Multisensor Data Fusion Implementation for a Sensor Based Fertilizer Application System

R. Ostermeier[1], H. I. Rogge[2], and H. Auernhammer[1]

[1]Technische Universität München, Crop Production Engineering, Am Staudengarten 2, 85354 Freising, Germany.

[2]Universität Hohenheim, Institute of Agricultural Engineering, Germany.

E-Mail: ralph.ostermeier@wzw.tum.de

ABSTRACT

"Mapping systems" ("mapping approach"), real-time sensor-actuator systems ("sensor approach") or the combination of both ("Real-time approach with map overlay") determine the process control in mobile application systems for spatially variable fertilization. Within the integrated research project "Information Systems Precision Farming Duernast" (IKB Duernast) the implementation of the "Real-time approach with map overlay" was done for intensive nitrogen fertilization. The bottom line of this sophisticated approach is a comprehensive situation assessment, a typical multisensor data fusion task. Based on a functional and procedural modelling of the multisensor data fusion and decision making process, it could be pointed out that an expert system is an adequate fusion paradigm and algorithm. Therefore, a software simulation with an expert system as core element was implemented to fuse on-line sensor technology measurements (REIP), maps (yield, EM38, environmental constraints, draft force) and user inputs in order to derive an application set point in real-time. The development of an expert system can be viewed as a structured transformation in five levels from the "specification level", the "task level", the "problem solving level" and the "knowledge base level" to the "tool level". In the "tool level" the hybrid expert system shell JESS (Java Expert System Shell) was selected for implementation due to the results of preceding levels. Knowledge acquisition was done within another IKB-subproject by the means of data mining. Typical and maximal times of 10 ms and 60 ms for one fusion cycle were measured running this application on a 32-bit processor hardware (Intel Pentium III Mobile, 1 GHz).

Keywords: Data Fusion, Expert System, Multisensor, Process Control, Real-time Approach with Map Overlay, Sensor Fusion, Site-specific Fertilization

1. INTRODUCTION

Three different system approaches determine the process control in mobile application systems for spatially variable fertilization. These are "mapping systems" ("mapping approach"), real-time sensor-actuator systems ("sensor approach") or the combination of both ("Real-time approach with map overlay"). Mapping approach and sensor approach have disadvantages depending on the system, however the "Real-time approach with map overlay" may overcome the disadvantages of both (Auernhammer, 2001). In principle, the basic idea of this approach is to lead a process or system, here plants and their surroundings, to an ecological and economic optimum. This requires information about the current state of the process and its inputs, i.e. "precision farming maps" and on-line sensor technology process data. The possibility for intervention on the process is fertilization. Thereby, the application set point is derived by expert knowledge and the input information at hand. Documentation completes the procedure. In summary, this means the current situation for each local site must be investigated and assessed. Thereupon, action (application set point) has to be derived.

From the information technology point of view, this system approach is a topic of multisensor data fusion und requires appropriate methods and terminology. Basically, multisensor data fusion systems can be analyzed and modelled by a functional model and a process model (Steinberg and Bowman, 2001). A functional model should describe at the highest abstraction level what analysis functions or processes need to be performed. While a process model describes at a high level of abstraction how this analysis is accomplished. Based on this abstract view of demands, requirements and problem-solving paradigm, a system architecture (high level abstraction of hardware - software implementation) can be designed. Established and appropriate systems engineering methods have to be applied for further transformation of this system architecture into a concrete technical implementation by hardware and software.

Ostermeier and Auernhammer (2004) defined a complete theoretical framework with a functional model, process model and system architecture for the “Real-time approach with map overlay” and applied it to a real-time process control for a sensor based fertilizer application system within the integrated research project “Information Systems Precision Farming Duernast” (IKB Duernast) (Auernhammer *et al.*, 1999). From a functional point of view the “Real-time approach with map overlay” can be completely specified according to the revised JDL data fusion model. This model was specified by the Joint Directors of Laboratories (JDL) and was revised by Steinberg and Bowman (2001) (fig. 1) in 1998.

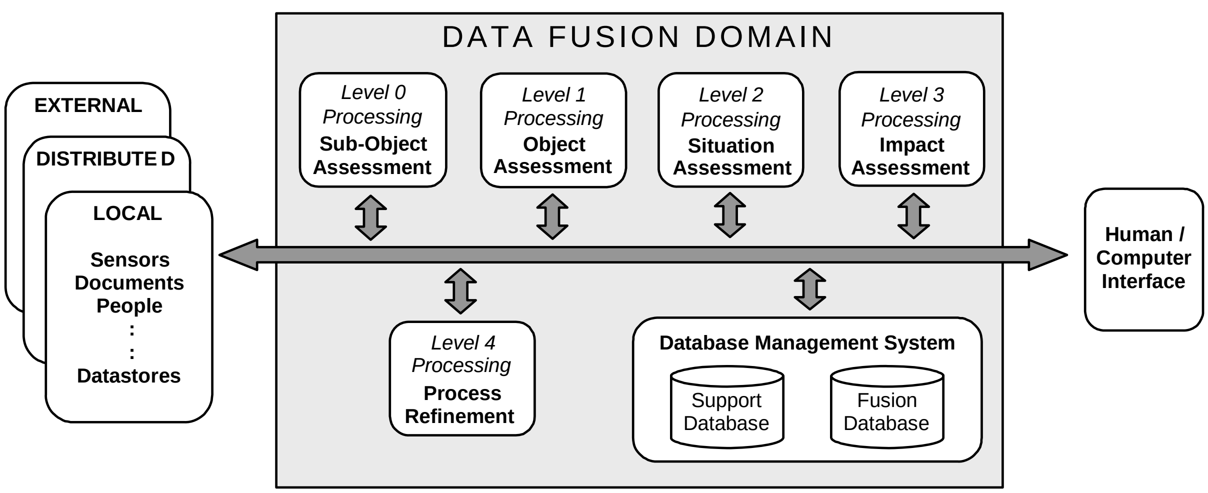

Figure 1. Revised JDL data fusion model (Steinberg and Bowman, 2001).

Due to noncommensurate information sources, there is no fusion at raw data level possible, but fusion at feature/state or decision level is demanded. Although the “Real-time approach with map overlay“ comprises all JDL functional levels, main focus is put on Level 2 Processing - Situation Assessment within the IKB Duernast project, since the bottom line is a comprehensive situation assessment, i.e. an assessment of current on-line sensor technology measurements with context-sensitive interpretation. Ostermeier and Auernhammer (2004) derived by the means of

Antony's (1995) process model an appropriate data fusion algorithm for the Level 2 Processing. A conventional expert system with its forward-inference production rule paradigm was determined as a suitable problem solving paradigm. Ostermeier *et al.* (2003) stated the proposal of an in-field controller in order to integrate this approach into ISO 11783 (ISO, 1998) compliant data communication systems. The important role of ISO 11783 for networking on farm level is emphasized by Munack and Speckmann (2001).

2. OBJECTIVES

Within the integrated research project IKB Duernast, the "Real-time approach with map overlay" for intensive nitrogen fertilization should be implemented in form of a personal computer (PC) based simulation including an intuitive man machine interface. The PC based simulation should be confined on the second nitrogen application. Therefore data of on-line sensor technology (vegetation index: REIP) and data of "precision farming maps" (historic yield of the year 1998, EM38 measurement, soil draft force and applied fertilizer rate of the first application in the same year), originating from the IKB-Duernast Farm Management Information System (FMIS), and environmental protection restrictions should be fused in real-time.

Main focus should be put on the implementation of the basic multisensor data fusion algorithm for JDL level 2 processing "Situation assessment" with resulting derivation of control action. As already stated in the introduction a suitable problem solving paradigm is a conventional expert system with its forward-inference production rule paradigm. In an effort to implement technical systems efficiently and goal-oriented, an integrated specification and design process is an essential element and plays an important role in successful translation of requirements into physical implementation. This should be especially pointed out for the implementation of the expert system within the simulation development.

3. MATERIALS AND METHODS

3.1 Architecture of an Expert System

The architecture of an expert system is well described in several artificial intelligence textbooks, e.g. according to Sriram (1997) an expert system comprises the main elements inference mechanism, knowledge base, context, knowledge acquisition facility, explanation facility and user interface.

3.2 Development Process of an Expert System

According to Luger (2004) it is necessary for the development of an expert system as in the case of many other artificial intelligence programs to differ from the conventional process and to create a prototype quite early and to work out gradually and to optimize the code. An explorative development cycle is therefore the suitable engineering method. An explorative development process, however, does not absolve the developer from a structured procedure.

The major steps of an evolutionary development process are depicted in figure 2. The identification phase describes what are problem domain characteristics, the scope, who will participate and what the resources are. In the conceptualization step needed information and techniques to solve the problems and tasks are identified. In the formalization phase the concepts should be put in a formalized representation. This involves the selection of a knowledge representing scheme and the appropriate tools for building the expert system.

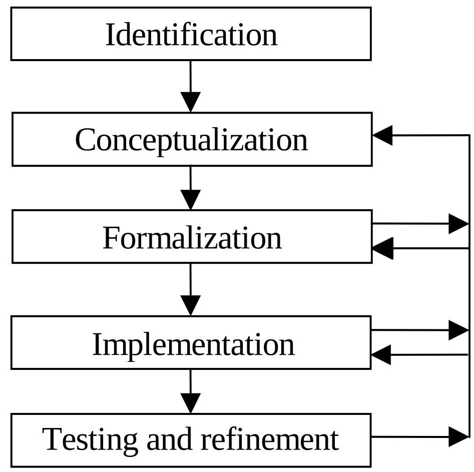

Figure 2. Explorative development process of an expert system (according to Sriram (1997)).

The implementation step comprises encoding the knowledge obtained in the previous step into software and hardware - a prototype. In the testing and refinement phase the capacity of the prototype is inspected. The problem solving capacity should be analyzed, as well as errors and the weaknesses, so that they could be eliminated.

In order to work straightforward, one part of the identification phase and the whole conceptualization and formalization phases can be supported by an appropriate method described by Sriram (1997). He defines the development of a knowledge based system as a structured transformation from a “specification level” to a “tool level”. So the one part of the identification phase and the whole conceptualization phase can be achieved by the first three steps “specification level”, “task level” and “problem solving level”. The formalization phase can be accomplished through the transformation from the “knowledge base level” to the “tool level”. In the following section there is a brief description of the five levels, which are necessary to determine the required tool, conducted:

- Specification level: The problem (derivation of the required nitrogen application rate), its constraints and the solution are described in this level.
- Task level: The “task level” describes problems, which engineers solve, such as diagnosis, planning, simulation, interpretation, control, monitoring, etc. The type of expert system to be chosen will have to cover the aspects of the problems.
- Problem solving level: In this level a selection of the knowledge evaluation and control mechanisms is conducted. These mechanisms are used to enable and accelerate the problem solving.
- Knowledge base level: In this level a selection of the possible knowledge representation is conducted.
- Tool level: In this level the programming paradigm(s) for implementation has to be chosen based on the previous analyses.

3.3 Knowledge Acquisition

Knowledge acquisition is one of the most significant parts for the implementation of an expert system. It is the process of eliciting domain knowledge from experts or textbooks for encoding in an appropriate tool. The spectrum of knowledge acquisition modes ranges from manual techniques to fully automated techniques. For this project, the agricultural engineering knowledge was acquired from textbooks and discussions with colleagues of the Technische Universitaet Muenchen subject group Crop Production Engineering and encoded manually. To elicit the crop production knowledge the following procedure was carried out. The knowledge acquisition was conducted in close cooperation with the IKB subproject 12 “Deriving of Decision Support Rules for Site Specific N-Fertilization” (Weigert and Wagner, 2003). This subproject developed the crop production knowledge by the application of data mining techniques and provided all the necessary field trial test data (on-line sensor technology measurements and overlay maps) for the simulation. Therefore research was conducted at fields located in Freising, Germany.

4. RESULTS

According to the presented method, the implementation of the simulation is explained in the following, starting with conceptualization and formalization since identification is mainly important for project management and here technical aspects should be in the focus.

4.1 Conceptualization and Formalization

4.1.1 Specification Level

The problem, its constraints and the basic solution approach by multisensor data fusion have been already described in the chapters introduction and objectives. So, only one main real-time requirement has to be determined. The whole process of gathering, transferring and processing data, plus the reaction time of the actuator (fertilizer spreader), has to be performed within the available time. Therefore the expert system's data processing, i.e. one data fusion and decision cycle, is limited by the time. The total time available for sensing, data processing and adjusting the fertilizer spreader actuator is calculated using the following equation:

$$t_T = s / v \qquad (1)$$

where:

t_T = total available time
s = total distance from starting point of sensor view to point of fertilizer application
v = velocity of the tractor-implement combination

Assuming a tractor-implement combination driving with a velocity of 4.0 m/s, using equation 1 and a total distance of 6.5 m the total time available is as shown below:

$$t_T = 6.5 \text{ m} / (4.0 \text{ m/s}) = 1.6 \text{ s}$$

The partial time t_P required for data communication and signal processing in the sensors (~ 30 ms) as well as the operating time of the fertilizer spreader (~ 650 ms; measurements at Technische Universitaet Muenchen subject group Crop Production Engineering in order to determine minimal response times of a fertilizer spreader actuator with modified control) can be determined to 680 ms. To determine the time available to the expert system t_{XPS}, these 680 ms have to be subtracted from the result of equation 1, as shown below:

$$t_{XPS} = 1.6\,s - 0.68\,s = 0.92\,s$$

4.1.2 Task Level

Sriram (1997) characterizes the "task level" as follows. "The range of expert system applications spans from derivation to formation problems. In derivation problems, the problem conditions are specified as parts of a solution description, i.e. the possible outcomes exist in the knowledge base. So the solution to these problems involves the identification of the solution path. In formation problems, problem conditions are given in the form of properties that a solution must satisfy as a whole. An exact solution(s) does not (normally) exist in the knowledge-base, but the inference mechanism can generate the solution by utilizing knowledge in the knowledge base."

It is obvious, that the "Real-time approach with map overlay" can be best described by monitoring and control. These tasks tend to be more derivative. In order to enable site specific fertilization, the process control has to be conducted within a given time. For this it is advantageous if main parts of the solution already exist in the knowledge base and guarantee a deterministic behavior in certain limits.

4.1.3 Problem Solving Level

The process of problem solving is a search in a solution realm (Barr and Feigenbaum, 1981). The number of possible derivations from the beginning to the target is the solution realm. Artificial intelligence uses steps of action for the search. In each step a number of solutions is possible. Some facts and rules may lead to a "combinatory explosion" in the steps of action for the search. Conflicts may also occur, when more than one rule is applicable. The objective is to search for the fastest way to get the answer. Therefore some typical problem solving techniques and architectures have proven to be useful. They can be subdivided into knowledge centered and search centered. A combination of both is appropriate for this task.

4.1.3.1 Knowledge Centered Problem Solving Techniques and Architectures

- Goal and data:
  As stated in the introduction an appropriate problem-solving approach for the data fusion algorithm for the Level 2 Processing was derived by the means of Antony's (1995) process model. Canonical problem solving form IX with its forward-inference production rule paradigm was determined to be appropriate. This is a generation-based algorithm and supports the selection of forward chaining out of the realm of knowledge-centered problem solving techniques and architectures. Forward chaining is a data driven reasoning, which means that the reasoning starts with known data from the data bases. The search goes from the premises to the conclusions. In the case of the derivation of the application rate by the expert system the symptoms are available (map and sensor data) and the system has to infer from them (data driven) to reach the solution (hypotheses that are the number of all possible application rates).

- Inexactness:
  Dealing with another knowledge-centered parameter the consideration of inexactness was done according to the following premises. Data coming from the overlay maps were already corrected by mapping correction algorithms like proposed e.g. by Blackmore (1999), Noack *et al.* (2003) or Bachmaier and Auernhammer (2005), i.e. some kind of JDL level 1 processing, and therefore do not need a special treatment. On-line sensor technology measurements have to be corrected either by JDL level 1 processing (e.g. default values, simple moving average, Kalman filter) or at JDL level 2 processing using a different rule set when values (attributes) are missing.

- Modularization:
  The functional modeling showed that the knowledge can be modularized. So it is obvious to structure the knowledge base in four rule set modules: crop production assessment, constraints effects, agricultural engineering assessment and evaluation & decision.

- Conflict resolution:
  Good rule based system design avoids the use of salience (giving rules a higher priority) and a strict processing control otherwise it would be better to select another problem solving paradigm. One important issue is to design a decision tree with ensured consistency, but there are cases with no or more solutions. To deal with these situations each of the four rule set modules has got a second decision stage of assessment (sum up) where rules were implemented to select the most appropriate solution or to insert a default value. On a more elementary stage, conflict resolution is done by the following search centered problem solving technique and architecture.

4.1.3.2 Search Centered Problem Solving Techniques and Architectures

Two common simple search strategies are the depth-first and breadth-first search. A depth-first search explores a path until the last level, before backtracking and exploring another path. A breadth-first search explores the nodes closest to the root, before exploring the ones in the next level. The depth-first search is more efficient as the breadth-first in situations where the required statement is not close to the root node and the branching factor is not small. For the case at hand it was identified that to represent the relations of premises and conclusions defined by the rules, a tree with deep levels (many input data types) is required. Therefore the depth-first search is recommended as the main search mechanism for the task.

4.1.4 Knowledge Base Level

Usually expert system applications deal with symbolic knowledge. Typical knowledge representations for this kind of knowledge are: Production rules, semantic networks, frames, scripts, logic and model based representation.

Weigert and Wagner (2003) showed within the IKB research project that decision support for site specific N-fertilization, i.e. defining the application rate, could be represented by production rules. These rules can be depicted as decision trees. All the crop production knowledge is formulated as a decision tree with a deep level of 10 at maximum. The resulting recommendations for an application set point are discrete steps of 10 kg/ha and range from 0 to 100 kg/ha. Also the knowledge about the constraints, like repeated application (overlapping) or limitations due to environmental regulations and user input in order to command a relative adjustment of the set point, are formulated as rules. In the same way the agricultural engineering assessment can be defined for the selection of the operating mode (manually or automatically) and the analysis of the "health state" of the implement and on-line sensor technology.

The facts are also a part of the knowledge base of an expert system and have an influence on the selection of an appropriate tool. Facts (data) should represent "precision farming maps", on-line sensor technology measurements and characteristics and different components and parameters of the tractor implement combination. A well adapted and intuitive representation for these facts is an object-oriented approach.

4.1.5 Tool Level

The main software tools for the development of expert systems range from conventional programming languages up to hybrid shells. Conventional programming languages, e.g. C or Fortran, can be used for the development of an expert system, but they are not the best choice since they do not support the construction of expert systems. Artificial Intelligence programming languages were developed specifically for the construction of expert systems. They can be classified as rule-based, logic-based, network-based and frame-based, according to the representation formalism applied. Examples are Prolog and OPS5. Hybrid artificial programming languages were also developed to build expert systems. Their main characteristic is to provide different programming paradigms and several different forms of inference strategies in one framework. Examples are Kappa, KEE, and Nexpert. Shells are expert systems with an empty knowledge base, built for a specific purpose. The programmer simply needs to insert the knowledge into the knowledge base. One disadvantage of using shells is the loss in flexibility, due to the fact that the knowledge engineer has to work with the given characteristics of the tool. Hybrid Shells provide in addition to a simple shell the possibility of using different knowledge representations and inference strategies different in one framework.

Since project time and funding was limited it would not be very useful to develop a whole expert system by using conventional or artificial intelligence programming languages if a shell could be found which fulfills all the requirements. The shell has to support forward chaining, depth-first searching, rule based knowledge representation, rule set modularization and object oriented representation of facts. Usually this variety is best fulfilled by a hybrid shell. The "real-time approach with map overlay" requires data processing in real-time according to the specification level, thus an expert system capable of processing the data in real-time is indispensable. Further main criteria for the selection were the development environment, programming interfaces, support and costs.

Based on those premises and especially due to the guaranteed real-time capability G2 (Gensym Corporation, USA) was the ideal candidate but the costs exceeded the research subproject budget. So JESS (Java Expert System Shell) (Sandia National Laboratories) (Friedman-Hill, 2003) was chosen. JESS is free for academic and non commercial use. It is a rule based hybrid expert system shell written in Java™ (Sun Microsystems Inc.), inspired by CLIPS, which can create, manipulate and reason about Java objects. JESS offers bi-directional chaining and uses a very fast matching algorithm (RETE) to process rules. This algorithm uses a data structure, which enables a fast match-recognize-act cycle, but it does not assure a response within a guaranteed time. The performance of a RETE based expert system depends principally on the number of partial matches generated by the rules, rather than on the number of rules and facts.

4.2 Implementation

The implementation was carried out on different notebook computers with the operating systems Microsoft Windows 2000 Professional or Microsoft Windows XP Home. Based on these hardware and operating system combinations a Java 2 SE Virtual Machine and the expert system shell JESS (Version 6.14) were installed. The rough system design of the simulation is depicted in figure 3.

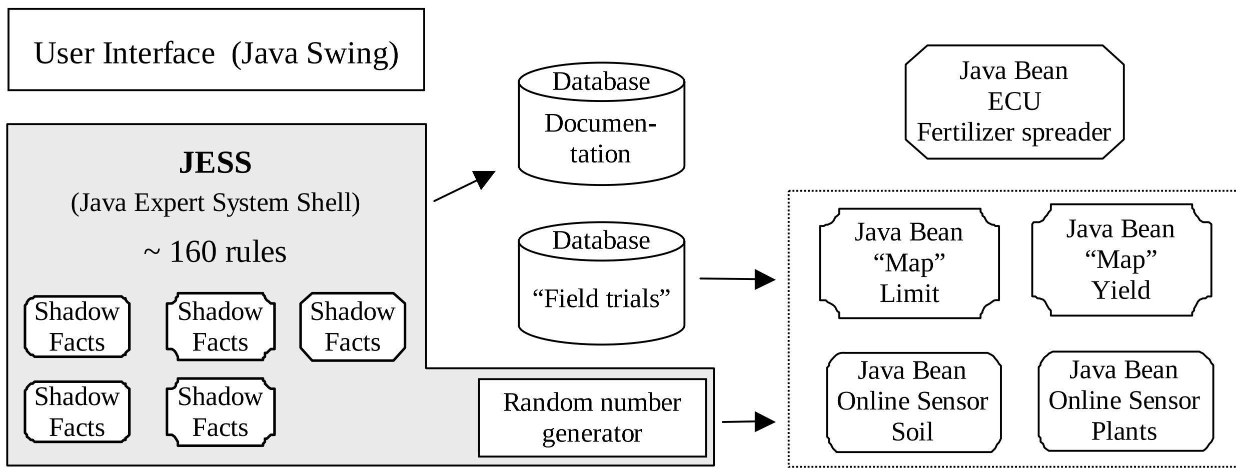

Figure 3. Schematic overview of the simulation.

The whole rule set comprises about 160 rules and is organized in four different components (modules). The first component covers rules for crop production assessment (fig. 4). The second component covers rules, which are processing predefined constraints. The third component consists of rules for an agricultural engineering assessment, which take e.g. prioritized user inputs or the "health state" of the on-line sensor technology into account. The fourth component sums up the results of the first three components and takes a final recommendation for the application set point. Furthermore some rules are also implemented for control of sequence of operations, cyclic triggering and module focus change.

Use of the object oriented software component technology JavaBeans (Sun Microsystems Inc.) allows to simulate the process surroundings and to get the facts for the expert system at the same time. For this purpose shadow facts are created in JESS, in which the attributes of an on-line sensor technology, overlay map or tractor implement combination component, each imitated by a JavaBean, are mirrored. The test data of the JavaBeans can be simulated by a random number generator or read from a database as real field trial data. The database was implemented using the open source database product MySQL. The read and write access was programmed as Java object, which can be integrated into JESS. The fertilizer spreader is also implemented as a JavaBean, the data flow direction only turns round. The derived application set point is put in the fertilizer spreader shadow fact and reflected to the fertilizer spreader JavaBean. In parallel, this set point is stored in the database for documentation purposes. Time measurement data about the period of an assessment and decision cycle are also stored in the database as well as the database access times. Therewith all relevant multisensor data fusion and documentation processes of the real-time process control for a sensor based fertilizer application system have been implemented.

```
(defrule CROP_PRODUCTION::N40-2 "Recommendation N40  Rule 2"

   (reip2 (plant_attribute ?p1&: (<= ?p1  722.25000)))
   (yield (soil_attribute ?s1&: (>  ?s1  8.577623)))
   (tforce (soil_attribute ?s2&: (>  ?s2  34.00000)))
   (yield (soil_attribute ?s3&: (<= ?s3  9.462849)))
   (reip2 (plant_attribute ?p2&: (>  ?p2  721.82001)))
   (tforce (soil_attribute ?s4&: (>  ?s4  30.00000)))
   (n1 (implement_attribute ?i&: (>  ?i  50.00000)))
   (reip2 (plant_attribute ?p3&: (>  ?p3  721.62000)))
   (reip2 (plant_attribute ?p4&: (<= ?p4  722.45001)))
   (yesREIP)
   (trigger)

   =>

   (assert (CROP_PRODUCTION::N_recommendation
   (setpoint 40)
   (explanation "40 kg/ha since rule < N40-2 >"))) )
```

Figure 4. Exemplary of a rule in JESS - Code.

Since JESS has got only a command line based I/O (Input/Output) interface an interactive user interface (fig. 5) was developed using Java’s GUI (Graphical User Interface) library Swing.

Both, the overlay map values and the on-line sensor technology values can be adjusted by the sliders. At the same time these sliders have a display function. At the “N-set point” panel this is the only functionality, for all other measurement values the display functionality is important when test data are created by a random number generator or read from the database. At the “User Controls” panel the user can command own decision and over-steering wishes, like e.g. the selection between manual or automatic operation or a proportional set point over-controlling. Furthermore certain adjustments also can be carried out. So the online sensor (REIP) can be switched off (fault simulation and simulation of the mapping approach respectively). The expert system recognizes this new situation and uses an alternative set of rules for the crop production assessment.

Without an explanation interface an expert system is incomplete. This interface is implemented by two components, the central panel and the “Explanation” panel. The central panel can interpret and display HTML (HyperText Markup Language) pages. In the currently implemented revision the user can overlook the whole process control and rule set in a graphical form (overview screen, decision trees and mind maps). Furthermore the inference sequence and the therefore needed time of each cycle are listed in the “Explanation” panel.

4.3 Testing and Refinement

Testing which means verification and validation is a critical element of expert system development. It is hard to verify these systems since specifications are non-existing or fuzzy and changing due to the evolutionary development process. So validation, i.e. the program performs the functions it is intended for, takes the centre stage. The following two evaluation measures were especially tested and ascertained.

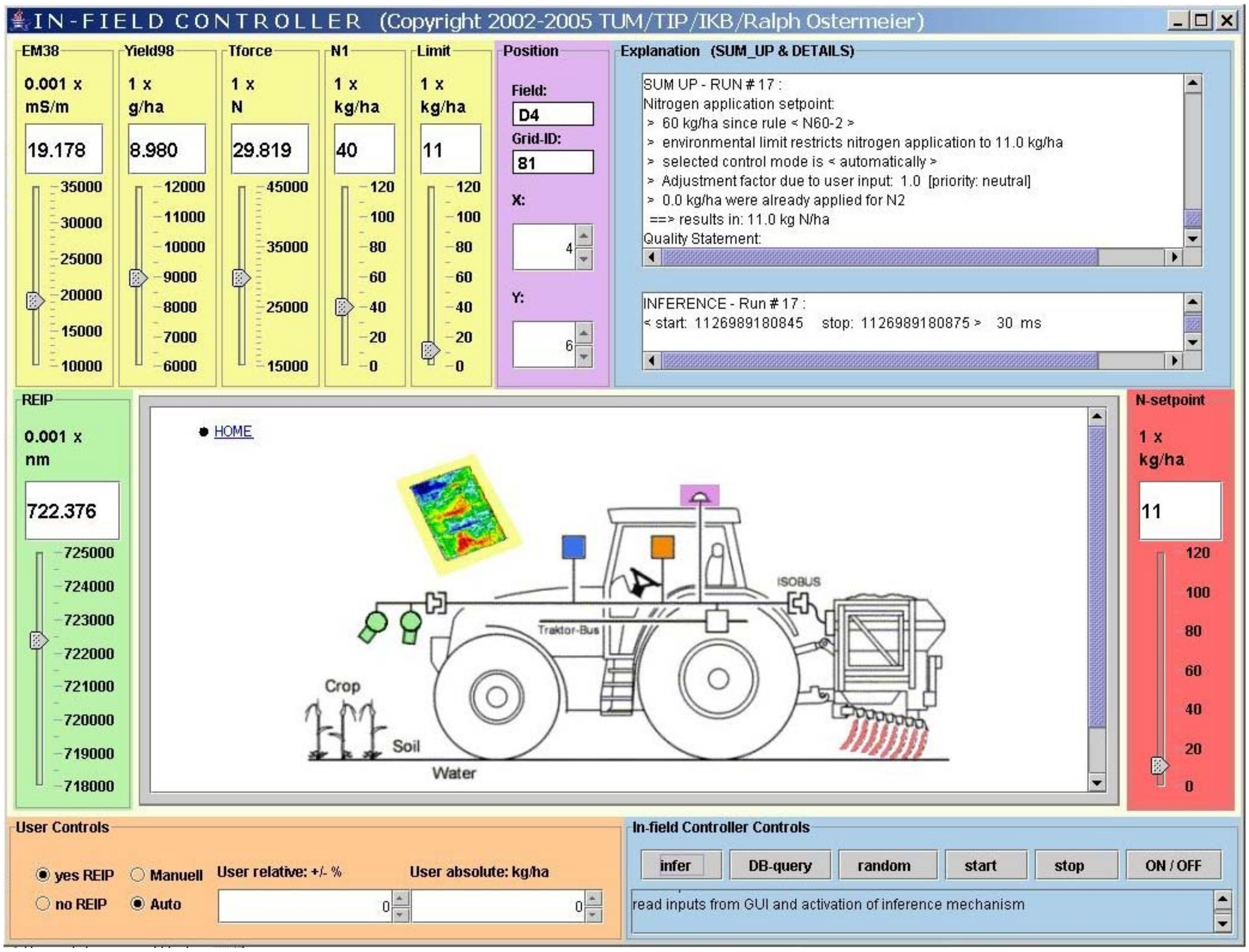

Figure 5. User interface of the simulation.

Within IKB Duernast subproject 12 the automatically by data mining generated decision trees were discussed with other crop production experts of the research group. So for field trial data the resulting application set point vector could be determined and used as a reference for testing the expert system in an automatic way. In order to cover also extreme value ranges and to test the constraints and the influence of user inputs, interactive examination (manual testing) of the system and an assessment by colleagues was conducted. Finally the system was presented to a broad audience at exhibitions and at the IKB Duernast research group's final colloquium.

Besides testing the correctness of the implemented rule set the computational performance regarding the real-time capability was carried out. Thousands of test data were generated using the random number generator feature of the simulation and were processed by the data fusion and decision functionality of the implemented expert system. The results for the typical (~ 85% of all passes) and maximum processing times for one assessment and decision cycle are listed for different hardware configurations in table 1. If 40 ms for the database read and write access are added in the worst case of 90 ms then the maximal available time of 0.92 s is only make used by 14%, i.e. the system is real-time capable within the defined boundary conditions and still offers growth potential.

Table 1. Processing times for one assessment and decision cycle of the simulation.

Index	Processor	Memory RAM	Operating System	Processing time typical	Processing time maximal
		MB		ms	ms
1	Intel Pentium III Mobile, 1.0 GHz	640	Microsoft Windows 2000 Professional	10	60
2	Intel Pentium III Celeron, 0.8 GHz	256	Microsoft Windows XP Home	20	90
3	Intel Pentium III Celeron, 1.5 GHz	256	Microsoft Windows XP Home	10	60

5. CONCLUSION

Within the integrated research project IKB Duernast, the “Real-time approach with map overlay”, a typical multisensor data fusion task, could be implemented in form of a personal computer based simulation including an intuitive man machine interface. Therefore a software simulation with an expert system as core element was implemented to fuse on-line sensor technology measurements (REIP), maps (yield, EM38, environmental constraints, draft force) and user inputs in order to derive a nitrogen application set point in real-time. An integrated development process could be established and exemplarily shows the implementation of a multisensor data fusion system for an agricultural engineering process control.

The explorative development process of the expert system can be viewed as a structured transformation in five levels from the “specification level” to the “tool level”. An analysis according to the five transformation levels, which was based on the functional and procedural modeling of the whole multisensor data fusion process, revealed that the hybrid expert system shell JESS (Java Expert System Shell) is an appropriate programming paradigm for implementation of the simulation. Although JESS possesses no hard real-time capability, carried out performance measurements showed that the simulated process control is soft real-time capable on a 32-bit processor hardware (Intel Pentium III Mobile, 1 GHz) and a Microsoft Windows 2000 or XP operating system.

Certainly the quality of the whole process control depends on the quality of the set of rules and of the facts. In this area further interdisciplinary research with colleagues from plant sciences, agricultural economics and ecology should be aspired. Additional further research is necessary for an enhanced inexactness treatment, for performance increase by the use of parallel architectures and a successful transfer to practical application needs techniques for an automatic maintenance of the knowledge base.

6. ACKNOWLEDGEMENTS

This work was part of the integrated research project “Information Systems Precision Farming Duernast” IKB Duernast (http://ikb.weihenstephan.de) which was financially promoted by the “German Research Council DFG”.

7. REFERENCES

Antony, R. T. 1995. Principles of Data Fusion Automation. Boston: Artech House.

Auernhammer, H., M. Demmel, F. X. Maidl, U. Schmidhalter, T. Schneider, and P. Wagner. 1999. An On-Farm Communication System for Precision Farming with Nitrogen Real-Time Application. ASAE-Paper No. 991150. St. Joseph, Mich., ASAE.

Auernhammer, H. 2001. Precision farming - the enviromental challenge. Computers and Electronics in Agriculture 30: 31-42.

Bachmaier, M., and H. Auernhammer. 2005. Yield mapping based on robust fitting paraboloid ones in butterfly and elliptic neighborhoods. In Precision Agriculture '05, ed. J. V. Stafford, 741-750. Wageningen, Netherlands: Wageningen Academic Publishers.

Barr, A., and E. Feigenbaum. 1981. The Handbook of Artificial Intelligence. Volume 1. Los Altos, California: William Kaufmann, Inc.

Blackmore, S. 1999. Remedial Correction of Yield Map Data. *Precision Agriculture* 1/1999: pp. 53-66.

Friedman-Hill, E. 2003. JESS in Action. Greenwich: Manning Publications Co.

ISO. 1998. *ISO 11783 - Tractors and machinery for agriculture and forestry - Serial control and communications data network*. Part 1 to 13. Geneva, Switzerland: ISO, International Organization for Standardization.

Luger, G. F. 2004. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 5th. edition. New York: Pearson Education - Addison-Wesley.

Munack, A., and H. Speckmann. 2001. Communication Technology is the Backbone of Precision Agriculture. *Agricultural Engineering International: the CIGR Ejournal* Vol. III.

Noack, P. O., T. Muhr, and M. Demmel. 2003. An Algorithm for Automatic Detection of Defective Yield Data. In Precision Agriculture, eds. J. Stafford and A. Werner, 445-450. Berlin, Germany: Wageningen Academic Publishers.

Ostermeier, R., H. Auernhammer, and M. Demmel. 2003. Development of an in-field controller for an agricultural bus-system based on open source program library lbs-lib. In Precision Agriculture, eds. J. Stafford and A. Werner, 515-520. Berlin, Germany: Wageningen Academic Publishers.

Ostermeier, R., and H. Auernhammer. 2004. Real-time process control for a sensor based fertilizer application system using multisensor data fusion. In *Proc. AGENG LEUVEN 2004, Engineering the future, CD-ROM (full papers), Session 10 -Nr. 352-Ostermeier.pdf*, 1-8. Leuven, Belgium, 12-16 September.

Sriram, R. D. 1997. Intelligent systems for engineering. Berlin, Heidelberg, New York: Springer-Verlag.

Steinberg, A. N., and Ch. L. Bowman. 2001. Revisions to the JDL Data Fusion Model. In Handbook of multisensor data fusion, eds. Hall D. L. and J. Llinas, 2-1 - 2-19. Boca Raton: CRC Press LLC.

Weigert, G., and P. Wagner. 2003. Development of decision rules for site-specific N fertilization by the application of data mining techniques. In Precision Agriculture, eds. J. Stafford and A. Werner, 445-450. Berlin, Germany: Wageningen Academic Publishers.