valve = (0 ∞)	(1)			

Table 5.7: Membership behavior in the center of the overlap

Process value	Intervals (ρ)		System Situations (ρ)		
T1 = 14	T1 = [0 20]	(0,6)	1	low ← low	(0,3)
	(10 60]	(0,4)	2	low ← filled	(0,3)
T2 = 15	T2 = [0 20]	(0,5)	7	filled ← low	(0,2)
	(10 60]	(0,5)	8	filled ← filled	(0,2)
Valve = left	Valve = (-∞ 0)	(1)			
T1 = 15	T1 = [0 20]	(0,5)	3	low – low	(0,25)
	(10 60]	(0,5)	4	low – filled	(0,25)
T2 = 15	T2 = [0 20]	(0,5)	9	filled – low	(0,25)
	(10 60]	(0,5)	10	filled – filled	(0,25)
Valve = 0	Valve = [0 0]	(1)			
T1 = 15	T1 = [0 20]	(0,5)	5	low → low	(0,3)
	(10 60]	(0,5)	6	low → filled	(0,2)
T2 = 14	T2 = [0 20]	(0,6)	11	filled → low	(0,3)
	(10 60]	(0,4)	12	filled → filled	(0,2)
Valve = right	Valve = (0 ∞)	(1)			

However, does it make sense to eliminate situations 4, 6, 7 and 9? To answer this question, a couple of system states can be examined. For example, observe the state corresponding to T1 = 19, T2 = 20 and the liquid flowing left through the valve (from T2 to T1). Because of the

level in T1 and T2, there is no certainty about whether the tanks are “low” or for “filled”. The labels “low ← low”, “low ← filled” and “filled ← filled” can be accepted to describe this process state. However the eliminated “filled ← low” indicates that the level in T1 (right) is greater than in T2 (left), which is clearly not the case. Besides, a liquid flowing from a tank with a low level to a filled one under isobaric conditions contradicts the laws of physics. The spurious situation 7 can, therefore, be eliminated from the situation table. Another critical point corresponds to the process state where T1 = T2 = 15 and there is no flow across the valve. Again, it is not easy to decide of 15 should correspond to a low level or to a filled tank, but in any case the level is the same. Therefore the labels “low – low” and “filled – filled” can be accepted without discussion, but the labels “low – filled” and “filled – low” of the eliminated situations 4 and 9 would lead to the false conclusion that the levels in the tanks are different.

Table 5.8: Membership behavior in randomly chosen points in the overlap

Process value	Intervals (ρ)	System Situations (ρ)
T1 = 12	T1 = [0 20] (0,8) (10 60] (0,2)	1 low ← low (0,08) 2 low ← filled (0,72)
T2 = 19	T2 = [0 20] (0,1) (10 60] (0,9)	7 filled ← low (0,02) 8 filled ← filled (0,18)
Valve = left	Valve = (-∞ 0) (1)	
T1 = 12	T1 = [0 20] (0,8) (10 60] (0,2)	3 low – low (0,64) 4 low – filled (0,16)
T2 = 12	T2 = [0 20] (0,8) (10 60] (0,2)	9 filled – low (0,16) 10 filled – filled (0,04)
Valve = 0	Valve = [0 0] (1)	
T1 = 15	T1 = [0 20] (0,5) (10 60] (0,5)	1 low ← low (0,05) 2 low ← filled (0,45)
T2 = 19	T2 = [0 20] (0,1) (10 60] (0,9)	7 filled ← low (0,05) 8 filled ← filled (0,45)
Valve = left	Valve = (-∞ 0) (1)	
T1 = 12	T1 = [0 20] (0,8) (10 60] (0,2)	1 low ← low (0,4) 2 low ← filled (0,4)
T2 = 15	T2 = [0 20] (0,5) (10 60] (0,5)	7 filled ← low (0,1) 8 filled ← filled (0,1)
Valve = left	Valve = (-∞ 0) (1)	
T1 = 19	T1 = [0 20] (0,1) (10 60] (0,9)	5 low → low (0,05) 6 low → filled (0,05)
T2 = 15	T2 = [0 20] (0,5) (10 60] (0,5)	11 filled → low (0,45) 12 filled → filled (0,45)
Valve = right	Valve = (0 ∞) (1)	
T1 = 19	T1 = [0 20] (0,1) (10 60] (0,9)	5 low → low (0,08) 6 low → filled (0,02)
T2 = 12	T2 = [0 20] (0,8) (10 60] (0,2)	11 filled → low (0,72) 12 filled → filled (0,18)
Valve = right	Valve = (0 ∞) (1)	

To confirm the previous analysis, six points are randomly chosen inside the overlap. Table 5.8 records the evaluation of the system rule at these points. Observe that the four eliminated situations (4, 6, 7 and 9) never go over $\rho = 0{,}25$, and in all the cases, interpreting these situations may lead to false conclusions about the system.

The study of this case demonstrates that eliminating situations with a small confidence value reduces the spurious situations in the situation table, which in turn, reduces the size of the SQMA model and facilitates the interpretation of process states. The described behavior is verified in all the analyzed cases, and no counterexample is found that could discourage the application of this situation reduction criterion. However, the modeler must take the final decision. In general, a reasonable threshold value for the rule confidence would vary between 0,5 and 0,8. If a situation is given a confidence lower than 0,5, assuredly its complements represent the real process state appreciably better.

5.1.3 Spurious situation reduction procedure

The following paragraphs describe an algorithm for the reduction of spurious situations based on rule confidence, according to the concept developed in the previous section. Since RI intersection is an operation defined using double-interval RI notation, the verification of the rule confidence must employ this notation as well. This would require processing each component and system rule twice (because of the double-interval) for each situation, which would complicate the situation validation process noticeably, jeopardizing the applicability of SQMA to complex technical processes. . In order to overcome this problem, a heuristic, based on the upper and lower approximation intervals, detects those cases where the rule confidence must be calculated, quickly and effectively, reducing the calculus with double-interval RI notation to its minimum. Considering the rule evaluation from the point of view of the crisp interval intersection with UAIs and LAIs (each RI intersection can be decomposed in the intersection of their corresponding UAI and LAI), three cases may arise:

- UAIs do not intersect each other
- LAIs do intersect each other
- UAIs intersect one another but LAIs do not

Each case corresponds to a particular case of confidence. When upper approximation intervals do not intersect each other, clearly there is no intersection between the corresponding lower approximation intervals. In this case, the rule is not fulfilled (RI do not intersect each other) and the situation is rejected with a rule confidence of zero ($C = 0$). Conversely, if the LAIs intersect one another, LAIs and UAIs will satisfy the rule. In this case, the RIs intersect one another completely, therefore a rule confidence of one is assigned ($C = 1$) and the situation is accepted beyond discussion.

In the third case, no immediate decision can be taken. The situation satisfies the rule when the UAIs are evaluated, yet the situation should be rejected if the LAIs are evaluated instead, because no intersection takes place (RI intersection in overlapping regions). In this case, the decision about accepting or eliminating the rule is taken upon the threshold value defined for the rule confidence (C_Req). If the currently analyzed situation surpasses this required rule confidence value, the rule is satisfied and the situation evaluation continues with the next rule. If the evaluation reaches a confidence that is equal to or lower than the one demanded by the modeler (C_Req), the situation is definitively rejected.

The diagram in Figure 5.3 represents the described algorithm. It examines the intersection between the Rough Intervals remaining at the end of the rule evaluation (RI1 and RI2) in their double-interval notation (UAI1:LAI1 and UAI2:LAI2 respectively). Here, the three described cases are illustrated, including the two variations of the third. An alternative algorithm can be implemented if the homogeneous form of SQMA system equation is considered. In this case, the rule is accepted if the resulting interval RI = RI2 - RI1 contains the value zero, i.e. the position of the resulting UAI = UAI2 - UAI1 and LAI = LAI2 - LAI1 relative to zero determine the acceptance of a rule. Both evaluation strategies are equivalent [Fröh94].

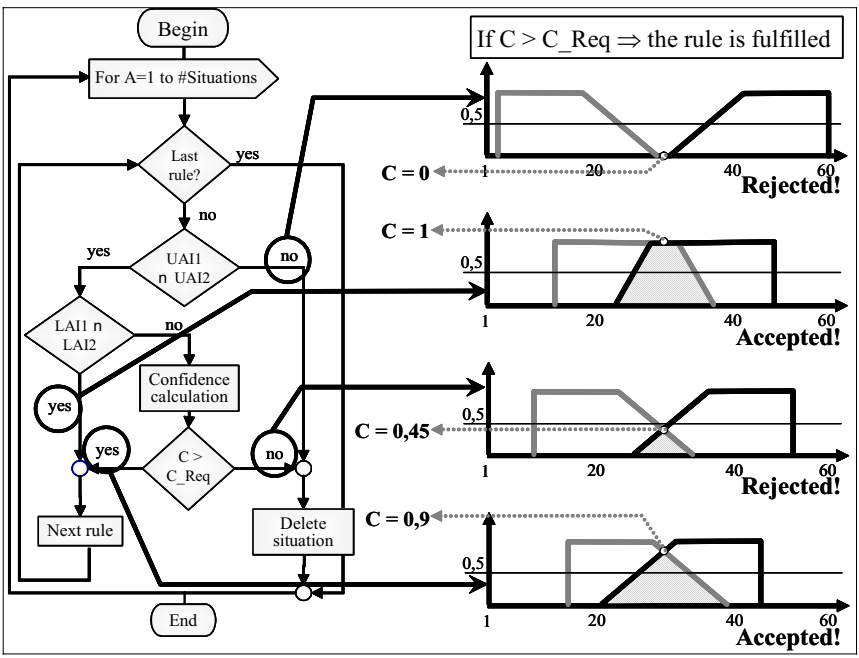

Figure 5.3: Spurious situation reduction algorithm based on rule confidence

5.2 Uncertainty by nondeterministic state transitions

The first element of the solution concept developed in the present research work is the Rough Interval concept, which represents vague concepts in SQMA. The second element, described in this section, introduces probabilities in SQMA to represent nondeterministic process dynamics in SQMA's transition matrixes. The introduction a further handling of uncertainty measures, namely conditional transition probabilities, in SQMA improves the uncertainty management, which enriches the information that the model can provide.

Representing and handling probabilities in SQMA relies on the development of transition models based on Stochastic Qualitative Automata (s. 2.3.3.1). This is possible, because SQMA transition models fulfill the homogeneous case of Markov's property (s. 2.2.1), i.e. a future process state can be determined (or at least delimited, in the case of a nondeterministic process such as SQMA) if the current process state is known. No process history is required. Consequently, SQMA transition matrixes remain constant as the process evolves.

In particular, the nondeterministic behavior of energy-storing components and their influence in the whole system is analyzed at the end of this section. Conclusions supporting the compact representation of SQMA dynamics models emerge from this study.

5.2.1 Probabilistic transition modeling

Upcoming situations are determined in SQMA's transition-based dynamics representation by looking for the situations in the row headers of the marked fields (target situations) in the column of the transition matrix corresponding to the source situation. This is equivalent to multiplying the entire transition matrix by a vector with one in the field corresponding to the current situation (or situations if RIs are used) and a zero elsewhere. The resulting vector will have a one in each reachable situation. This procedure matches the description of the Qualitative Automata (s. 2.3.3.1) and its mathematical representation in equation (2.11); therefore, dynamics modeling in SQMA accepts the same representation forms (transition matrix, automaton diagram, mathematical automata expression, evolution trees; s. Figure 5.4) and modeling principles of nondeterministic automata. This, and the fact that situations are similar to states in automata, suggests using nondeterministic automata for the modeling of transitions in SQMA.

Some of the transitions in the model are surely more frequent than others, but the SQMA model has no information about it. Qualitative Automata does not support the probabilistic management of uncertainty as it is introduced by the RI membership function and is demanded for the solution concept. Therefore, this representation is upgraded to a Stochastic Qualitative Automaton, as suggested by Lunze [Lunz95]. The resulting automata concept is *qualitative* because it represents abstract process states using qualitative descriptions (the SQMA situations)

and *stochastic* because of the probabilistic transition matrix and the probabilistic meaning of membership values in Rough Intervals.

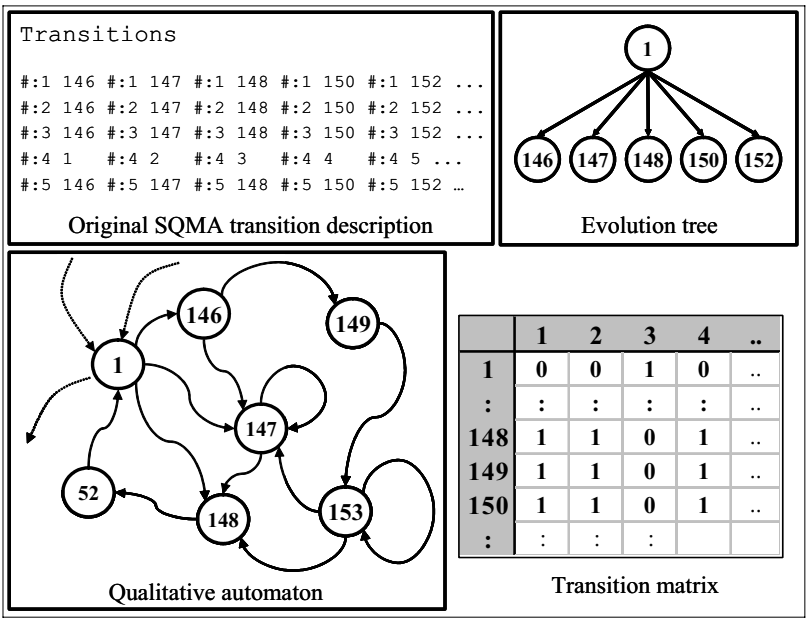

Figure 5.4: Various representations of situation transitions

In principle, a probabilistic transition matrix only differs from a conventional SQMA transition matrix in the value of its elements. The SQMA transition matrix only accepts ones or zeros, while the introduced probabilistic model accepts continuous values between 0% and 100%[5] (both limits inclusive) that correspond to the conditional probabilities of each situation transition. Methods to determine these conditional probabilities are available in literature [AvJe99],[Lunz95] and some are commonly used in the industrial and academic worlds. Nevertheless, this section describes an alternative method of approximating this model by adapting the SQMA transition matrix in the absence of a-priori probabilistic information.

The modeling procedure for the system dynamics is the same as the one used to model transitions in the conventional SQMA. With no a priori probabilistic information about the situation transitions, it can be assumed that each possible transition is equally probable, i.e. the inverse of the number of possible target situations establishes the conditional transition probability in each nonzero field of each column. This enables the conversion of transition matrixes into probabilistic transition matrixes, which is equivalent to using a uniform probability

[5] Or between zero and one, according to the notation chosen for probabilities.

distribution function to approximate the unknown probabilistic information. Figure 5.5 illustrates the conversion of an SQMA transition matrix (left) into a probabilistic transition matrix (right).

Sit	122	125	131	134
112	1	0	0	0
122	1	1	1	0
125	0	1	1	0
128	1	0	0	0
131	0	0	1	1
132	1	0	0	0
134	0	1	1	1

SQMA Transition matrix

Uniform probability distribution

Sit	122	125	131	134
112	25%	0	0	0
122	25%	33%	25%	0
125	0	33%	25%	0
128	25%	0	0	0
131	0	0	25%	50%
132	25%	0	0	0
134	0	33%	25%	50%

Probabilistic transition matrix

Figure 5.5: Conversion of SQMA transition matrix in probabilistic transition matrix

The interpretation of the probabilistic transition matrix is similar to the transition analysis in SQMA's conventional transition matrix. The system can go from any given situation in the column headers to any situation in a row header corresponding to a nonzero field in the same column. For example, the modeled system can transit from situation 122 (source situation) only to situations 112, 122, 128 and 132 (target situations). Since each transition is assumed equally probable, the probability of reaching any of these four target situations from situation 122 is 25%, to any other situation corresponds then 0% (compare with first column of the probabilistic matrix in Figure 5.5)

The described procedure does not modify the SQMA model of the system dynamics; actually, no information is included in the model by considering a uniform probability distribution. The behavioral information contained in the original transition matrix and in its probabilistic counterpart is the same. Nonetheless, using a priori probabilities improves the precision and descriptive power of SQMA transition matrixes, and assures the compatibility of existent SQMA models with the new concepts developed in this research work. Furthermore, this initial distribution of transition probabilities can be adapted online, based on observed transition frequencies. Such implementation would rely on historical data about the situation transitions to periodically update the probabilistic transition matrix, thus progressively enhancing the exactness of the SQMA dynamics model.

5.2.2 Considerations about energy-storing components

Transition matrixes are nondeterministic time-invariant representations of the system dynamics that can neither describe time-dependent nor cause-effect relationships. SQMA models control commands (information from actuators) and effects on the technical process (sensor data) at the same level. There is no differentiation between them in SQMA's situation concept. Therefore, no cause-effect relationship can be explicitly modeled in the transition model. All dynamic

information in the model depends on the storage and flow of energy in the system. Energy-storing components are those that are capable of storing and releasing energy according to rules of physics. This is the case, for example, of a tank that can neither be depleted nor overflow instantaneously, or a boiler, which cannot go from producing steam (with water over 100°C) to being available (e.g. with water about 25°C) before going through a cooling process (shutdown: hot → warm → cold).

Energy-storing components govern the process dynamic behavior in SQMA models[6], or what is the same, components without energy-storing capability cannot determine transitions between SQMA situations at system level. In fact, a component whose dynamic behavior does not follow specific rules (non-energy-storing component), can reach instantaneously and unrestrictedly any of its valid situations at any time. The total probability of the situations in these components for any particular combination of the others is always equal to 100%, therefore, non-energy-storing components do not participate in the determination of valid transitions at system level. For example, a three-way valve can change its state at any time, independently of the system dynamics; on the contrary, a heat exchanger cannot change from hot to cold without going through warm. Therefore, the heat exchanger, and not the three-way valve, determines the transitions of a system containing both components. The three-way valve imposes no restriction to the transitions in the system, whereas the heat exchanger certainly does.

Table 5.9: Case C of the Tank-Valve-Tank system

Case C: with two RIs			
T1.h	[0 20]	(10 50)	(40 60]
T2.h	[0 20]	(10 50)	(40 60]
Valve.dp	(-∞ 0)	[0 0]	(0 ∞)

Defining new intervals (Case C, Table 5.9) for the Tank-Valve-Tank system in Figure 5.2 can helps illustrate this analysis. Tanks T1 and T2 are energy-storing components; their level variables (T1.h and T2.h) are constrained to the following transitions:

- [0 20] → (10 50)
- (10 50) → (40 60]
- (40 60] → (10 50)
- (10 50) → [0 20]

Transitions [0 20] → (40 60] and (40 60] → [0 20] are excluded by the steadiness condition (s. 3.1.1 and 4.2.1) of the tanks' levels, which is consequence of the energy-storing capability of T1 and T2. As a result, the transition matrix of the system Tank-Valve-Tank accepts any

[6] Also SQMD (s. 3.1.3) takes advantage of this feature. SQMD uses numerical equations describing dynamic behavior in energy-storing components, in order to improve SQMA's description capability regarding the entire system.

transition between valid situations, where T1.h and T2.h change according to the described pattern. The analysis of the situation transitions in the component Valve does not contribute to the definition of the system transition matrix.

Thus, together, the probabilistic models of energy-storing components and the system situation table enclose all the information about the process dynamics in SQMA models. Even though an explicit all-comprising system transition matrix (considering all the components of the system) may be advantageous in cases where time response is an important criterion, dynamics models in SQMA can be reduced to its minimum by using only the transition matrixes of the energy-storing components, which is suitable for cases where the model size becomes more important.

5.3 Analysis of uncertainty management

A concept for the representation and handling of uncertainty in SQMA models is introduced in this chapter. It considers two sources of uncertainty: the uncertainty introduced in the modeling process by the boundary regions in Rough Intervals and the nondeterminism inherent to the transition-based representation of process dynamics in SQMA. This chapter describes and treats both sources independently.

The first uncertainty source, the boundary regions in Rough Intervals, causes the appearance of spurious situations in SQMA situation tables. These are an effect of the interval overlapping, the rough membership function and the nondeterminism of the intersection operator in interval arithmetic. With the removal of these spurious situations, SQMA models become more compact, and the situation-based description of process states become easier to understand and more consistent with process reality. A rule confidence value, related to the maximum membership value that a given situation can reach, may be used to eliminate these spurious situations.

The second uncertainty source concerns the SQMA transition matrixes, which are by nature nondeterministic. The precision of these matrixes can be improved if transition probabilities are represented instead of the original “transit”/“do not transit” transition descriptions. The use of transition probabilities in a Stochastic Qualitative Automaton allows projecting the measure of uncertainty provided by the rough situation membership to enrich the information delivered by the situation forecasting.

Introducing transition probabilities in SQMA, as with the introduction of RIs, does not imply the escalation of modeling expenses. The information available in an SQMA transition matrix, actually in the transition matrixes of its energy-storing components, suffices for the definition of an initial set of transition probabilities, which can be further adapted by the monitoring application. This ensures compatibility of the new concept with original SQMA matrixes. However, the employment of more precise methods for determining transition probabilities is not discouraged.

6 Enhanced situation-based process monitoring

The task of a process monitoring application is to follow the system operation, and detect deviations in its normal behavior as early as possible. In general, a deviation from the normal operation is determined if the process is in situations that are explicitly marked as abnormal or undesirable, or if the behavior observed in the technical process is not consistent with the previously calculated model. A situation-based process monitor compares the information from an SQMA model of the correct process behavior with online data from the technical process (sensors and actuators), in order to make opportune safety-critical decisions.

The following sections describe an approach for process monitoring that integrates the Rough Interval concept in the SQMA monitoring (3.1.3) and combines the resulting approach for the analysis of the current process state with a Stochastic Qualitative Automaton (2.3.3.1) for the determination of reachable situations. The combination of these techniques enables the management of vagueness and uncertainty in SQMA, which improves process monitoring precision and SQMA's ability to deal with complex technical systems. Further features of this monitoring approach are:

- Precise identification of the current state.
- Early detection of transitions and critical situations
- Probabilistic forecasting of reachable situations

6.1 Monitoring of complex systems

Memory consumption is a very sensitive matter when dealing with SQMA models. In many applications, the amount of storage capacity, not the computing power, restricts the implementation of a process monitoring applications. Memory units are usually more expensive and require more physical space than the processing units performing an equivalent task. This is the case, for example, in embedded products and complex mass production systems such as cars and airplanes.

The definition of an adequate structure for the SQMA monitoring concept must consider storage needs as well. Section 3.1.3 describes the basic structures of a process monitoring application based on SQMA models. These structures were defined considering scenarios determined by the available computing resources. Adopting a configuration as in basic structure A (s. Figure 3.2, top) for the monitoring may imply time responses that are too low for a real-time application such as safety-critical process monitoring, because it requires many time consuming calculations online.

Yet, basic structure B is also unsuitable for monitoring complex technical systems. The construction of a system transition matrix as the one required for the process monitoring in basic structure B, is often impracticable. Combining all valid system situations, and filling each field by verifying the single component transitions, produces the system transition matrix. The problem is that sometimes the SQMA modeling tools cannot complete this procedure, because the number of situation combinations at system level is too huge to fit in the memory available for the model. This geometrical explosion takes place because the combinations in the system transition table increase in a square ratio with the number of situations in the system situation table, which is already very large.

The solution is to adopt an intermediate monitoring configuration between the structures described in 3.1.3. The new structure uses a centralized structure (as basic structure B) to analyze the current process state, and a distributed configuration (similar to basic structure A) to analyze the process dynamic behavior to determine reachable situations. Both tasks can be performed independently, which allows using different time basis and even parallel computing processes. Figure 6.1 illustrates this hybrid monitoring structure.

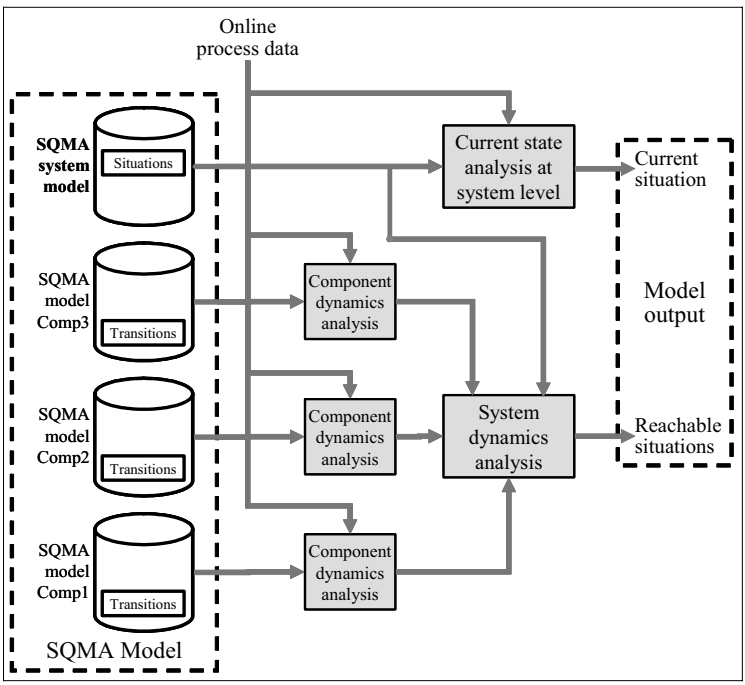

Figure 6.1: Hybrid SQMA monitoring structure

Observe in Figure 6.1 that the current state remains centralized at the system level, which assures an optimal time response by the detection of critical situations. On the other hand, a situation forecasting using component transition matrixes instead of a single system transition matrix, noticeably increases the applicability of SQMA monitoring to complex technical systems. Computing the transition matrix at system level is no longer necessary; the monitoring application generates and evaluates a reduced system transition table for the time t, based on the observed component situations and the component transition matrixes. The reachable component situations are then combined at system level. Since component models do not consider the system rules, the combination of component situations may deliver spurious situations. Therefore, the resulting component situation combinations are validated against the system situation table.

Using the structure shown in Figure 6.1, all the information required for the process monitoring is available, even after eliminating the system transition matrix. This hybrid structure is suitable in applications with limited memory resources or when the system complexity and the demanded application precision delivers a system transition matrix, which is too huge to be calculated or effectively used.

6.2 Identification of the current process state

The identification of the current process state is the first task in the online supervision of technical processes. It consist on recognizing and describing the current process state, based on a model of the correct process behavior, and data acquired periodically from the running process. The process state identification is the basis of the analysis of the current process state, which is the first of the two processing stages described in the monitoring structure introduced in the previous section.

This section introduces the SQMA state observer concept, which integrates Rough Intervals in an SQMA monitoring structure centralized at system level, for the situation-based identification and description of the process state, making possible its further analysis and projection.

6.2.1 State observer procedure

A situation-based monitoring application searches system situations in the process model matching the current process state. Since RI can represent soft transitions and overlapping of characteristics in SQMA models, several system situations may be found that correspond to the current state. Accompanying membership values determine, which situations correspond unequivocally with the observed process state, and in which cases, this correspondence is only a possibility. The described procedure, illustrated in Figure 6.2, describes an SQMA state observer, step-by-step.

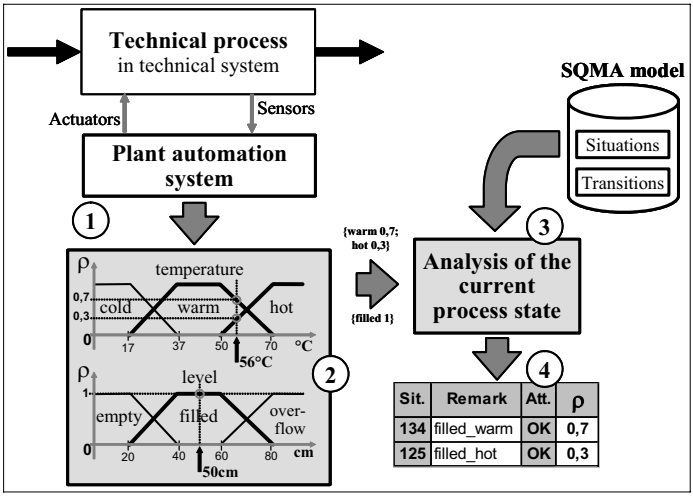

Figure 6.2: Process monitoring with SQMA and RI

(5) The SQMA state observer works in a hierarchical level over the process control system. It reads periodically from the running technical process, more exactly, from its control system, the measured variables and control commands required for the state identification.

(6) These values are compared with the previously defined Rough Intervals. If a measured value hits a lower approximation interval (LAI), as in the case of the level in Figure 6.2, the corresponding qualitative variable takes the qualitative value of this LAI ("filled" in this example) with a rough membership of one (precise state identification). If, on the contrary, the measured value lays in the overlap of two upper approximation intervals, as it happens with the temperature, the qualitative variable assumes the qualitative values from both UAIs ("warm" and "hot" in the example) simultaneously, with complementary membership values. Section 6.2.2 describes the calculation of the RI membership in detail.

(7) The component situations corresponding to the possible Rough Interval combinations are retrieved from the component situation table in SQMA model. Component situations are then combined and validated against the system situation table. For the example in Figure 6.2, two qualitative values ("warm" and "hot", with $\rho = 0{,}7$ and $\rho = 0{,}3$ respectively) for temperature and one level qualitative value ("filled", with $\rho = 1$) for level produce two possible combinations, and both of them correspond to valid system situations.

(8) The observed situations are displayed with their corresponding attributes ("danger", "warning", "ok" ...), labels (such as "empty_increasing", "filled_warm" or "overrun") and situation membership values. The SQMA state observer may display an entire group of situations (instead of just one, as it happens with conventional SQMA models), ordered

according to their membership values. If no valid situation is found for the observed values, a notice must be issued to indicate that the observed behavior is abnormal or faulty. For the example, it results:

Sit. #134: filled_warm, Normal (Ok), with ρ = 0,7

Sit. #134: filled_hot, Normal (Ok), with ρ = 0,3

Having multiple system situations to describe the current process state does not imply a difficult reading. Quite the opposite, the delivery of multiple situations allows a more detailed description of the observed state. The following example illustrates the improvement in the descriptive power of situation-based process monitoring through the identification of multiple situations for each process state. Figure 6.3 shows two possible outputs from the SQMA state observer corresponding to the same process state. The first case (top) uses an SQMA model, where only crisp intervals were defined. At most, the plant operator would describe this information as: *"The system is in a normal situation. The tank is filled with warm liquid and has nominal inflow and outflow"*.

Sit.	Inflow	Level	Outflow	Temp.	Attribute
	85ml/sec	*50cm*	*70ml/sec*	*56°C*	
134	Nominal	Filled	Nominal	Warm	Ok

SQMA using crisp intervals

Sit.	Inflow	Level	Outflow	Temp.	Attribute	ρ
	85ml/sec	*50cm*	*70ml/sec*	*56°C*		
134	Nominal	Filled	Nominal	Warm	Ok	*0,42*
131	High	Filled	Nominal	Warm	Warning	*0,28*
125	Nominal	Filled	Nominal	Hot	Danger !	*0,18*
122	High	Filled	Nominal	Hot		*0,12*

SQMA using Rough Intervals

Figure 6.3: Comparison of situation-based monitoring with and without RIs

The second table (bottom) corresponds to an SQMA model with Rough Intervals. Now the same process state could be described based on the identified situations as*: "The liquid filling the tank is moderately hot* (between "Warm" and "Hot"). *The tank inflow is a little high* (between "High" and "Nominal") *while the outflow is nominal; the inflow is in any case greater than the outflow, therefore, the level is increasing. With that, the system transits towards a dangerous state"*. The descriptive power of the second description of the process state (using RIs) is visibly superior to that of the first case (using crisp intervals), even though the number of situations in both models (and consequently their sizes) is the same.

The enhanced information is a consequence of the interval overlapping and rough membership functions. A variable defined over n Rough Intervals (which defines the size of the model) contains information distributed in n lower approximation intervals plus (n-1) overlapping areas,

i.e. in (2*n-1) sections. Since each of these (2*n-1) sections can be independently perceived and interpreted, the resulting model delivers almost twice as much information as an SQMA model using crisp intervals, which only can identify n different qualitative values. For instance, “Hot” (with $\rho = 1$), “Warm” (with $\rho = 1$) or both qualitative values simultaneously, with different but complementary ρ values, correspond to three different states. The third case could be described as “moderately Hot”, “between Hot and Warm” or “almost Hot”, depending on the relationship between the membership values and given linguistic conventions. Vague linguistic expressions can be used, which are closer to the human way of description.

6.2.2 Rough Interval membership computation

One of the strengths of the Rough Interval concept is that it requires no additional parameters for the representation of the rough membership function. Rough membership is calculated in real time during the process monitoring based on an SQMA model using RIs. Figure 6.4 represents the three RIs defined in a process variable (temperature): [0 37), (17 70) and (50 ∞). These intervals correspond to the qualitative values “cold”, “warm” and “hot” respectively. This partitioning originates the membership function in Figure 6.4, which is used to demonstrate the online determination of the Rough Interval membership (ρ).

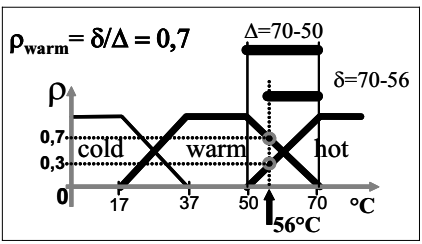

Figure 6.4: Computation of RI membership

Assuming a real temperature value of 56°C, two RIs (“warm” and “hot”) can be found in the SQMA component model that corresponds to the current process state. If just one RIs is found, this RI gets the total membership ($\rho = 1$) in this variable. These two RIs, however, contribute to the current component situation with different but complementary weights. The respective membership value, which is calculated based on the relationship depicted in Figure 6.4, symbolizes the contribution of each RI to the respective component situations. A real temperature of 56°C corresponds to a “warm” situation with a rough membership of 0,7. The accompanying qualitative value, “hot”, gets a rough membership value of 0,3 ($\rho_{hot} = 1 - \rho_{warm}$). Calculating the rough membership value of 56°C to the RI “hot” delivers the same results:

$$\rho_{hot} = (56 - 50)/(70 - 50) = 0{,}3 \quad \Rightarrow \quad \rho_{warm} = 1 - \rho_{hot} = 1 - 0{,}3 = 0{,}7$$

The qualitative values associated with a component variable must be probabilistic handled as independent random variables, because SQMA is a noninteracting method, i.e. it does not model interdependencies between variables and components. Hence, the compound probability of process variables having the qualitative values established by a given situation is calculated by multiplying the single occurrence probabilities for each variable. That means that, since rough membership has a probabilistic meaning, the component membership value results from the multiplication of the RI membership of the single variables that compose the component. System situation memberships are then calculated as the product of the corresponding component situation memberships, i.e. through the multiplication of all the RI memberships in the system. In the example of Figure 6.2, the rough membership of level = "filled" is one, therefore the situation membership values for the above-calculated memberships in the temperature variable result:

$$\rho_{134} = \rho_{filled} * \rho_{warm} = 1*0{,}7 = 0{,}7$$
$$\rho_{125} = \rho_{filled} * \rho_{hot} = 1*0{,}3 = 0{,}3 \qquad \Rightarrow \Sigma(\rho) = 1$$

The example of Figure 6.3 involves computing more variables for each situation:

$$\rho_{134} = \rho_{nominal} * \rho_{filled} * \rho_{nominal} * \rho_{warm} = 0{,}6*1*1*0{,}7 = 0{,}42$$
$$\rho_{131} = \rho_{high} * \rho_{filled} * \rho_{nominal} * \rho_{warm} = 0{,}4*1*1*0{,}7 = 0{,}28$$
$$\rho_{125} = \rho_{nominal} * \rho_{filled} * \rho_{nominal} * \rho_{hot} = 0{,}6*1*1*0{,}3 = 0{,}18$$
$$\rho_{122} = \rho_{high} * \rho_{filled} * \rho_{nominal} * \rho_{hot} = 0{,}4*1*1*0{,}3 = 0{,}12 \qquad \Rightarrow \Sigma(\rho) = 1$$

This membership calculus guarantees the fulfillment of the rough membership properties enunciated in Table 2.1. For example, no value greater than one or lower that zero can result of this calculus. Additionally, the described mechanism for the aggregation of component and system membership, which is based on compound probability, satisfies the rule $\rho^B{}_{X\cap Y}(z) \leq \min(\rho^B{}_X(z), \rho^B{}_Y(z))$, because the product of numbers between zero and one results in a value, which is always lower than the minimum factor in the multiplication. The fulfillment of the remaining properties in Table 2.1, such as the RI complementarity and the conjunction of RIs based on total probability, can be also verified.

6.3 Detection of transitions and critical situations

The goal of the situation-based monitoring application developed in this research work is to improve the safety of industrial installations (the technical system). Critical requirements on the process monitoring are then, the detection of dynamical trends in the system (transition characterization) and early recognition of critical situations. This section describes the performance of the new process monitoring concept by accomplishing these requirements.

6.3.1 Analysis of dynamic process trends

Using Rough Intervals allows reaching a degree of detail about the process behavior, which greatly exceeds the information delivered by SQMA using crisp intervals, where the notion of transition is inapplicable to the current state. The observation of multiple situations per process state, not only supports the formulation of vague but also detailed linguistic descriptions of a static view of the system, it also provides information about dynamic trends in it. Multiple system situations describe a dynamic system, which is changing from one state to other. Following changes in the situation memberships, it is possible to be aware of the direction and rapidness of the change of state in the system.

Because of the rough membership function, the RI concept improves SQMA precision in the transition areas. Memberships allow discerning the position of the process variables with respect to intervals defined in the SQMA model (s. 4.1). The information about the process dynamics derives from this new SQMA capability.

If only energy-storing components (s. 5.2.2) are considered, the list of current situations signals already, those situations that, as a matter of fact, can be successors of the current ones in the immediate time-period afterwards (imminent future, within the next sampling period). This works as a filtering mask, which reduces the monitoring uncertainty in the short term. This filtering mask is determined by the transition side (or direction) of each changing variable. If a variable is inside of an LAI it can be assumed that within the next sampling period it will remain in this area. In case other RI is reached, its corresponding membership must be small enough to allow discarding this possibility. On the contrary, a variable in the overlapping between two RIs can go definitively to any one of the two RIs (go to their LAIs) or remain in the overlap. The resulting combination of these intervals contains the same situations observed for the current process state.

The described operation is more intended as a temporal magnifying glass (to highlight imminent situations) than as a forecasting procedure, which is the matter of section 6.4. For example, if there is a dangerous situation among the situations describing the process state, corrective actions must be immediately deployed to move the process to the nearest safe situation.

The augmented degree of detail about the dynamic behavior of the process also enriches the description of the identified process state. Figure 6.5 resumes the example of Figure 6.2. The situations observed for the current process state (top) can be read together as: *"the tank is filled with a moderately hot liquid"*. This information only provides a static view the process. However, the fact that multiple situations have been detected, describes that the system is transiting between to clearly defined situations: "filled_warm" and "filled_hot". It is even obvious that changes in the variable temperature drive this dynamic change in the system.

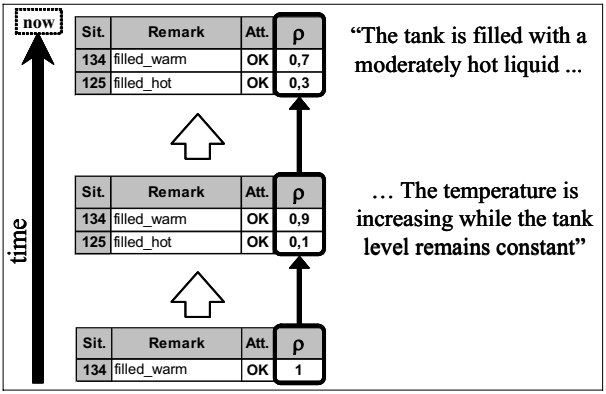

Figure 6.5: Trend analysis

Considering successive views of the SQMA state observer's output can enhance the previous analysis. Figure 6.5 shows three temporally subsequent readings of the process state. The analysis of the rough membership value evolution in each situation can indicate the direction and speed of the situation transition. This allows complementing the information delivered by the system: *"The tank is filled with a moderately hot liquid. The temperature is increasing while the tank level remains constant"*. SQMA with RI can detect situation transitions and thus become aware of instantaneous trends in the process run, improving the characterization of the current process state.

6.3.2 Early recognition of critical situations

The early recognition and interpretation of critical process situations in a running technical process is of main importance for assuring the safe operations. SQMA's situation concept is an effective way of supporting this task. SQMA models describe critical situations, such as undesirable process states and dangerous operation conditions, using system situation attributes, which facilitates their recognition and correct interpretation during the process monitoring. Nonetheless, if only the observation of the current process state is considered, SQMA using crisp intervals merely detects a critical situation once the process has reached it. The identification of undesirable process states or dangerous operation conditions depends on the feasibility of having all the process variables inside of the critical intervals, i.e. those associated with critical situations in the model.

The new concept for situation-based process monitoring using Rough Intervals improves the SQMA capability of detecting critical situations. As it was introduced in the discussion about the filtering mask in the previous section, using Rough Intervals critical situations are detected very early. They appear at the beginning together with other situations, and characterized by a

low membership value. Additionally, it has the advantage that, as soon as an undesirable situation appears among the solutions, its neighboring states appear as well, pondered consistently with their relative certainty (represented by ρ). This supports the immediate definition of correcting actions, such as taking the process to the nearest safe situation (that with highest membership value), which is also displayed as part of the current process state

To support the evaluation of operative conditions in a technical system, it is possible to follow closely the evolution of the situation membership value of critical situations, from the moment they first appear. Depending on the criteria employed by the design of the SQMA model and the safety requirements of the monitored process, a tolerance level can be defined for the membership of such critical situations, under which no preventive or corrective action is taken on the process. Several intervention scenarios may be defined as well, based on these situation membership values and situation attributes ("undesirable", "dangerous", "warning", etc.).

Another case is the detection of "false" impossible situations. SQMA explicitly eliminates impossible situations from the SQMA model. However, faulty sensors can read, erroneously, such situations. This is the case of water flowing out of a full tank. The monitoring system can fail if a faulty level sensor signals an empty tank, because this situation does not exist in the corresponding situation table (no water can flow out of an empty tank). An alternative is describing these impossible system situations in the SQMA model. Doing this, the SQMA state observer will display the modeled impossible situation, instead of issuing a failure. A matching situation attribute (e.g. "impossible", "not allowable", etc.) may be used to separate these cases from the traditional "danger", "warning" and "normal" situations.

On the other hand, considering each single obstruction, leak or sensor defect is impracticable. Even if each possibility can be pondered, including them in model as additional variables would cause the fast growth and eventual geometrical explosion of the SQMA system model. Besides, the explicit monitoring of failures goes beyond the functionality described for the online process monitoring (s. 2.1.3) into the process diagnosis. In many applications, it is advisable not to consider failures in the monitoring model, but to detect "suspicious" cases (such as not finding a matching system situation in the model) online, and to activate a separate diagnose procedure using a fault model (compare Figure 2.4). Grzesiak investigated the feasibility of using a separate SQMA model for fault diagnosis [Grze03]. He remarked the convenience of using Rough Intervals for the fault model, because the additional information provided by the membership values facilitates the probabilistic failure isolation and the associated decision taking. Section 8.3.2 presents a brief description of this study.

6.4 Stochastic forecasting of reachable situations

A precise analysis of the current process state is generally insufficient to assure the safe operation of technical systems. The in the previous sections introduced procedure, may certainly

support the efficient recognition and correction of critical situations in the system, however it would be better to avoid reaching such critical situations at all, for example. This requires interpreting the long-term system behavior, i.e. the forecasting of those system situations, which can be reached from the current process state.

Transitions represent in SQMA the dynamic system behavior. Given that the transition space grows in relation to the number of situations with a geometric progression of ratio two, the construction of the system transition table not always is possible. Consequently, SQMA transition matrixes are usually available only for components. This problem demands the reformulation of the conventional SQMA forecasting approach to use component transition matrixes instead of a single system transition matrix.

The new approach also introduces probabilities in the calculation process. It adapts the Stochastic Qualitative Automata technique, originally suggested by Lunze [Lunz95], for it use with SQMA. This Stochastic Qualitative Automata is called Qualitative because automaton states are described using qualitative descriptions (the SQMA component situations) and Stochastic because of the probabilistic component transition matrix and the probabilistic meaning of membership values in Rough Intervals. This section described this new approach for the forecasting of reachable situations, which enriches the information displayed during the process monitoring.

6.4.1 General approach: *Stochastic Qualitative Automata*

The use of Rough Intervals in SQMA for the analysis of the current process state provides the identified situations with a measure of their certainty: the system situation rough membership value. Membership in rough sets, and therefore in Rough Intervals, has a probabilistic meaning, contrary to what happens with the fuzzy membership. The membership values can enhance the accuracy of the information given to, and therefore produced by, the situation forecasting procedure of the monitoring application.

This approach uses probabilistic transition models, instead of SQMA's conventional "Yes/No" (or one/zero) transition matrixes, to facilitate handling the system situations in probabilistic and stochastic terms. This probabilistic transition model and the measured membership values make possible the use of a stochastic qualitative automaton (s. section 2.3.3.1) for a more accurate situation forecasting. However, in this case, the starting point of the automata (the current situation) is not as precisely defined as in traditional stochastic automata. For this reason, there can be several complementary nonzero entries in the vector S(0) of equation (2.11). The consequence is the development of alternative evolution trees for the system (s. Figure 6.6), which need to be consolidated in one definitive tree using the rough membership value of their

respective initial (current) situations as weighting factor. The set of current situation membership values determines the vector S(0).[7]

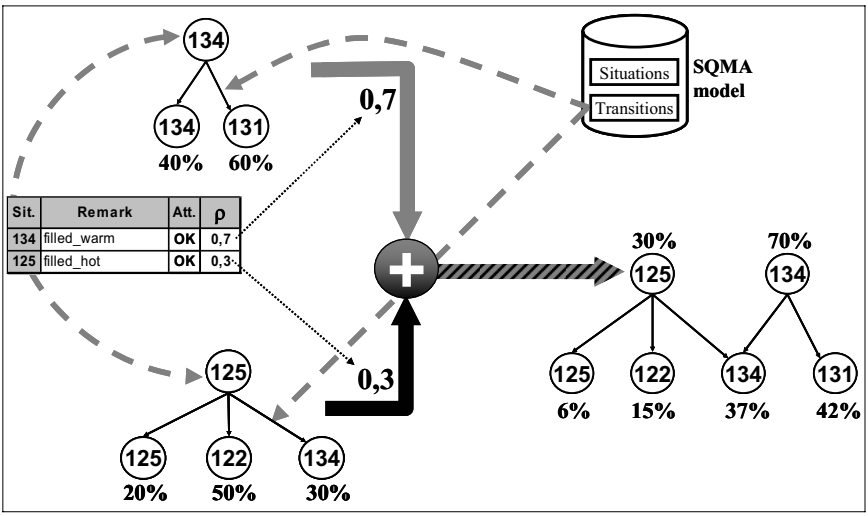

Figure 6.6: Situation-based Stochastic Qualitative Automaton

Figure 6.7 illustrates this transition analysis procedure, resuming the example presented in Figure 6.3. A first step is to define a reduced stochastic transition matrix (or diagram) containing the current situations and those that can be reached from these. From the currently observed situations and this reduced transition matrix, all the different transition alternatives are isolated and the corresponding probability computed (top of the figure). The described procedure forecasts possible process states with their occurrence probability. The probability of particular instances of situation labels or attributes can be also calculated. The example in Figure 6.7 shows an initial process state described by four situations, and a reduced probabilistic transition matrix (top). These are used in the automaton equation (center) to determine which situations can follow the current one. The resulting probabilistic information can be additionally grouped by situation attribute. This classification/filtering operation may support triggering correcting measures opportunely, such as to shut the plant down if the total danger probability exceeds 30%. The system operator must take these decisions, after interpreting the delivered information.

[7] In the described example, the symbol % and values between 0 and 100 were used to indicate transition probabilities, whereas ρ with a value between 0 and 1 indicates current rough memberships. Despite the fact of having separate uncertainty sources, both cases represent probabilities. Although both representations are equivalent, different symbols and notation systems are used in each case, as a way of supporting their discrimination. This convention will be kept in following developments.

Sit	122	125	131	134
112	30%	0	0	0
122	10%	50%	60%	0
125	0	20%	10%	0
128	30%	0	0	0
131	0	0	20%	60%
132	30%	0	0	0
134	0	30%	10%	40%

x

Sit	ρ
134	0,42
131	0,28
125	0,18
122	0,12

=

Sit	%
112	4%
122	27%
125	6%
128	4%
131	31%
132	3%
134	25%

$$\mathbf{T}_{7x4} \quad \mathbf{x} \quad \mathbf{P}(k)_{4x1} \quad = \quad \mathbf{P}(k+1)_{7x1}$$

Sit	%	Remark	Attribute
131	31%	filled_warm_increasing	Warning
122	27%	filled_hot_increasing	Danger!
134	25%	filled_warm	OK
125	6%	filled_hot	Danger!
112	4%	filled_warm_decreasing	OK
128	4%	filled_hot_decreasing	OK
132	3%	overflow_hot	Danger!

Reachable situations

Danger !	36%
Warning	31%
Ok	33%

Figure 6.7: Example of situation forecasting

The described procedure relies on a unified system transition matrix, which is not always available, as it was explained in previous section. To break this dependency, the solution concept for the forecasting of reachable situations uses a dynamic model of the system, distributed at component level. Knowing a system situation determines knowing the comprising component situations. These component situations are projected using the component transition matrixes to determine reachable component situations and their probabilities. These situations are then combined at system level. The probability of each combination is calculated multiplying probabilities of all its components. The predicted system situations are finally validated against the system situation table. After eliminating invalid system situations, the system situation probabilities have to be renormalized. This produces a list of possible future situations with their respective occurrence probabilities.

6.4.2 Optimized approach: *Filtering impossible transitions*

The probabilistic forecasting of reachable system situations with SQMA using RIs realizes a qualitative stochastic automaton using the system situations at a given time and the probabilistic component transition matrixes. However, since probabilistic component transition matrixes are nondeterministic, several system situations may be found that are reachable from the current process state. By working with real systems, the list of reachable situations is usually too large to be handled in real-time. In this section, a new automaton concept is introduced that is implemented based on the systematic elimination of unreachable situations from the system

situation table, and not on the progressive development and combination of component transition trees.

This concept, illustrated in Figure 6.8, does not verify each component transition, because only energy-storing components influence the dynamic system behavior. Therefore, only energy-storing components contribute to the compound probability of each one of the valid situations that a system can reach at a given time. Based on this fact, non-energy-storing can be eliminated from the dynamical analysis. Their well-known probabilistic contribution can be included afterwards as a multiplying constant by displaying the forecasting results.

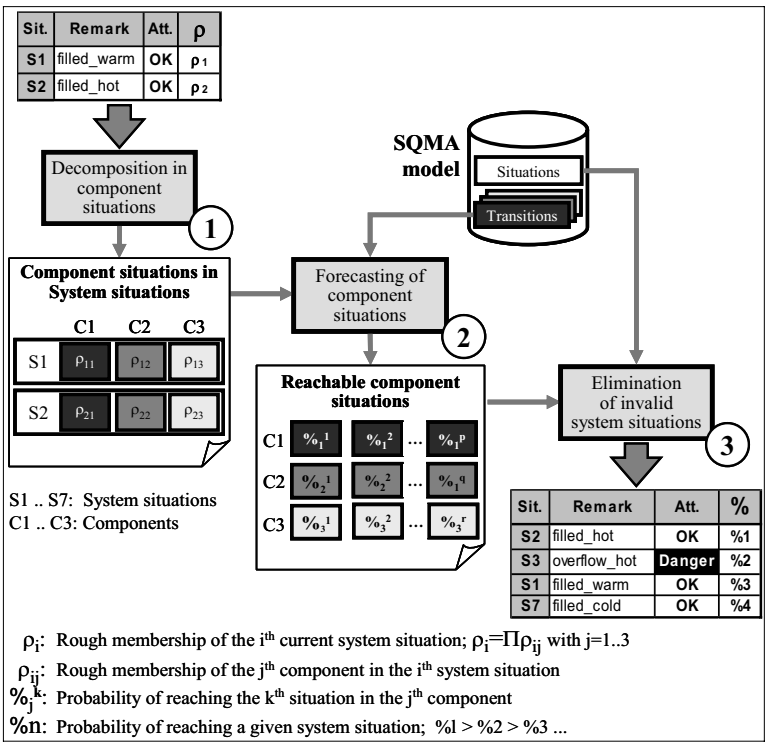

Figure 6.8: Schematic representation of transitions forecasting

Another important factor to be considered by the formulation of this optimized SQMA forecasting approach is that, even though many components can be eliminated from the dynamical analysis according to the above-discussed premise, the combinatorial situation space generated with the energy-storing components is usually appreciably bigger than the table of valid system situations. For example, a vessel with three connections can be precisely represented by about 100 situations. A system with three so described tanks (they can represent

the two MIC tanks and the water tank in Bhopal, compare 1.1) generates about 47.000 situations. Assuming that, each tank can transit at a given time to 60, 50 and 30 situations respectively, it is necessary to review and validate all 90.000 possible combinations, which is about twice the total number of valid situations.

Considering the last argument, the new strategy to forecast reachable situations is based on the successive elimination of those system situations (#3 in Figure 6.8) containing component situations (determined in #2 in Figure 6.8), which cannot be reached from the current process state (decomposed in #1 in Figure 6.8). The conventional forward-chaining combinatorial approach of the automata is in this case neglected in favor of an approach based on situation reduction, which is frequent in situation-based applications. Additionally, this valid-situations-reduction approach assures that invalid situations will be excluded in the solution space.

System situation probabilities are calculated in this approach during the situation verification, by multiplying the probabilities of the component situations that compose the validated system situation. Since not all the component situation combinations will be represented in the final list of reachable system situations, the resulting situation probabilities must be renormalized at the end of the forecasting process.

6.5 Analysis of the enhanced monitoring approach

A monitoring approach was introduced in this chapter, which consists of two main parts: the online analysis of the current process state and the analysis, in a second step, of those situations that can be reached from here (s. Figure 6.9).

Through the online analysis of the current process state, with SQMA using RIs it is possible to identify the condition and behavior of the technical process at any time. This information is qualitatively represented by a single situation or by a set of situations contributing to the characterization of the process state in the extent indicated by the respective membership values. These situations and their attributes make possible the identification of dangerous, faulty and not allowable process behaviors and conditions, more efficiently than by using crisp intervals. Additionally, now it is also possible the recognition of transitions through the coincidence of system situations. The detected changes of state can be traced to the very process variables that cause it. Even the instantaneous direction and intensity of these transitions can be appraised by following the evolution of the system membership values. Transition recognition and characterization is a result of the interval overlaps and the membership functions, which improve the description of the process state.

The analysis of the current process situation using SQMA with RI is more complicate than by employing crisp intervals, because of the multiple situations and the rough membership values that may result. The analysis based on crisp intervals is very simple and limited to the static

process behavior. The new approach also supports this interpretation level; then again, it encourages the realization of short-term dynamic analyses based on the additional information that is made available. That is to say, any additional complication by the analysis of the current process state is directly associated with the exploitation of RI capabilities.

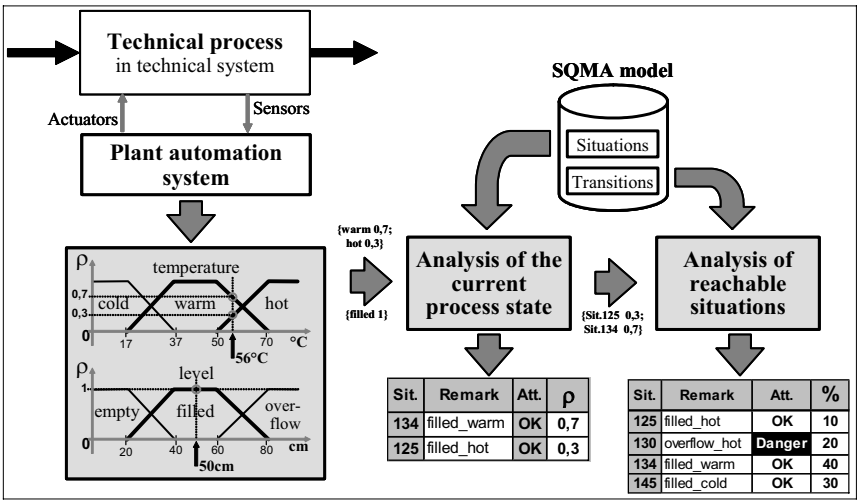

Figure 6.9: Integrated monitoring concept

The second element of the SQMA monitoring application is the probabilistic forecasting of reachable situations. Integrating probabilistic information in SQMA does not complicate the forecasting procedure gravely, yet it has the advantage of improving the quality of the delivered information. Transition probabilities improve the precision of the forecasting, and provide new criteria for the filtering, grouping and classification of the displayed information, facilitating the reading of the information and thus its usefulness.

A forecasting procedure based on Stochastic Qualitative Automata had to be optimized, because of the great number of resulting combinations of reachable component situations and the multiple system evolution trees that must be checked. Two ideas served as basis for this optimization:

- Components without dynamical restrictions (such as an open/close valve) do not contribute to the system dynamics.
- The number of possible combinations of reachable component situations is usually larger than the total number of valid situations in the system.

This optimized forecasting procedure computes all the situations that each energy-storing component can reach. Then those system situations containing unreachable component situations are eliminated from the solution space.

Another important asset of the new monitoring application is the introduction of a new schema for the structure of an SQMA monitoring application (s. Figure 6.1). This hybrid structure, with a centralized and single system situation table and component transition matrixes disseminated on the different components, is added to the two original structures introduced by Fröhlich [Fröh97]. This expands the applicability of SQMA to new areas, where the memory consumption is a critical criterion, or where the process complexity or the required application precision does not allow the application of the other two structures.

Once the representation and management of vague and uncertain information in the SQMA model and the use of this model for process monitoring have been described, the formulation of the solution concept is complete. This concept is evaluated in the next chapter based on its application on a case study: the online monitoring of a Three-Tank System.

7 Situation-based monitoring of a Three-Tank System with Rough Intervals

In order to prove the correctness and evaluate the efficiency of the formulated solution concept, the different elements that compose this concept were prototypically implemented on a nonlinear process consisting of tree tanks (Figure 7.1). This section, after describing this process in detail, systematically develops a situation-based qualitative model, considering Rough Intervals and a distributed probabilistic transition matrix. Once developed, the model is implemented as a relational database, which supports the situation-based process monitoring. This chapter closes with a brief discussion of the observed behavior, from the point of view of the concept requirements formulated in section 3.4.

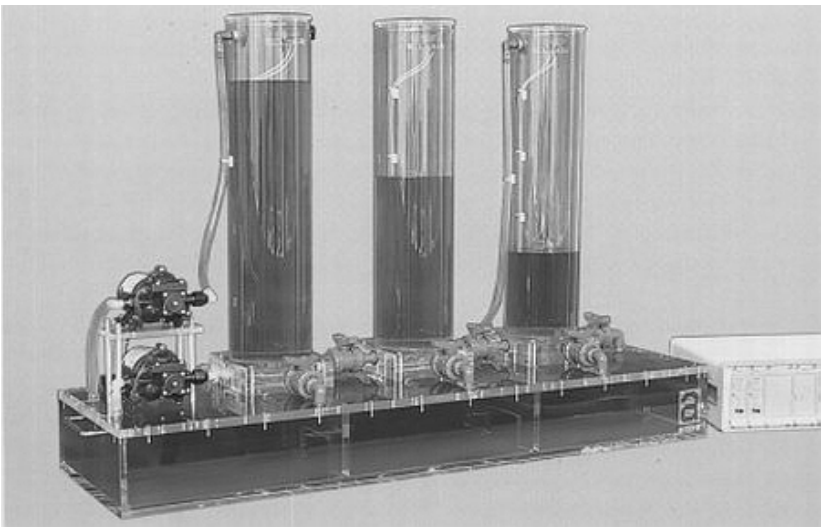

Figure 7.1: Pilot process “Three-Tank System”

The Three-Tank System is a laboratory implementation of one of the most frequent technical processes in industrial production complexes. Different configurations of pumps, tanks, pipes and valves are found as part of chemical and physical processes or in storage and distribution yards. For example, the chemical plant failure at Bhopal (s. section 1.1) can be reproduced, considering the problem as a Three-Tank System: One operation tank filled with MIC that cannot be discharged into a second tank because this is already full of MIC. Although water must never be mixed with MIC to avoid dangerous emanations, cooling water is leaking into the

first tank from a reservoir, which can be represented by another tank. Ergo, a system of three tanks, a set of pipes, valves, pumps and leaks, and safety conditions (bridged safety devices, overloaded reserve tank, etc.), that should not be present simultaneously.

7.1 Three-Tank System pilot process

The "Three-Tank System" (3TS) pilot process demonstrates the technical realization of a nonlinear MIMO system with two manipulated variables, two controlled variables, an uncontrolled system variable and several parameters to simulate loads and perturbations.

Three Plexiglas cylinders are serially connected via pipes. The liquid flowing out of the right tank (see picture) is collected in the large pool below. The pumps supply both outermost tanks with the fluid from the pool. All three tanks have pressure gauges to measure fluid levels. A digital regulating unit controls the volume flow of both pumps so liquid levels of the outermost tanks can be individually set. The middle tank level is always a resulting uncontrolled value. The connection channels have additional valves that can be manually operated to simulate flow, blockage and leaks. Table 7.1 summarizes 3TS's technical data.

Table 7.1: Technical data of the Three-Tank System

SYSTEM	
Number of tanks	3
Number of valves	6
Number of pumps	2
TANKS	
Vessel cross-section	154 cm^2
Channel cross-section	0,5 cm^2
Nominal height	60 cm
Maximal height	62 cm
VALVES	
Valve type	manual, open/closed
Cross-section	0,5 cm^2
PUMPS	
Pumping characteristic	linear
Maximal flow	100 cm^3/min

The manual valves allow the system to emulate many types of tank processes. The configuration implemented for this monitoring application is a 2-input-3-output system (s. Figure 7.2). Valves are not modeled; their positions are set to assure the shown configuration. The valves between the tanks are modeled as communicating pipes. This is also the case of the valves in front of the tanks, discharging to the collector pool. The lateral valve in tank T2 remains closed and is therefore not modeled. A possible process description for this 3TS configuration is:

This is a system of three fully communicating Tanks without valves for the online blending of substances A pumped into T1, and B pumped into T2. T3 is located between T1 and T2 and has no pump. Three products A', B' and C' result from the process as the output of T1, T2 and T3 respectively. A control system assures the composition of these products by controlling the inflow rates in T1 and T2. Because of the nature of the process and its elements, overflowing or completely draining the tanks must be avoided.

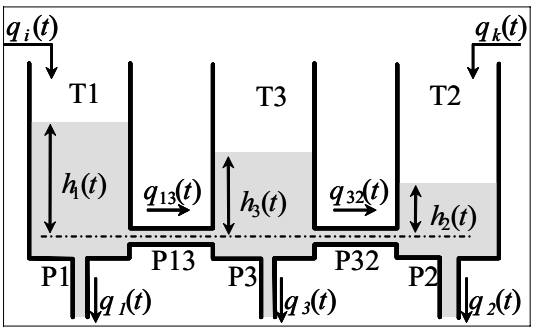

Figure 7.2: Structure of the Three-Tank System

The three-tank system is a benchmark process widely used for evaluating modeling, monitoring and control strategies for nonlinear MIMO systems. The complexity of the 3TS is determined by changing its configuration. For this reason, the 3TS is suitable for the development, implementation and evaluation of SQMA modeling strategies.

7.2 Development of the SQMA model with Rough Intervals

SQMA models do not describe what happens inside the modeled technical process. The modeler defines component terminals with their associated variables, and then qualitatively formulates the equations that govern all the relationships taking place between these terminals. For example, to model a resistor, terminals and the components are related through the Ohm's law with the current, however the relationship with the temperature and Jules' law is not described, because this temperature is not directly related to the terminals. Based on this qualitative component description, the situations that the resistor may exhibit are calculated. Finally, the components are unified at the system level. The following paragraphs show, step-by-step, the SQMA process applied to the 3TS case study.

7.2.1 Definition of the system component

SQMA model elaboration begins with the system structure's definition. Pumps and the collection pool are not modeled as part of the 3TS; instead, they define border conditions for the

whole system. Defining a suitable system structure for the Three-Tank System comprises the following steps:

(1) System decomposition in hierarchical layers, subsystems and components. Multilayer decomposition is frequently possible; nevertheless, no subsystems are required for the Three-Tank System. The 3TS is divided into eight components distributed in one layer. They are three tanks (T1, T3 and T2) and five pipes (P1, P2, P3, P13 and P32). Compare with Figure 7.2.

(2) If several subsystem or hierarchical layers were defined, the terminals in each level and subsystem must be determined and characterized. This step is unnecessary for the 3TS.

(3) Representation of the system as a circuit of components and subsystems, as exactly as possible, with a net-list, taking note of the correspondence of terminal types in each connection. For the 3TS, the resulting block diagram is shown in Figure 7.3:

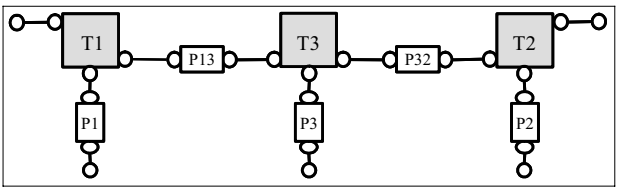

Figure 7.3: Net structure of the Three-Tank System

(4) Structurally and qualitatively similar components are grouped in component types that are modeled and reuse for each instance as required. Four 3TS components types are defined: Tank12 for T1 and T2 which includes a top input and two bottom channels, Tank3 for the T3 with no top input but three bottom channels, Pipe for the inter-tank channels P13 and P32, and finally V_Out for the three atmospheric discharge channels P1, P2 and P3. In these three channels, high precision is required to characterize the process products, which is not the case of P13 and P32, which are internal process variables.

7.2.2 Component modeling

At this point, the work begins at component level in the innermost hierarchical layer. Since 3TS has no multilayer structure, this step corresponds to the modeling of the eight defined components:

(5) Variables inside the component are defined, paying special attention to the energy storage in the components. In the tanks, for example, the inflow is separately defined from the two outflows, whereas in the pipes (no energy-storing capability) inflow and outflow are modeled by the same variable. In a pipe, the inflow is always equal to the outflow. Table

7.2 shows the components of the Three-Tank System with their corresponding variables. Observe correspondence between the model's component variables and the variables in the real process that these component variables represent [8].

Table 7.2: Components, component types, and variables

Component	Comp. Type	Terminal and type	Model Variable	Process Variable	Remark
T1	Tank12	1.potential	pp		*not measured*
		1.current	Qp	$-q_i$	Flow1
		2.potential	H	h_1	Level
		2.current	Qo	q_1	Flow2
		3.potential	pl		*same as Level*
		3.current	Ql	q_{13}	Flow3
T2	Tank12	1.potential	pp		*not measured*
		1.current	Qp	$-q_k$	Flow1
		2.potential	H	h_2	Level
		2.current	Qo	q_2	Flow2
		3.potential	pl		*same as Level*
		3.current	Ql	$-q_{32}$	Flow3
T3	Tank3	1.potential	pi		*same as Level*
		1.current	Qi	$-q_{13}$	Flow1
		2.potential	pl		*same as Level*
		2.current	Ql	q_{32}	Flow3
		3.potential	H	h_3	Level
		3.current	Qo	q_3	Flow2
P1	V_Out	1.potential	Q_AB		*not measured*
		1.current	p_AB	q_1	Flow1
P2	V_Out	1.potential	Q_AB		*not measured*
		1.current	p_AB	q_2	Flow1
P3	V_Out	1.potential	Q_AB		*not measured*
		1.current	p_AB	q_3	Flow1
P13	Pipe	1.potential	Q_AB		*not measured*
		1.current	p_AB	q_{13}	Flow1
P32	Pipe	1.potential	Q_AB		*not measured*
		1.current	p_AB	q_{32}	Flow1

(6) Each qualitative variable is declared in the corresponding component model file: intervals, labels, type and attributes ("Danger", "Ok", etc.) of the potential and current variables associated with each terminal. Rough Intervals are determined using the methods described in 4.3. Only level variables (h) in the 3TS tanks are modeled with RI, all other variables are modeled using crisp intervals, according to the original SQMA modeling procedure. This facilitates following the effect of RI in the whole system. For the level variables, three intervals are defined: [0 5), [20 40] and (55 ∞). These intervals correspond to the level regions where the tank can be said to belong 100% (LAI) to the qualitative descriptors

[8] Gauge pressure could have been defined instead of the tank level, because their behaviors are equivalent.

"Too_low", "Nominal" and "Too_full" respectively. Following the heuristic modeling method, the corresponding upper approximation intervals are determined and declared in the model files. Table 7.3 shows the intervals defined for the Three-Tank System.

Table 7.3: Interval definition for the Three-Tank System

Comp. Type	Model Variable	Intervals			
Tank12	pp	[0 ∞)			
	Qp	(-∞ -90]	(-90 -5]	(-5 0]	
	H	[0 20)	[5 55]	(40 ∞)	
	Qo	[0 0]	(0 10)	[10 80]	(80 ∞)
	pl	[0 20)	[5 55]	(40 ∞)	
	Ql	(-∞ 0)	[0 0]	(0 ∞)	
Tank3	pi	[0 20)	[5 55]	(40 ∞)	
	Qi	(-∞ 0)	[0 0]	(0 ∞)	
	pl	[0 20)	[5 55]	(40 ∞)	
	Ql	(-∞ 0)	[0 0]	(0 ∞)	
	H	[0 20)	[5 55]	(40 ∞)	
	Qo	[0 0]	(0 10)	[10 80]	(80 ∞)
V_Out	Q_AB	[0 0]	(0 10)	[10 80]	(80 ∞)
	p_AB	[0 0]	(0 ∞)		
Pipe	Q_AB	(-∞ 0)	[0 0]	(0 ∞)	
	p_AB	(-∞ 0)	[0 0]	(0 ∞)	

(7) Border conditions are established for the variables inside the component, establishing the rules that must be fulfilled for each valid situation. These relationships are called Situation Rules. For this process, flow leaving the tank requires the existence of water in the tank (level ≠ 0), and the pressure heads in the channels at the bottom of the tank are the same.

(8) Component comment rules are defined in order to determine which qualitative descriptor will be assigned to each situation. The qualitative values of each interval are declared in this section.

(9) The component situation table is built, with all the possible combinations of the intervals previously defined for each variable. The tank would require the construction of a table with 405 (1x3x3x3x5x3) rows, which are the possible combinations from the tank variables.

(10) The component situation table in (9) is reduced by checking each possible combination with situation rules defined in (8). Interval combinations that contradict at least one rule are eliminated from the situation table. For the component type Tank12 in the 3TS, the 405 original combinations are reduced to 90 valid situations.

The process repeats steps (5) through (7) for each component, until the present level is complete.

7.2.3 Modeling component dynamics

Three new steps are added to represent the dynamic component behavior, enabling therefore their situation-based monitoring. They are:

(11) Formulation of equations as in (8), describing the dynamic behavior for each energy-storing component. These equations will determine the dynamic behavior of the entire system. In 3TS modeling, indicate that levels cannot change instantaneously (energy-storing variable) and that any change must follow the total equivalent flow into or out of the tank.

(12) A dynamic model in the form of a transition matrix is automatically produced in each component by verifying the above-defined dynamic conditions for each situation transition. Each allowable transition is marked with one in a quadratic matrix with situations in rows and columns. Impossible transitions are marked with zero.

(13) For the monitoring application in the 3TS, the previous matrix is re-expressed using transition probabilities. If there is no a priori information about which situations are more likely reached from a given situation, equal probability is assumed for all the transitions from a given source situation to the different target situations. Transition probabilities are then calculated by dividing the value in each matrix field, by the total number of situations that can be reached from the current one.

As with the modeling of the component situations, the steps (11) through (13) are repeated for each component, until all the components of the present level are modeled.

7.2.4 System modeling

The SQMA modeling of system components is at this stage complete. This includes the modeling of the system dynamics, which is distributed in the different component transition matrixes. Now the modeling continues at system level to produce the table of valid system situations.

(14) Process variables are defined considering voltage (Potential Energy: pressure, tension, height, elongation of a spring, etc. → Potential Dimension) and current (Kinetic Energy: flows in general, movement, etc. → Fluent Dimension), in such a way that variables in different components can be connected using the Kirchhof's laws of nodes and meshes. Almost for any real technical process, analogies to electric circuits can be found. With the net-list from (3) and the component definitions, establish the system equations by solving the analog electric circuit following Kirchhof's laws. A collection of fourteen linearly independent mathematical relations (system equations) is the result for the 3TS.

(15) A system situation table is built by representing all the combinations that may result from the situation tables of the precedent level. Here the resulting tank situations are combined

with the situations in the pipes. This results in more than 397 million possible combinations for the 3TS, which is an important reduction from the original 6,6 trillions interval combinations.

(16) All these combinations must be validated, based on the system equations produced in (14) and interval arithmetic. If a situation does not satisfy all these expressions, this confluence of component situations is invalid and must be deleted from the system situation table. For the Three-Tank System, 1,85 million valid situations are determined.

(17) The above produced system situation table results from evaluating the UAIs in the case of the three levels. If only the LAI is evaluated, the size of the system situation table reaches its minimum. However, the model developer may establish a confidence (C) threshold to accept rule evaluations. For the 3TS, the system situation table can be reduced to about 966 thousand valid situations by demanding a C > 0,5.

Steps (14) through (17) are repeated until all the subsystems in the present level are modeled, then go to the immediately superior one, repeating steps (5) through (17) for the remaining subsystems. For the 3TS no further hierarchical levels are defined, therefore, the 3TS SQMA modeling is at this stage complete. The system situation table together with the component transition matrixes constitutes the Situation-based Qualitative Model of the technical process 3TS, which is ready to be used for situation-based process monitoring.

7.2.5 SQMA relational model

In order to meet real-time requirements, the monitoring application must access data quickly. Since nondeterminism, model size, and search structures in the SQMA application consume a lot of memory, a flexible data storage format is necessary for rapid data access. After analyzing different data storage formats, a relational database format was selected. A relational database provides fast and flexible access to the information in the model. Since the table-based data representation in relational databases is compatible with the way information is represented in SQMA; an SQMA relational model is easily readable. According to the structure of SQMA models, the following relational tables are defined:

- **INTERVALS** table, described in Table 7.4, is the first table verified during the process monitoring. It defines the space partitions for each component variable. INTERVALS table support the definition of crisp intervals, Rough Intervals or combinations of both.

- **COMPONENT_SITUATIONS** table is the main table of the SQMA model. It describes the valid situations in each component by defining the corresponding interval values for each component variable. Its fields are described in Table 7.5.

Table 7.4: Structure of table INTERVALS

Field name	Data type	Field description
ID	Integer	Interval identifier
VARIABLE	Text	Variable upon which intervals are being defined
COMPONENT	Text	Component where the variable is being characterized
FLAG1	Text	Left border type (Open/Closed)
FLAG2	Text	Right border type (Open/Closed)
INTERVAL1	Double	Value of the left interval limit
INTERVAL2	Double	Value of the right interval limit
REMARK	Text	Qualitative interval description

Table 7.5: Structure of table COMPONENT_SITUATIONS

Field name	Data Type	Field description
SITUATION ID	Number	Component situation identifier
COMPONENT	Text	Component where the situation is defined
LEVEL	Number	Interval identifier in the level variable (0 in pipes)
FLOW1	Number	Interval identifier for the first flow variable
FLOW2	Number	Interval identifier for the second flow variable (0 in pipes)
FLOW3	Number	Interval identifier for the third flow variable (0 in pipes)
LABEL	Text	Situation descriptor (from comment rules)
ATTRIBUTE	Text	Situation attribute: D (danger) ,U (undesirable) or N (normal)

- **SYSTEM_SITUATIONS** table validates the combination of component situations resulting from the online monitoring. Component situations are determined by using the tables above and checked against the combinations in this table. Table 7.6 listed the fields declared in this table.

Table 7.6: Structure of table SYSTEM_SITUATIONS

Field name	Data type	Field description
ID	Number	System situation identifier
TANK1	Number	Component situation identifier in TANK1
TANK2	Number	Component situation identifier in TANK2
TANK3	Number	Component situation identifier in TANK3
PIPE1	Number	Component situation identifier in PIPE1
PIPE2	Number	Component situation identifier in PIPE2
PIPE3	Number	Component situation identifier in PIPE3
PIPE12	Number	Component situation identifier in PIPE12
PIPE23	Number	Component situation identifier in PIPE23

- **COMPONENT_TRANSTIONS** table implements the probabilistic forecasting of reachable situations in the monitoring application, by specifying valid situation transitions for each component. COMPONENT_TRANSTIONS table consists of the information detailed in Table 7.7

Table 7.7: Structure of table COMPONENT_TRANSTIONS

Field name	Data type	Field description
COMPONENT	Text	Component identifier
SITUATIONID_S	Number	Source component situation
SITUATIONID_T	Number	Target component situation
PROBABILITY	Number	Transition probability

7.3 SQMA Monitoring Application

This section describes the main software modules of the monitoring application developed to validate the formulated concepts. The monitoring application consists of a dialog system for monitoring the Three-Tank System, based on its SQMA relational model. The combinatorial nature of SQMA makes it difficult to implement the concept generically. Optimized memory consumption and real time requirements compete as design criteria during the development of the prototype. The SQMA monitoring application is implemented specifically for the 3TS using the existing knowledge about it and its behavior. The results of the application evaluation are discussed in section 7.4.

The situation-based monitoring application is implemented in Visual C++ and uses a Microsoft Access database for the SQMA relational model. Important features of the developed application are:

- The indistinct use of crisp and Rough Intervals for SQMA process monitoring.
- The situation forecasting based on probabilistic component transition matrixes.

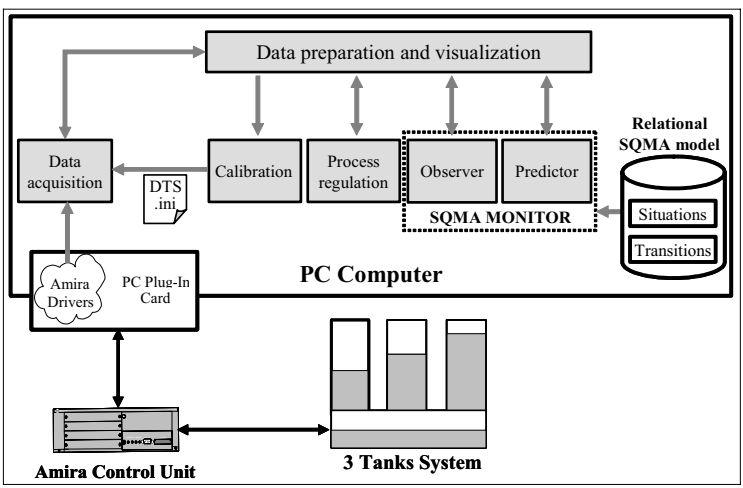

Figure 7.4: Architecture of the monitoring application

Implementing a Stochastic Qualitative Automata to forecast reachable situations and using RI for system observation introduces the management of probabilistic information, which in turn enriches the information delivered by monitoring applications (compare [Fran01]).

Figure 7.4 shows the architecture of the monitoring application for the 3TS. The hardware consists of an Amira Control Unit, Three-Tank System DTS200 and data acquisition card DAC98 with a 50-poles connection cable. The monitoring software runs on a personal computer with MS Windows 98 operating system, because Amira drivers exclusively work in this environment. The following sections describe the main modules of this monitoring application.

7.3.1 Data preparation and visualization

The *Data preparation and visualization* module administers the complete situation-based monitoring application by getting sensor data from the *Data acquisition* module and to calculate the flow between the tanks and between a tank and the basin. These flows are calculated with the formula (7.1) for a system configuration with the references in Figure 7.5. The equation parameters g (gravity acceleration, $g = 981$ cm/sec^2) and δ (water density, $\delta = 1$ g/cm^3) are system constants.

$$Q = \text{sign}(h1 - h2) * a * \delta * \sqrt{2 * g * |h1 - h2|} \tag{7.1}$$

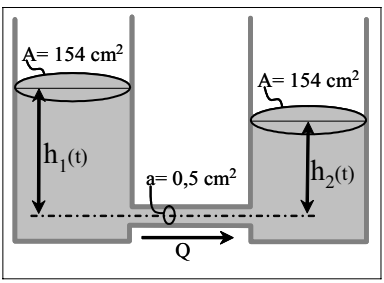

Figure 7.5: Parameters of flow calculation in the Three-Tank System

The interface window in Figure 7.6 displays the tank levels gathered by the *Data acquisition* module, and allows the user to manipulate the amount of water pumped into the tanks. This module also administers the calibration of pumps and sensors (*Calibration* module), by changing the calibration parameters in a separate calibration dialog window. The *Data preparation and visualization* module, adapted in a Master Thesis by Katamaneni [Kata03] from its original version [Laud99] to support the *Observer* module, is the first stage in SQMA monitoring. The description of the observer module is the purpose of the next section.

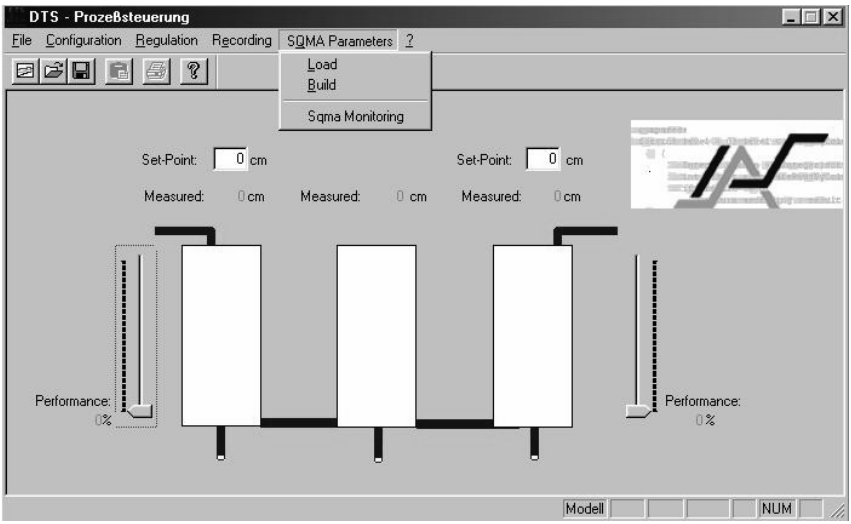

Figure 7.6: Process monitoring window

7.3.2 SQMA monitoring: *Observer*

The observer module, which is triggered from the SQMA monitoring dialog window (s. open menu in Figure 7.6), opens a separate window (Figure 7.7), and displays situation labels and membership values matching the current process state. Online water level data from three tanks is obtained, together with the pump settings and the calculated flows from the visualization module. This data is compared with the information in the SQMA relational model in order to derive the valid situations, i.e., membership values and situation labels that are determined by the system's behavior. Once all the values for the variables under the components are available, they are compared with the interval data present in the table INTERVALS. This results in the interval IDs to which the online data belongs. If only one interval ID is found, then $\rho = 1$; otherwise, rough membership value (ρ) is calculated using the formulas:

$$\rho_{interval1} = (\text{real value} - \text{begin interval 2})/(\text{end interval1} - \text{begin interval2}) \quad (7.2)$$

$$\rho_{interval2} = 1 - \rho_{interval1} \quad (7.3)$$

Where interval1, interval2 are the first and second intervals correspondingly, where the coming online data was found. For example, if interval1 is (-50 -5), interval2 is (-20 0) and the real value is -10 then the membership value of interval1 is calculated as (-10 + 20)/(-5 + 20) = 0,66 and the membership value of interval2 results 1 - 0,66 = 0,33.

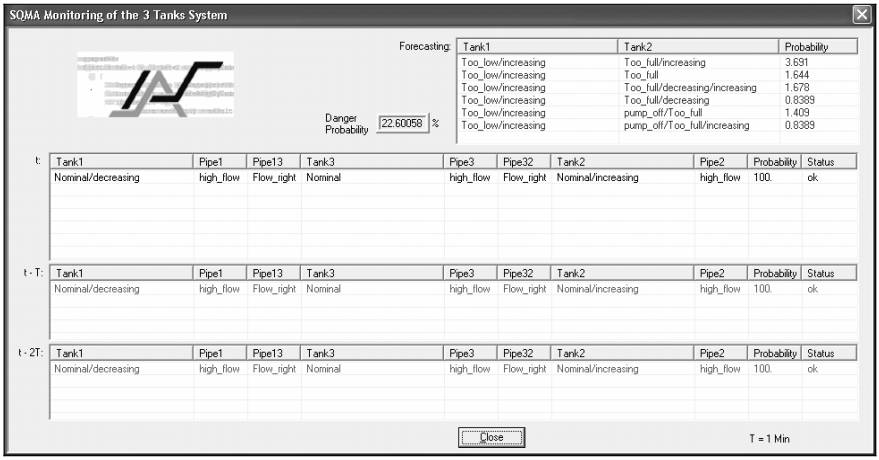

Figure 7.7: SQMA monitoring application

After acquiring the interval IDs for each variable, these IDs are combined at the component level. If the online data is present in more than one interval, then for every extra interval one extra combination is prepared. Therefore, each component has one or more combinations of intervals. The combinations of interval IDs are compared with the data in the COMPONENT_SITUATIONS table for validation and information retrieval. By using these combinations, the corresponding component situation IDs are found. This process is repeated for all the components in the system until the situation IDs for all the components are located. Then, the membership value (ρ) is calculated for each combination by multiplying the membership values of all corresponding component variables.

The component situation IDs combinations are compared with the entries in the SYSTEM_SITUATIONS table to determine whether they are valid. Many combinations may result if Rough intervals are used. If a combination of situation IDs is present in the system situation table, then the situation is valid; if not it is invalid, and thus removed. The corresponding membership values are normalized again to compensate for the eliminated combinations.

The remaining information about the current situations is transferred to the SQMA monitoring display window (s. Figure 7.7) to be displayed. The information corresponding to the current state appears in the second table from the top (here the system is in a normal operation mode) and updated every ten seconds. System situations one and two minutes before (third and fourth table) are also shown.

Every 60 seconds, the information in the three boxes shifts to the next list box down, and data in the list box at the bottom of the window vanishes. These windows help the user verify the results for a consistent time period and analyze instantaneous dynamic behaviors by comparing

the evolution of the situations and their membership values. These windows are periodically refreshed as long as the SQMA monitoring application runs.

7.3.3 SQMA monitoring: *Predictor*

Forecasting is implemented using the SQMA dynamic models at component level. The module obtains from the observer module the current situation IDs from tanks and pipes. With these component situations and reading data from COMPONENT_TRANSITIONS table, transition vectors (reachable situations and probabilistic value) are calculated for each component. Since some situations are reachable from several source situations, the probability of repeating component situations must be added together. These situations are more likely reached than others are. These vectors are used to determine possible future system situations. Pipes are non-energy-storing components (s. 5.2.2); therefore, they reachable situations are not calculated.

From the component transition vector, the predictor module reads each situation that T1 can reach from current process state, then uses these values to retrieve from SYSTEM_SITUATIONS all the possible situations in T1. This reduced set of situations is then filtered first with respect to the situations to which T2 can transit and then considers the T3 transitions.

In order to speed up the monitoring application and facilitate the operator's situation assessment, from the massive number of situations that are reachable from a given system state, only dangerous situations and their corresponding probabilities are visualized. The total probability of reaching a dangerous situation is also displayed. Furthermore, since only T1 and T2 can reach dangerous situations, only the labels associated with these components are retrieved and displayed, as shown in the corner top-right in Figure 7.7. This notably speeds up the monitoring application.

The SQMA situation forecasting module, implemented in Cachero's Master Thesis [Cach03], has a timer which is set to t = 3 min. In this example, the probability of the system reaching a dangerous situation is almost 23% (s. small edit-box in Figure 7.7).

7.4 Concept evaluation with the case study

In accordance with 3.4, five requirements are formulated on the solution concept to assure the accomplishment of the goals of the present research:

- State observation and transition recognition (s. 3.4.1)
- Identification of impossible or hazardous states (s. 3.4.2)
- Vagueness and uncertainty management (s. 3.4.3)
- Model size reduction in relation to demanded precision (s. 3.4.4)
- Compatibility with existent SQMA models (s. 3.4.5)

This section verifies the fulfillment of these requirements based on the evaluation of this 3TS case study.

7.4.1 Evaluation of the model and the modeling process

Seven models of the 3TS, organized in two sets (Set A and Set B), support the evaluation of the new SQMA method. In Set A, the only difference between the three models is the number and kind of the intervals used to model variable level in the three tanks. Two reference models, following the conventional SQMA procedure, define three and five crisp intervals respectively for the level. A third model uses three Rough Intervals for the level, and crisp intervals for the remainder variables. The system consisted of three of these hybrid components (tanks) and five components defined after the original method (s. section 7.2). Using the third model, the compatibility of the new method with existent SQMA models is verified. Moreover, the same version of the SQMA Modeling Software set - version 3.0, developed by Fernandez [Fern03] - was used for all three cases.

For the variables where Rough Intervals are used (the level variables of the three tank components), vague concepts such as “almost_empty” and “too_full” are represented by the gradual overlapping of precise concepts as “empty”, “filled” and “overflow”, which demonstrates the possibility of representing vague concepts with RIs. The representation of the uncertainty resulting from SQMA’s nondeterminism is also improved with the re-expression of the component transition matrixes using transition probabilities.

The number and kind of intervals determine the model’s precision and size. The number of distinguishable information areas of the solution space determines model’s precision. What is calculated as the number of possible combinations of the intervals and overlaps (in the case of using RI) that are defined in each process variable (physical variables in the technical process, not their counterparts in the model, compare Table 7.2). The number of intervals, together with the rules, also determines how many situations are contained in the model, and thus its size. Size of the transition matrix is thereby not considered.

The four models in Set B support the examination of the relationship between process complexity, model’s size and the required precision. These models are related to one another as in Set A (i.e. similar distribution of intervals for the three level variables), but are larger and more precise. One difference is that Set B has two models with RIs, instead of only one as Set A. The first model (RI-Model 1), similar to the RI model in Set A, does not consider rule confidence (equivalent to requiring $C > 0$ to accept rules), whereas for the second one (RI-Model 2) $C > 0{,}5$ is demanded.

The models with RIs must be compared with their respective reference models. The information summarized in the Table 7.8 and the diagram in Figure 7.8 representing the models in Set B enables this comparison. The models with Rough Intervals are clearly smaller than conventional

SQMA models of comparable precision, and more precise than conventional models of its size. That can be observed in the two sets of models. Moreover, the effect of the confidence-base reduction of situations on the model size is evident in Set B. The size of the RI model determined with C > 0,5 reaches its minimum, while keeping its precision. This minimum model size coincides with the number of situations in the equivalent SQMA model with three crisp intervals per level variable. The original RI model, where all overlaps are accepted (C > 0), is appreciably larger, yet still smaller than the accurate model with five crisp intervals.

Table 7.8: Comparison of size and precision in test models

	Size	Precision	Description
Set A			
Model 1	47.000	$1{,}2 \times 10^5$	Accurate reference model
Model 2	14.400	$0{,}26 \times 10^5$	Compact reference model
RI-Model	28.225	$1{,}2 \times 10^5$	Test model
Set B			
Model 1	$3{,}05 \times 10^6$	$6{,}5 \times 10^5$	Accurate reference model
Model 2	$0{,}97 \times 10^6$	$1{,}4 \times 10^5$	Compact reference model
RI-Model 1	$1{,}85 \times 10^6$	$6{,}5 \times 10^5$	Test model
RI-Model 2	$0{,}97 \times 10^6$	$6{,}5 \times 10^5$	Test model with C > 0,5

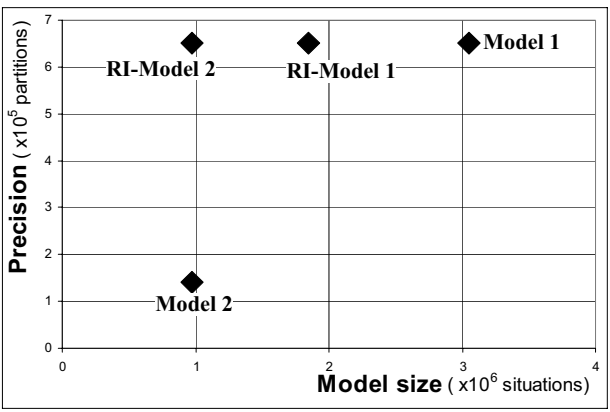

Figure 7.8: Comparison of SQMA models

All of these systems are completely developed in spite of the high accuracy requirements. In addition, there is no modeling restriction because production of the system transition matrix is unnecessary. As shown in section 6.1, if limited resources of a desktop computer are used, a system transition matrix can only be developed for the smallest model.

7.4.2 Evaluation of the monitoring concept

Figure 7.9 displays the result of monitoring the Three-Tank System, using three of the four models of Set B (the same models represented in Figure 7.8). Since RI-Model 2 (C > 0,5) is a subset RI-Model 1 (determined without considering confidence-based reduction, C > 0), the performance of monitoring with RI-Model 1 must be at least as good as the performance using RI-Model 2. Therefore, the RI-Model 1 is not used for the monitoring task. After observing these results, the monitoring based on Rough Intervals is clearly possible, and the expected behavior is reproduced. On one hand, the model with three Rough Intervals is appreciably more exact than the model with three crisp intervals (Model 2), even though they are the same size.

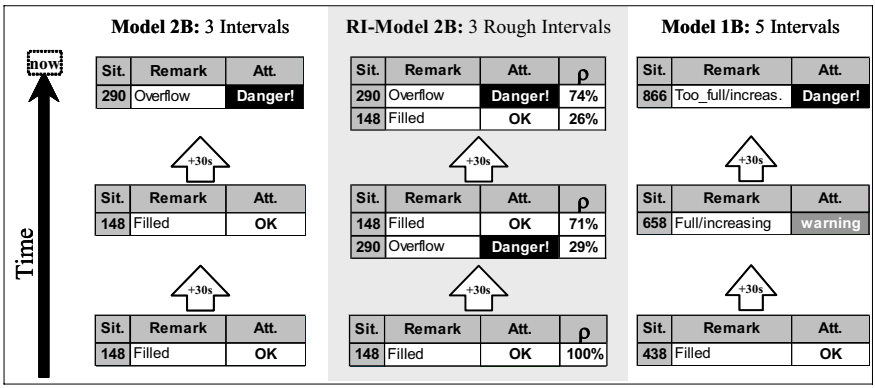

Figure 7.9: Model performances during process monitoring

On the other hand, precision in RI-Model 2 and Model 1 (with five crisp intervals) is comparable; situations of the last two readings corresponding to the RI model in Figure 7.9 are marked "OK" and "Danger!". This combination of situations reproduces first the warning and then the dangerous situation in Model 1. The model with RI represents the difference between danger and warning through the relationship between the membership values (ρ). The model with Rough Intervals is more accurate than the other model because the continuous evolution of the situation membership values enable following the process more closely, when it goes from a safe situation (Sit. 148) to a dangerous one (Sit. 290).

The enhanced precision becomes clear by interpreting the observed situations. The information delivered by the models with RIs could be read as: *"The tank is too full, and is thereby in the proximity of a dangerous condition"*. The process tendency can be weighed up as well, by interpreting the changes in the membership values. In this case, for example, the observation of the system can be refined: *"The tank is too full with a fast rising liquid level, and is thereby in the proximity of a dangerous condition"*. The possible future system situations can be also

considered. Following this example, the information can be completed with: *"This dangerous condition will be reached with a probability of 85%"*.

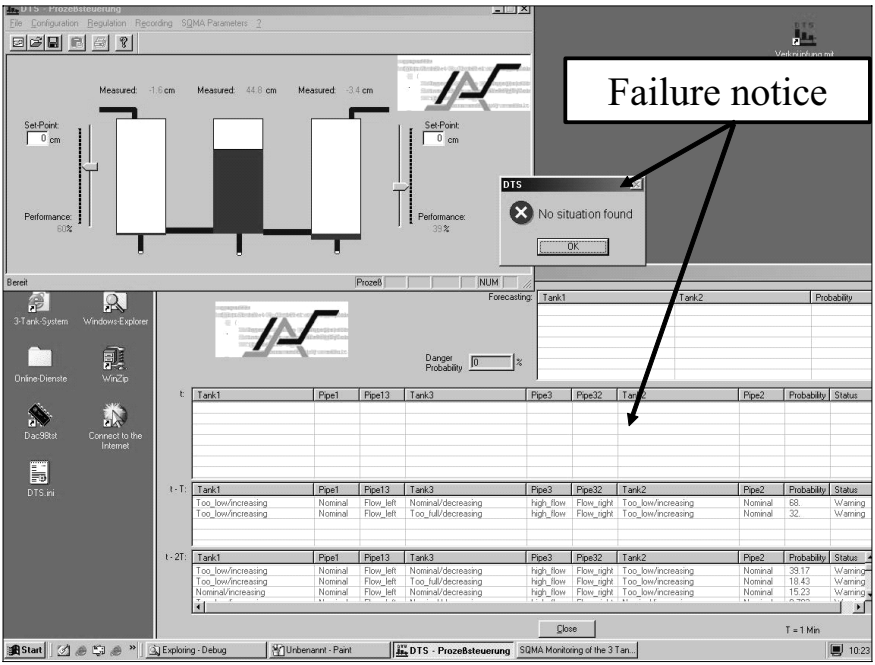

Figure 7.10: Simulation of an impossible state in the SQMA monitoring application

The test resulted as it was expected. Inclusive not modeled (impossible, not allowable) states are simulated in the real system and correctly identified by the monitoring application. This is shown in the following example (Figure 7.10). An impossible situation is simulated using signal disturbance controllers in the digital regulating unit of the 3TS: The outermost tanks are almost empty, whereas the center tank is almost full. Such a state cannot take place in reality and therefore was eliminated from the SQMA model. Consequently, the monitoring application is unable to find a corresponding situation or set of situations in the model; so no situation is shown in the corresponding window. Additionally an application message window pops up. This condition can be originated, for example, by a defect in the level sensors of T1 and T2 (regard the inflow provided by both pumps), however also a combination of obstructions and leaks in the pipes, i.e. structural changes in the model, may cause similar unmodeled behaviors.

8 Research summary and future work

8.1 Summary of the research work

A solution concept for modeling complex systems considering vagueness and uncertainty was developed following the project goals and the requirements defined upon these, after analyzing the state of the art of the related knowledge fields. SQMA serves as the basis for this solution concept, attending particularly to its situation-based and table-driven features, hierarchical structure and component orientation, experience with safety-critical automation applications, and simple modeling procedure.

The new concept represents vagueness, inherent in common sense and empirical knowledge about the technical process, directly in the declaration of the component variables. At the same time, this representation is compatible with SQMA's interval-based (qualitative) representation and allows the further managing of transition probabilities between situations. Rough Intervals are based on rough sets [Pawl92]. An RI shares with the original SQMA intervals its operational framework, but it is supported on a more convenient representation and interpretation system that identifies and labels "interesting" operations and uncertainty areas. These models are, besides, notably smaller than the original SQMA models for a given application precision and information contents.

The information represented by the Rough Interval concept exceeds the information in the original crisp intervals, because it handles vagueness and uncertainty in a single theoretical framework. RIs also make possible the representation and further processing of transition probabilities in SQMA. This second element of the solution concept pays attention to the further management of the uncertain knowledge. Whereas the first element is centered on situations, this one is developed around the transitions. The probabilistic transition model is further managed online with a stochastic qualitative automaton and the observed (rough) situations that deliver a reliable prediction of system situations.

Precise process modes are achieved by combining three approaches: situation-based qualitative modeling, Rough Intervals and probabilistic transition models. These models are suitable for the situation-based monitoring of complex systems. Forecasting reachable situations is particularly enhanced using Stochastic Qualitative Automata, which are optimized, for its reliable operation in real-time applications.

8.2 Advantages, disadvantages and limitations of the proposed method

SQMA modeling precision is greatly improved by the representation of vague and uncertain information. In addition to the process information that is represented in an SQMA model, now it is also possible to include vague, ill-defined concepts and, when available, measures of the uncertainty about process state changes. Vague concepts are represented in SQMA through the superposition of precise characteristics and uncertain information about the concept borders expressed as rough membership values. Uncertainty measures are directly integrated into the process dynamics model (s. Figure 8.1).

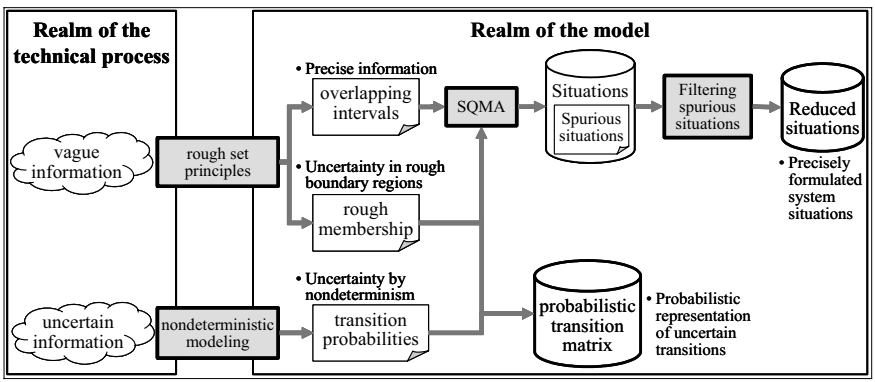

Figure 8.1: Vagueness and uncertainty management in SQMA models

The effect of the Rough Interval overlaps is twofold: on the one hand it contributes to the precision enhancement if a model size is to be kept; on the other hand it allows substantial model size reduction for a given application precision. All in all, the final effect is SQMA's improved ability to effectively represent complex systems.

The new developments in the SQMA models have a positive effect on the process monitoring, and expand the method's applicability. Compared with the original SQMA monitoring applications [Fröh97], the analysis of the current process state delivers information about the process' steady condition and its short-term dynamic behavior. Forecasting reachable situations is improved with the introduction of transition probabilities; the delivered information is richer and better represented. Classification, grouping and filtering situations is better with this information. Overall, the descriptive power of SQMA models is greatly improved, and this can be appreciated by their application to process monitoring.

The introduced SQMA method has limitations as well. SQMA was conceived for the qualitative modeling of complex dynamic (continuous and discrete) technical systems, where the

application of deterministic, well-defined models of these systems is impossible. SQMA, even with the enhanced precision introduced by the new concepts, will never match the exactness and precision of a well-developed deterministic model. On the other hand, in spite of the SQMA improvement concerning its capability of representing complex systems, this method is still based on a combinatorial engine, which examines all possibilities exhaustively. Therefore, a process complexity level is eventually reached, where the application of SQMA models is no longer possible or sensible. This complexity limit is, in any case, superior to the complexity level manageable when crisp intervals are used, which application spectrum is already superior to that of the qualitative and fuzzy methods analyzed in the present research work (s. 2.3). In general, SQMA using Rough Intervals is applicable in the cases where SQMA with crisp intervals is also applicable and in many other cases that, because of their complexity or the demanded model precision, are intractable with this technique.

8.3 Future work

Before closing this discussion, ideas about future SQMA upgrades may be beneficial to coming researchers. The concepts in this section are merely a selection of many ideas, which are driven by newly discovered capabilities and the recognition of needs, opportunities, limitations and strengths when working with SQMA, rough sets and other techniques considered during this investigation.

8.3.1 Improvement of computer support in SQMA modeling

One of SQMA's most valuable assets is its suitability for the organization of model libraries and the automation of the modeling process. This aspect of SQMA, however, was not extensively developed during this research work. Although a new and integrated modeling environment for SQMA in MATLAB and Simulink was conceptualized during this research, the idea of developing an SQMA MATLAB/Simulink toolbox is not new. Grzesiak [Grze00] realized a first explorative study in 2000 in an undergraduate project. This work was unsuccessful in delivering an SQMA Toolbox prototype because of the demanded programming work and the conceptual immaturity of the considered approaches. However, the importance and convenience of developing such a tool, as well as the main aspects to be considered, were clearly stated.

One of the more sensitive activities in developing SQMA models is the initial definition of component, system description and netlist files, which must be consistent with each other. Most, if not all, modeling failures can be traced to this phase because it must be done manually in spite of its importance and complexity.

While this research was being conducted, a new graphical approach based on Simulink was conceptualized to support this stage. Components may be defined as Simulink masked subsystems of rules using the SQMA interval arithmetic block set. These components must

follow the SQMA netlist conventions while variable partitioning can be implemented using the mask definition.

A converting MATLAB application would complete this Simulink tool set. The existing SQMA model structure, based on chained ASCII files, may be used as an intermediate format to warrant compatibility with previous SQMA models. The existing SQMA modeling programs may be recompiled to be callable directly from MATLAB in order to minimizes the impact on the original code, which is considered mature and reliable, and keeps the compatibility with previously developed SQMA models.

MATLAB and Simulink would provide a common platform for the development of SQMA models and make possible the development of SQMA models directly by the technical system's design and integrating qualitative and quantitative models in a single application, as SQMD requires. Additionally, MATLAB is a highly supported application that would eliminate the dependency of the SQMA modeling tools on the operating system and provide a very interesting platform for the further growth of SQMA; for example, to support the development monitoring applications.

8.3.2 SQMA for fault diagnostics

In [Grze03], within the context of the present research work, a concept for model-based diagnosis of complex systems, based on SQMA, was examined. Several methods for combining qualitative and quantitative information for process diagnosis applications were discussed. Several combined solutions regarding the SQMA modeling, analysis and monitoring were thereby characterized.

An SQMA solution with integrated partial approaches, such as Rough Set Theory and data mining, was chosen for detailed consideration. To enhance the quality of the extracted knowledge and decision-making, SQMA models must be redesigned to represent faulty process behaviors. Diagnostic reasoning is based on this fault model, whose structure was developed in this study as well. The method comprises three basic units: fault modeling based on SQMA normal behavior model, building an adequate knowledge base and a diagnostic algorithm. A future goal is the combination of the Rough Intervals with this SQMA-based diagnosis structure and other conventional tools and finally their integration within condition monitoring systems.

The qualitative model has essential advantages that justify the proposed approach. The algorithms for processing the qualitative values are efficient and their results are directly interpretable by humans. However, the benefits of these systems seem to depend on how well they can use the information available in the form of rules and empiric operation knowledge.

8.3.3 Improvement of transition modeling

During the present research work, the definition of system situation was reviewed and upgraded by integrating rough set principles. However, this study also emphasized the sensitiveness of SQMA transition management, as well as the relevance of transitions for effective process condition monitoring. Transition management is also crucial for the future development of fault diagnosis applications based on SQMA (s. previous section).

Probabilities in the SQMA transition matrix resemble dynamic process behavior better than the original transition matrix with ones and zeros. Nevertheless, neither the probabilistic nor the original version of the SQMA transition matrix considers the important influence of time in this behavior. Even if the probabilistic transition matrix is kept, time influence can be introduced in the model by substituting the a priori probabilities with probabilistic time functions. External parameters such as events, system conditions and history (for n^{th}-order Markov processes) can also be included as arguments of these functions. The actual probability values can be computed in real time and further used in the forecasting.

Another idea worthy of being considered is modeling dynamic systems with Stochastic Petri Nets. The Petri net concept, however, is not directly compatible with SQMA [Cach03]. The development of a new concept to integrate whichever one in the other or to merge both techniques in a third concept should be investigated.

Bibliography

[AvJe99] T . Aven, U. Jensen: Stochastic models in reliability, Applications of Mathematics 41, Springer, New York, 1999.

[Babû98] R. Babûska: Fuzzy Modeling and Identification Toolbox, Control Engineering Laboratory, Delft University of Technology, Netherlands, August 1998.

[Bart87] B. Bartsch-Spörl: Ansätze zur Behandlung von fallorientiertem Erfahrungswissen in Expertensystemen, KI - Künstliche Intelligenz, Vol. 1, No. 4, pp. 32-36, 1987.

[BeCi89] J. Bechta Dugan, G. Ciardo: Stochastic Petri Net Analysis of a Replicated File System, IEEE Transactions on Software Engineering, SE-15, No. 4, pp. 394-401, 1989.

[Berl02] F. Berlin: Integration von Fuzzy-Intervallen in die qualitative Modellierungssprache SQMA, Diplomarbeit DA-1869, IAS, Universität Stuttgart, 2002.

[Bezd81] J. Bezdek: Pattern recognition with Fuzzy Objective Function Algorithm. New York, Plenum Press, 1981.

[Bezd93] J. Bezdek: Editorial: Fuzzy models - What are they, and why?, IEEE Transactions on Fuzzy Systems, Vol. 1, No. 1, pp. 1-6, 1993.

[Bezd94] J. Bezdek: The thirsty traveler visit Gamont: A rejoinder to comments on Fuzzy models - What are they, and why? IEEE Transactions on Fuzzy Systems, Vol. 2, No. 1, pp. 43-45, 1994.

[Bieg03] U. Biegert: Ganzheitliche modellbasierte Sicherheitsanalyse für Prozessautomatisierungssysteme, Ph.D. Thesis, Universität Stuttgart, 2003.

[BoBo94] A. Bonarini, G. Bontempi: A Qualitative Simulation Approach for Fuzzy Dynamical Models, ACM Transactions on Modeling and Computer Simulation, Vol. 4, No. 4, pp. 285-313, 1994.

[BrLi98] G. Brajnik, M. Lines: Qualitative modeling and simulation of socio-economic phenomena, Journal of Artificial Societies and Social Simulation, Vol. 1, No. 1, 1998.

[BSBH98] I. Baturone, S. Sánchez, A. Barriga, J. L. Huertas: Optimization of adaptive fuzzy processor design, Proceedings of the XIII Conference on Design of Circuits and Integrated Systems (DCIS'98), Madrid, pp. 316-321, 1998.

[BTGN84] J. Bechta Dugan, K. Trivedi, R. Geist, V. Nicola: Extended Stochastic Petri Nets: Applications and Analysis, in E. Gelenbe: Performance '84, North-Holland, Paris, 1984.

[Buch99] K. Buchacker: Combining fault trees and petri-nets to model safety-critical systems. In A. Tentner: High Performance Computing, The Society for Computer Simulation International, pp. 439-444, 1999.

[Cach03] L. Cachero: Online probabilistic forecasting in a three tanks system, Master Thesis MT-1909, IAS, Universität Stuttgart, 2003.

[Cell91] F. Cellier: Continuous System Modeling, Springer-Verlag, New York, 1991.

[dKBr84] J. de Kleer, J. Brown: A qualitative physics based on confluences, in D. Bobrow: Qualitative reasoning about physical systems, North Holland, Amsterdam, ISBN 0-444-87670-7, pp. 7-83, 1984.

[DIN66201] DIN 66201: Deutsche Industrie Norm für Begriffe bei Prozessrechensystemen, Teil 1, Beuth Verlag, Berlin, Mai 1981.

[DuPr94] D. Dubois, H. Prade, J. Lang: Possibilistic logic, in Gabbay, Dov M. et al: Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3, Oxford University Press, pp. 439-514, 1994.

[Dvor92] D. Dvorak: Monitoring and diagnosis of continuous dynamic systems using semiquantitative simulation, Ph.D. Thesis, University of Texas, 1992.

[Falt97] B. Faltings: Qualitative Models as Indices for Memory-Based Prediction, in IEEE Expert, IEEE, May/June, pp. 47-53, 1997.

[Fern03] M. Fernández: Integration of the concepts for handling vagueness and uncertainty in the SQMA modeling software, Master Thesis MT-1906, IAS, Universität Stuttgart, 2003.

[Forb84] K. Forbus: Qualitative process theory, in Artificial Intelligence Vol. 24, Elsevier Science Publishers, North-Holland, ISBN 0-444-87670-7, pp. 85-168, 1984.

[Fran01] R. Frank: Entwicklung einer Internetanbindung für den Modellprozess Drei-Tank-System, Diplomarbeit DA-1789, IAS, Universität Stuttgart, 2001.

[Freg03] G. Frege: Grundgesetze der Arithmetik 2, in Geach and Black: Selections from the Philosophical Writings of Gotlob Frege, Blackweil, Oxford 1970. Original paper written in 1903.

[Fröh94] P. Fröhlich: Fuzzy-based qualitative modeling for hazard analysis, AAAI 94 Workshop AI & Systems Engineering, 1994.

[Fröh97] P. Fröhlich: Überwachung verfahrenstechnischer Prozesse unter Verwendung eines qualitativen Modellierungsverfahrens, Ph.D. Thesis, Universität Stuttgart, 1997.

[GoSi84] G. Goodwin, K. Sin: Adaptive Filtering prediction and Control, In Prentice-Hall Information and System Science Series, Prentice-Hall Inc., Englewood Cliffs, NJ, U.S.A. ISBN: 0-13-004069-X, 1984.

[Grze00] A. Grzesiak: Konzeption einer Matlab-Toolbox für die qualitative Modellierungssprache SQMA, Studienarbeit SA-1758, IAS, Universität Stuttgart, 2000.

[Grze03] A. Grzesiak: Analysis of the applicability of SQMA to fault analysis, Diplomarbeit DA-1899, IAS, Universität Stuttgart, 2003.

[Heid96] R. Heider: Troubleshooting CFM 56-3 Engines for Boeing 737 Using CBR and Data Mining, in Smith and Faltings: Advances in Case-Based Reasoning, Lecture Notes in Artificial Intelligence, Springer Verlag, pp. 512-518, 1996.

[Heyl97] F. Heylighen: The Growth of Complexity, in The Evolution of Complexity, Kluwer, Dordrecht, 1997.

[Kata03] P. Katamaneni: Online Process Supervision of a Three Tank System, Master Thesis MT-1890, IAS, Universität Stuttgart, 2003.

[KeLe99] U. Keller, T. Wyatt, R. Leitch: FrenSi - a fuzzy qualitative simulation method, MISC Workshop Application of Interval Analysis to System and Control, University of Girona, 1999.

[Klir85] G. Klir: Architecture of Systems Problem Solving, Plenum Press, New York, U.S.A., 1985.

[KlYu95] G. Klir, B. Yuan: Fuzzy Sets and Fuzzy Logic: Theory and applications, Prentice Hall, Upper Saddle River, NJ (U.S.A.), 1995.

[Krol94] A. Kroll: Identifikation nichtlinearer Fuzzy-Modellen, Gerhard-Mercator-Universität-GH Duisburg, department of Measurement and Control Engineering, Research report No. 10/94, 1994.

[KrRe95] A. Kroll, H. Reuter, H. Hebisch: Zur experimentellen Fuzzy-Modellbildung eines pneumatischen Translationsantriebes, Gerhard-Mercator-Universität-GH Duisburg, department of Measurement and Control Engineering, Research report No. 13/95, 1995.

[KrSc98] V. Krebs, E. Schäfers: Dynamische Fuzzy-Systeme zur qualitativen Prozessmodellierung, Proceedings of the GMA Workshop Computational Intelligence, Berlin (Germany), pp. 115-136, March 03-04, 1998.

[KrSc00] V. Krebs, K. Schmid: Qualitative modeling using dynamic fuzzy systems, Proceedings of the 3rd International Symposium on Mathematical Modeling, Vienna (Austria), pp. 543-546, Feb. 2000.

[Kuip86] B. Kuipers: Qualitative Simulation, Artificial Intelligence, Vol. 29, pp. 289-338, 1986.

[Kuip94] B. Kuipers: Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge, MIT Press, ISBN 0-262-11190-X, 1994.

[Kuo80] B. Kuo, Digital control systems, in HRW series in electrical and computer engineering, Holt-Saunders International Edition, ISBN 4-8337-0013-1,1980.

[LaGö99a] R. Lauber and P. Göhner: Prozessautomatisierung 1, 3rd Edition, Springer-Verlag, Berlin (Germany), ISBN 3-540-65318-X, 1999.

[LaGö99b] R. Lauber and P. Göhner: Prozessautomatisierung 2, Springer-Verlag, Berlin (Germany), ISBN 3-540-65319-8, 1999.

[Lauf96] X. Laufenberg: Ein modellbasiertes qualitatives Verfahren für die Gefahrenanalysis, Ph.D. Thesis, Universität Stuttgart, 1996.

[Laud99] A. Laudwein: Konzeption und Entwicklung einer Steuerungs- und Regelungssoftware für den Modellprozess “Drei-Tank-System”, Diplomarbeit DA-1709, IAS, Universität Stuttgart, 1999.

[Lunz92] J. Lunze: Qualitative modelling of continuous-variable systems by means of nondeterministic automata, Intelligent Systems Engineering Vol. 1, pp.22-30, 1992.

[Lunz95] J. Lunze: Künstliche Intelligenz für Ingenieure. Band 2: technische Anwendungen, R. Oldenburg Verlag GmbH, Munich (Germany), 1995.

[LRP01] Landesarchivverwaltung Reinland Pfalz: Vor 80 Jahren; Der 21. September 1921. Explosionsunglück bei BASF, in Blick in die Geschichte (http://www.landeshauptarchiv.de/blick/), 2001.

[Manz98] S. Manz: Qualitative Modellierung einer Einspritzanlage für Dieselmotoren, Workshop: Einsatz qualitativer Modelle für die Überwachung und Steuerung technischer Systeme, Hamburg/Emmelndorf, 1998.

[Manz99] S. Manz: Qualitative Modeling of a Three-Tank-System. Interkama-ISA Tech Conference, Düsseldorf, 1999.

[Manz01a] S. Manz: Online Fault Detection and Diagnosis of Complex Systems based on Hybrid Component Models, 14th International Congress on Condition Monitoring and Diagnostic Engineering Management (COMADEM 2001), Cambridge (U.K.), ISBN 0-08-0440363, 2001.

[Manz01b] S. Manz: Fuzzy based qualitative models in combination with dynamical models for online monitoring of technical systems, International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA 2001), Las Vegas, U.S.A., 2001.

[MBC84] M. Marsan, G. Balbo, G. Conte: A class of generalized stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. ACM Transactions on Computer Systems, Vol. 2, No. 2, pp. 93-122, May 1984.

[Moor66] R. Moore: Interval analysis, Prentice-Hall, Englewood-Cliffs, NJ, U.S.A., 1966.

[Motr93] A. Motro: Sources of uncertainty in information systems, Proceedings of the 2nd Workshop on Uncertainty Management and Information Systems: From Needs to Solutions, Catalina Island (CA, U.S.A.), 1993.

[MaWo00] M. Malko, H. Wohlfarth: Gesundheitliche Schäden der weißrussischen Bevölkerung. Press release of the Tschernobyl-Initiative (ti.welcomes-you.com), Wolfenbüttel/Minsk, 2000.

[Natk80] S. Natkin: Reseaux de Petri Stochastiques, Ph.D. Thesis, CNAM-Paris, 1980.

[Nebo94] A. Nebot: Qualitative Modeling and Simulation of Biomedical Systems Using Fuzzy Inductive Reasoning, Ph.D. Thesis, Universitat Politècnica de Catalunya, 1994.

[Neap92] R. Neapolitan: A survey of uncertain and approximate inference, in L. Zadeh, J. Kacprzyk: Fuzzy Logic for the management of uncertainty, Whiley Profesional Computing, U.S.A., ISBN 0-471-54799-9, pp. 55-82, 1992.

[Nguy78] H. Nguyen: A note on the Extension Principle for Fuzzy Sets, Journal of Mathematical Analysis Applications, Vol. 2, No. 64, pp. 369-380, 1978.

[Pawl82] Z. Pawlak: Rough sets, International Journal of Computer and Information Science, pp. 341-356, 1982.

[Pawl92] Z. Pawlak: Rough sets: A new approach to vagueness, in L. Zadeh, J. Kacprzyk: Fuzzy Logic for the management of uncertainty, Whiley Profesional Computing, U.S.A., ISBN 0-471-54799-9, pp.105-118, 1992.

[Pawl94] Z. Pawlak: Rough sets, rough relations and rough functions, ICS Research Report 24/94, Warsaw University of Technology, 1994.

[Pedr84] W. Pedrycs: An identification algorithm in fuzzy relational systems, Fuzzy Sets and Systems, Vol. 13, pp. 153-167, 1984.

[Perm90] G. Permantier: Verfahren der Wissensrepräsentation bei der Entwicklung wissensbasierter Systeme für die Automatisierungstechnik, Ph.D. Thesis, Universität Stuttgart, 1990.

[Rebo95] M. Rebolledo: Reconciliación de datos utilizando Algoritmos Genéticos, Master Thesis, Universidad de Carabobo, 1995.

[Rebo02] M. Rebolledo: Development of a concept for the handling of vagueness in the SQMA modeling approach, Master Thesis MT-1832, IAS, Universität Stuttgart, 2002.

[Rebo03a] M. Rebolledo: Situationsbasierte Prozessüberwachung komplexer Systeme mit Rough-Sets, VDI-Berichte 1756, GMA-Kongress 2003 (GMA-03), Baden-Baden (Germany), ISBN 3-18-019756-3, pp. 245-252, 2003.

[Rebo03b] M. Rebolledo: A combined rough-stochastic approach to process monitoring based on qualitative models, Proceedings of the IASTED International Conference on Intelligent Systems and Control (ISC-03), Salzburg (Austria), ISBN 0-88986-355-5, pp. 69-74, 2003.

[Rena01] A. Renaut: Industrial risks: Negligence and outsourcing blamed, in Trade union world, No. 11, Brussels, Belgium, November, 2001.

[RiRü98] A. Riid, E. Rüstern: Comparison of fuzzy function approximators, *Proceedings of the 6th Biennal Baltic Electronic Conference, Tallinn, Estonia, pp. 139-142, Oct 7-9 1998.*

[Samb04] J. Sambeth: Zwischenfall in Seveso. Unionsverlag, Zürich, ISBN: 3-293-00329-X, 2004.

[ScKe99] D. Schaich, R. King, U. Keller, M. Chantler: Interval Identification - A modelling and design technique for dynamic systems, QR99 - 13th International Workshop of Qualitative Reasoning, Loch Awe Scotland, June 6-9, 1999.

[ShLe93] Q. Shen, R. Leitch: Fuzzy qualitative simulation, IEEE Transactions On Systems, Man, and Cybernetics, Vol. 23, No. 4, pp. 1038-1060, Jul./Aug. 1993.

[STQS00] L. Shen, F. Tay, L. Qu, Y. Shen: Fault diagnosis using rough sets theory. In Computers in Industry 43, Elsevier Science, pp. 61-72, 2000.

[SuYa93] M. Sugeno, T. Yazukawa: A fuzzy-logic-based approach to qualitative modeling, IEEE Transactions on Fuzzy Systems. Vol. 1, No. 1, pp. 7-31, Feb. 1993.

[Schn98] H-J. Schneider: Lexikon der Informatik und Datenverarbeitung, Version 4.0, Munich (Germany): R. Oldenburg Verlag. ISBN 3-486-24538-4, 1998.

[StDr89] P. Struss, O. Dressler: Physical negation: Integrating fault models into the general diagnostic engine, in Hamscher, Walter et al: Readings in model-based diagnosis, Morgan Kaufmann Publishers, San Mateo, CA, pp. 1318-1323, 1992.

[Tikk02] D. Tikk et al: Improvement and Critique on Sugeno's and Yasukawa's Qualitative Modeling, IEEE Transactions on Fuzzy Systems, Vol. 10, No. 5, October 2002.

[TMW98] The Math Works Inc: Fuzzy Logic Toolbox User's Guide – Version 2, Natick MA (U.S.A.), 1998.

[TMW99a] The Math Works Inc: Using MATLAB- Version 5, release 11, Natick MA (U.S.A.), 1999.

[TMW99b] The Math Works Inc: Using Simulink- Version 3, release 11, Natick MA (U.S.A.). 1999.

[Tris01] Trish: The Bhopal Incident: A Catalyst to Environmental Awareness, student paper, University of Maryland, 2001.

[UN04] UN Office for the Coordination of Humanitarian Affairs: History of the United Nations and Chernobyl, 2004, http://www.un.org/ha/chernobyl/history.html

[WaMe92] L-X. Wang, J. Mendel: Fuzzy basis functions, universal approximation, and orthogonal least-squares learning, IEEE Trans. Neural Networks, Vol. 3, No. 5, pp. 807-814, Sept. 1992.

[Will03] J. Wille: Development of a concept for the experimental determination of Rough Intervals, Diplomarbeit DA-1896, IAS, Universität Stuttgart, 2003.

[WLZW02] W. Wang, K. Liu, D. Zhou and C. Wang: A fuzzy and rough sets integrated approach to fault diagnosis, 15th Triennial World Congress, Barcelona, Spain, 2002.

[Worb98] M. Worboys: Computation with imprecise geospatial data, Computers, Environment and Urban Systems, Vol.22, No. 2, pp. 85--106, 1998.

[WuMa02] J. Wu and D. Marceau: Modeling complex ecological systems: an introduction. Ecological Modelling, No. 153, pp. 1-6, 2002.

[Zade65] L. Zadeh: Fuzzy Sets, in Information and Control, Vol. 8, pp. 338-353, 1965.

[Zade78] L. Zadeh: Fuzzy sets as a basis for a theory of the possibility, in Yager, Ovchinnikov, Tong and Nguyen: Fuzzy sets and applications: Selected papers by L.A. Zadeh, John Wiley & Sons, Inc, U.S.A., ISBN 0-471-85710-6, 1987. Original paper written in 1978.

[Zade83] L. Zadeh: The role of fuzzy logic in the management of uncertainty in expert systems, in Fuzzy Sets and Systems, No. 11, pp. 199-228, 1983.

[ZuZh94] R. Zurawski and M. Zhou: Petri Nets and industrial applications: A tutorial, in IEEE Transactions on Industrial Electronics, Vol. 41, No. 6, pp. 567-583, 1994.

Lebenslauf

Persönliche Daten:

28.05.1968	geboren in Maracay, Venezuela

Schulbildung:

1974 - 1980	Grundschule Grupo Escolar “Las Delicias” in Maracay
1980 - 1985	Gymnasium Liceo Militar “Los Próceres” in Valencia (Venezuela) - Abschluss als “Bachiller en Ciencias" (Abiturient)

Studium:

1985 - 1989	Studium der Elektrotechnik am “Instituto Politécnico de las Fuerzas Armadas Nacionales” Abschluss Ingenieur

Beruf begleitende Weiterbildung:

1989 - 1996	Studium der Systemtechnik an der “Universidad Nacional Abierta” (ohne Abschluss)
1991 - 1995	Masterprogramm Regelungstechnik in verfahrentechnischen Prozesse an der “Universidad de Carabobo” Abschluss Master-Ingenieur
1997 - 1999	Spezialisierung in Informationsmanagement an der “Universidad Tecnológica del Centro” Abschluss Spezialist in Informationsmanagement

Berufstätigkeit:

1990	Mitarbeit bei Electrocomp als Elektrotechniker (Ingenieur)
1990 - 1991	Mitarbeit bei Orión Electrónica als Produktionsmanager
1990 - 1991	Dozent für I.U.T. “Isaac Newton” – Elektrotechnik und PASCAL
1991	Mitarbeit bei Elsag-Bailey (ABB) im Außendienst, Automatisierungssysteme
1991 - 2000	Angestellt bei “Petroquímica de Venezuela”, PDVSA als Mitarbeiter in den Abteilungen Automatisierung, Prozessregelung und -steuerung, Wartung und Instandhaltung von Automatisierungssystemen, sowie in verschiedenen Automatisierungs- und IT-Projekten
2001 - 2004	Wissenschaftlicher Mitarbeiter am Institut für Automatisierungs- und Softwaretechnik der Universität Stuttgart