Proc. Estonian Acad. Sci. Phys. Math., 1993, 42, 1, 77—89
https://doi.org/10.3176/phys.math.1993.1.08

UDC 681.3:007 **To my teacher and friend**

Boris TAMM *

SOME FRAGMENTS OF EXPERT SYSTEM TECHNOLOGY

Pure art terminates where science starts. Very often we cannot determine whether we are on one or the other side of the border. **Expert System Technology,** being one of the problem areas of what we call Artificial Intelligence, is too new, too fuzzy, too vague to be defined as to which side it actually belongs to. Yět, the technology of **knowledge-based systems** is equally applicable to a factory, to an office, or to any other area where people deal with managing or control problems, where they have to make decisions, carry out plans and projects.

The situation under consideration occurs when the decision-making process can not be analytically determined, when no clear algorithm can be written, where we have to do with some uncertainty in specifying the process properties, variables or goals.

The human performance in these cases involves the skillful use of **heuristics** — certain rules of thumb — and expert experience coupled with analytical and other well-determined methods, integrating them into one single intelligent system.

Knowledge Acquisition

In these situations we can distinguish between at least three peculiarities which separate ordinary information or control systems from intelligent reasoning systems.

First. Instead of data base we need a **knowledge base,** which means that besides structured data we have to know the relations between the data object and the computation models for handling the relations and data in certain problem-solving episodes.

Second. We need expert knowledge for structuring an appropriate model of our **problem domain.** This is the case where the ordinary mathematical models (set of analytical equations, statistical tables, etc.) do not work.

Third. Instead of predetermined logic, however complicated, we need an **inference engine** for obtaining reliable results, which in most cases is an output for the user — a human operator — to consider.

So obviously we need expert knowledge for all the three: structuring the knowledge base, the problem domain and the particular inference engine. The process of getting such information from the expert as well as storing it in the computer, is called **knowledge acquisition.** And this is the most difficult and time-consuming part of any expert system project.

Very often knowledge acquisition starts from making a survey about the problem area, about the assumed knowledge base and about the robust decision mechanism or rule base. The result of this stage is a first structured iteration of a robust model of the expert system.

---

* Eesti Teaduste Akadeemia Küberneetika Instituut (Institute of Cybernetics, Estonian Academy of Sciences). Akadeemia tee 21, EE-0026 Tallinn, Estonia.

Structuring

After having formulated a somewhat structured survey, the expert knowledge acquisition process continues being now already concentrated on more precise structuring of different parts of the system to be prototyped. The usual start of **structured analysis** is often the preliminary specification of the systems' components. The process of building a knowledge base begins by watching experts make decisions and distilling out of that performance the factors that go into this process. Deriving a rough set of decision factors relevant to the domain of expertise and creating some general rules, is the result of the so-called pre-prototype version of the knowledge base. Usually this phase must be repeated several times, as the versions have to be corrected, updated, changed [1].

It is important to emphasize that in expert system design we have to deal with two kinds of people — the analysts or **"knowledge engineers"** and the users or **"domain experts"**. A domain expert is usually a high quality user, and a knowledge engineer is an analyst who is able to apply particular new analytical techniques for solving problems better than an ordinary user.

After a certain application has been selected, it is the user's requirements that have to be defined by means of structure analysis as to the finished (expert) system. The output must be a structured specification of the system and also a prototype knowledge base.

If the process to which we want to apply our expert system already exists, then we have to document the **current physical system.** This requires unambiguous and nonredundant documentation of the expert's present work.

The aim of the next step is to distill from the current physical documentation both the explicit decision or planning functions the expert is carrying out, and the specific information (not documents) used to perform those functions. The result is exactly what models the **current functional system.** This is an important step, because our real goal is to preserve the functional essence of what the human expert does and embed it in some new technological environment by means of the expert system.

After structuring the current functional system, one can proceed to specifying the new system. This is also a two-stage process whose goal is to get to a specification of the new system, including the technology to be used in running it.

The first step here is on the functional level, and so we do not care how the expert system will be implemented technologically. Now we add any functionality or data which may be needed in the new system. It is requirable to work on functional level as long as possible in order to be as clear as possible about **what** we wish to do before considering **how** we are going to do it. This step of the analysis is termed **new functional system.**

The second step is to specify the physical details of the new system and create documentation of the **new physical system.** Here one has to decide whether the new system will operate interactively or in batch mode, will it involve centralized or distributed processing, is it a multi-user and/or multitask system with significant concurrence problems, are there users with different priorities, is the updating of the knowledge base a continuous, periodical, aperiodical process, and some other factors. Thought must be also given to whether it is appropriate or not to use an existing domain-independent **expert-system shell** if one is available.

Prototyping

From the structured analysis we know that system development is an iterative and highly interactive process of discovering the optimal blend of functional requirements and technological capabilities. This process is carried out by various methods most of which could be assembled under a common notion of modelling. Prototyping is one of modelling methods.

Prototypes as models could be various — rapid prototypes, early prototypes, working prototypes, advanced prototypes, just to give a few examples. Regardless of the qualification of a particular prototype, it carries a connotation of being a model, of being somewhat unfinished, of somehow falling short of the real thing. A prototype is not a finished product, but it can be a very useful approximation to the final goal. Prototyping is actually a prolific modelling tool in specifying nonanalytical problem areas where one has to make certain simplifying assumptions about data structures, relationships, modes of processing. Prototyping is also convenient for building open modular systems where you have to start an early implementation with a "minimum working prototype", and continue to build the whole system around it.

In expert-system technology prototyping is combined with other analysis methods of structured analysis, such as knowledge base development, data flow diagrams (DFD), decision trees, data dictionaries, logic programming languages, etc.

In order to demonstrate some properties of expert system technology, let us consider a simplified, yet real problem.

The Problem

Let us assume that the Estonian Government intends to make a contract with one of the Baltic-Sea Governments for exporting Kunda Cement-Work cement. Let us also suppose that the Kunda port is ready to serve a carrier (ship) about 5000 brutto-register tons, and that, in principle, one of these carriers is available for making about one trip per week. The contract is supposed to be concluded for the duration of one year.

Possessing this preliminary information, the Government must now make a decision about the real **suitability** of this kind of contract.

Before starting to structure and prototype the given problem we can make some general observations, such as

— it is a decision and not a planning task with a nonalternative, may be just a probabilistic answer to a given set of initial data;

— there is no existing system based on this decision, so we have to start knowledge acquisition with the aim of building a new functional system.

Decision Structure

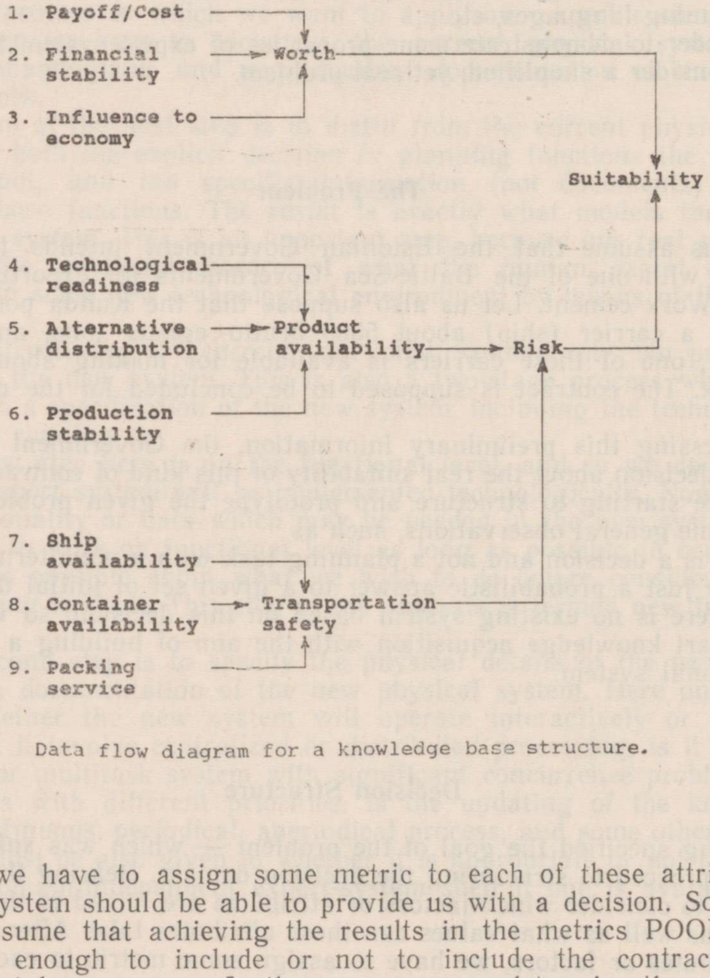
Having specified the goal of the problem — which was **suitability** — we now proceed to structuring the decision domain. Bearing this in mind we have to consider which factors or attributes are linked to this decision making as well as what values can these attributes take. After specifying the attributes or factors, we have to assign some metric to each of those in order to obtain determined values.

While structuring, it is appropriate to start with the final solution, with the SUITABILITY and then go backwards through all the nodes of the hierarchical decision graph. Somewhat simplifying our task we can say that there are two major dimensions for the SUITABILITY to be considered. These are WORTH and RISK.

In commercial ventures, the payoff is frequently measured in terms of profitability, the bottom line. The payoff/cost function is one of the WORTH attributes, and the other two main factors are Financial Stability and Influence on Economy. The first evaluates to what extent the production output is financially guaranteed, whereas the second specifies how does this contract influence the foreign trade.

RISK is a factor which is present in any business contract. In our case it depends mainly on the Product Availability and the Transportation Safety. In other words, what is the risk of having the cement at the port by the right time, in necessary quantities and properly packed.

Refining the decision structure a bit more by specifying some attributes for Product Availability and Transportation Safety, we can now build a hierarchical graph for our decision-making system, which can easily serve also as a diagram of data flow for our knowledge base structure (see below Fig. 1).



Data flow diagram for a knowledge base structure.

Next we have to assign some metric to each of these attributes so that the system should be able to provide us with a decision. So, first of all we assume that achieving the results in the metrics POOR, FAIR, GOOD is enough to include or not to include the contract in the governmental programme. In the same way we determine the values the attributes can take, every one of them. For some factors it is appropriate to specify the values in numerical intervals, inequalities or percentages (Table 1).

By specifying the data flow diagram on p. 80, we have performed many important operations, including decomposing our problem to 5 independent decisions and, respectively, to five mini-domains, listing all the factors and determining the data flow between the mini-domains.

Knowledge Base Decision Structure

Now we have to specify precisely the decision structure for each of the mini-domains of the expert system, which is one of the most important parts of the knowledge base.

The decision structure is presented in Table 1. It should be mentioned that in a decision structure such as this, the number of combinations of values is equal to the product of the number of values each factor can take on. In other words, if we tried to treat each of the factors in combination with the others, we would find that the total number of possible combinations in our decision space would be exactly 3,779,136. So our efforts have to be concentrated on minimizing the number of combinations which must be considered in reaching the desired decision.

In fact, we started this process already with the general structuring of the problem and with decomposing it. Now the number of combinations of values is the sum of the combinations of values for each mini-system, which means in particular that the 3.77 million has been reduced to 136 decision rules, thus reducing the complexity of the decision space and knowledge base by a factor of more than 10,000.

Now we have to continue along that line and refine the decision space structure further by structuring each of the mini-systems. The result of this part of system-analysis is presented in Table 1, and here are a few comments to it.

First of all, the set of factors, their names and meanings as well as their interrelations were found as a result of expert analysis of the general political, economic and monetary situation of Estonia by the end of 1991. Some information about the particular Kunda Cement Works and transportation outlooks was also considered.

In each mini-expert system first the "Decision" is listed along with different values that the data flow can have. Then all of the "Factors" which affect the "Decision", with their allowable values, are shown. For instance

"Product availability" = COMPLICATED/POSSIBLE/EASY

It is easy to see that in several factors and values some uncertainty or probability factor is taken into account not just as an explicit mathematical value but rather as a component of expert evaluation.

Data Dictionary

The structured analysis specification technique has been applied here to the knowledge base system in order to reduce a gigantic knowledge base to a manageable one [2]. As one of the results, a Data Dictionary has to be defined. This defines the information content of every data flow of our problem data-flow diagram on p. 80.

Table 1

Knowledge base decision structures for mini-expert systems of the problem

Decision structure for the expert system "Worth"

Decision:

Worth

Choices:

NEGATIVE
LOW
MODERATE
HIGH

Factors:

Payoff/Cost	Financial stability	Influence to economy
Values:	Values:	Values:
< 1	50%	NEGATIVE
1-1.5	75%	INDIFFERENT
1.5-3	90%	USEFUL
> 3	90%	VERY USEFUL

Decision structure for the expert system "Product Availability"

Decision:

Product availability

Choices:

COMPLICATED
POSSIBLE
EASY

Factors:

Technological readiness	Alternative distribution	Production stability
Values:	Values:	Values:
75%	NO	LOW
90%	DESIRABLE	REASONABLE
100%	NECESSARY	HIGH

Decision Structure for the expert system "Transportation safety"

Decision:

Transportation safety

Choices:

BAD
AVERAGE
GOOD

Factors:

Ship availability	Container availability	Packing service
Values:	Values:	Values:
UNCERTAIN	LOW	LOOSE
FAIR	AVERAGE	OK
CERTAIN	GOOD	CORRECT

Decision structure for the expert system "Risk"

Decision:

Risk

Choices:

LOW
MODERATE
HIGH

Factors:

Production availability	Transportation safety
Values:	Values:
COMPLICATED	BAD
POSSIBLE	AVERAGE
EASY	GOOD

```
Decision structure for the expert system "Suitability"

Decision:
        Suitability
                Choices:
                        POOR
                        FAIR
                        GOOD
Factors:
      Worth                          Risk
        Values:                        Values:
          NEG/LOW                        LOW
          MODERATE                       MODERATE
          HIGH                           HIGH
=========================================================
```

The data for all the mini-expert systems must be structured in a uniform manner, because despite their working independently they all share the same data. For this purpose structured English is used together with some constructs of structured programming.

The Data Dictionary for our example is presented in Table 2. In this Dictionary the information items are split into three different types — **data structures, data elements** and **values** of data elements. So, for instance, Risk Factor is a data structure defined in terms of two data elements: Product Availability and Transportation Safety, which in turn are defined in terms of their values.

An equality sign in the equation means that the item on the left consists of, or is intended to be, whatever is on the right. Information between the quotation marks on the right side is a content comment about the definition, but not the definition itself. The definition is captured into the square brackets and may consist of values of a data element, or data elements, when defining a data structure. The values are written in upper case letters and each data element can take only one value at a time by which it is defined. On the left side of the equation are listed both data structures defined by the data elements (on the right side), and data elements defined by their values (on the right side)

Table2

Data dictionary. It defines the information content of every data flow of our problem data flow diagram in Fig. 1

Alternative distribution	= "Are there other competitive cement distribution projects?" [NO, DESIRABLE, NECESSARY]
Container availability	= "Transportation reliability between the cement mill and the port" [LOW, AVERAGE, GOOD]
Financial stability	= "Is the production output financially guaranteed?" [50%, 75%, 90%, >90%]
Influence on economy	= "How does this contract influence the foreign trade economy policy?" [NEGATIVE, INDIFFERENT, USEFUL, VERY USEFUL]

Packing service = "How is this part of production line backuped?"
[LOOSE, OK, CORRECT]

Payoff/Cost = "What is the estimated ratio of Payoff/Cost in known conditions?"
[<1,1 - 1,5, 1,5-3, >3]

Product availability = "Decision made based on Product Availability Factors"
[COMPLICATED, POSSIBLE, EASY]

Product availability factor = "Factors making the product available for the contract"
[Technological readiness, Alternative distribution, Production stability]

Production stability = "Is the production guaranteed also from other than technological breakdowns (personnel etc.)
[LOW, REASONABLE, HIGH]

Risk = "Decision made based on Product availability and Transportation safety"
[LOW, MODERATE, HIGH]

Risk factor = [Product availability, Transportation safety]

Ship availability = "Availability of the main transformation units"
[UNCERTAIN, FAIR, CERTAIN]

Suitability = "Final decision made based on Worth and Risk"
[POOR, FAIR, GOOD]

Suitability factor = [RISK, WORTH]

Technological readiness = "Cement mill readiness to produce the product at required quality level".
[75%, 90%, 100%]

Transportation safety = "The decision made based on transportation safety factors"
[BAD, AVERAGE, GOOD]

Transportation safety factor = "Factor related to product transportation problems"
[Ship availability, Container availability, Packing service]

Worth = "The decision made based on Worth factors"
[NEGATIVE, LOW, MODERATE, HIGH]

Worth factor = "What makes the worth of the contract"
[Payoff/Cost Financial stability, Influence on economy]

A fraction of the IF-THEN Rule Base

1.0 Rule Base fraction for the mini-ex system **"Determine Worth"**. (64 rules)

```
Rule 1
     If:
        Payoff/Cost              is    > 3
        Financial stability      is    > 90%
        Influence on economy     is    VERY USEFUL
   Then:
        Worth                    is    HIGH

Rule 2
     If:
        Payoff/Cost              is    > = 1
        Financial stability      is    *
        Influence on economy     is    > = USEFUL
   Then:
        Worth                    is    HIGH

Rule 3
     If:
        Payoff/Cost              is    > 3
        Financial Stability      is    > 90%
        Influence on economy     is    NEGATIVE

   Then:
        Worth                    is    LOW

Rule 4
     If:
        Payoff/Cost              is    > = 1.5
        Financial stability      is    > = 75%
        Influence on economy     is    > = INDIFFERENT
   Then:
        Worth                    is    HIGH
```

1.1 Rule Base fraction for the mini-ex system **"Determine Product availability"** (27 rules)

```
Rule 1
     If:
        Technological readiness      is   *
        Alternative distribution     is   NECESSARY
        Production stability         is   *
   Then:
        Product availability         is   COMPLICATED

Rule 2
     If:
        Technological readiness      is   > = 75%
        Alternative distribution     is   NO
        Production stability         is   >=REASONABLE
   Then:
        Product availability         is   EASY

Rule 3
     If:
        Technological readiness      is   > 90%
        Alternative distribution     is   DESIRABLE
        Production stability         is   >=REASONABLE
   Then:
        Product availability         is   POSSIBLE
```

---

* The asterisk instead of a value means that for this particular rule the value of this factor is irrelevant.

1.2. Rule Base fraction for the mini-ex system **"Determine Transportation safety"** (27 rules)

```
Rule 1
      If:
         Ship availability          is  UNCERTAIN
         Container availability     is  < = AVERAGE
         Packing service            is  < = OK

    Then:
         Transportation safety      is  BAD

Rule 2
      If:
         Ship availability          is  CERTAIN
         Container availability     is  > = AVERAGE
         Packing service            is  > = OK
    Then:
         Transportation safety      is  GOOD

Rule 3
      If:
         Ship availability          is  FAIR
         Container availability     is  AVERAGE
         Packing service            is  OK
    Then:
         Transportation safety      is  AVERAGE
```

1.3. Rule Base fraction for the mini-ex system **"Determine Risk"** (9 rules)

```
Rule 1
      If:
         Production availability    is  EASY
         Transportation safety      is  GOOD
    Then:
         Risk                       is  LOW

Rule 2
      If:
         Production Availability    is  COMPLICATED
         Transportation safety      is  < = AVERAGE
    Then:
         Risk                       is  HIGH

Rule 3
      If:
         Production availability    is  > = POSSIBLE
         Transportation safety      is  GOOD
    Then:
         Risk                       is  MODERATE
```

1.4. Rule Base fraction for the mini-ex system **"Determine Suitability"** (9 rules)

```
Rule 1
      If:
         Worth                      is  HIGH
         Risk                       is  < =MODERATE
    Then:
         Suitability                is  GOOD

Rule 2
      If:
         Worth                      is  > =MODERATE
         Risk                       is  LOW
    Then:
         Suitability                is  GOOD
```

```
Rule 3
     If:
          Worth                               is  MODERATE
          Risk                                is  MODERATE
   Then:
          Suitability                         is  FAIR

Rule 4
     If:
          Worth                               is  < =MODERATE
          Risk                                is  HIGH
   Then:
          Suitability                         is  POOR
```

Rule Base

Now we are ready to design the Rule Base for each mini-expert system. It will be accomplished by means of the so-called IF-THEN type of rules. As has been seen, we have reduced the necessary amount of rules already to not more than 136 as a result of the structuring technology. However, while designing the Rule Base we can still apply various methods to simplify it further. Systematic analysis and demonstration of those methods as well as training and teaching the reasoning mechanism is not the subject of this article, therefore only some fractions of the mini-expert systems Rule Bases are presented in Table 3 together with some comments to them.

Although each expert system shell has its own ways of trying to simplify the decision problem, it is very often started with eliminating possibly large classes of solutions up front, thus considerably simplifying the inference process. This strategy has also been followed in the present work.

Comments

1.0. "Determine Worth" Rule Base

The first rule is evident.

Rule 2. "If the contract is strategically USEFUL or VERY USEFUL to the government and the payoff/cost factor is more than one, then regardless of the financial stability (50% anyhow), the "Worth" is HIGH".

Rule 3. "If the influence on the economy is NEGATIVE, then even with the best values of the other factors the "Worth" value cannot be better than LOW".

Rule 4 is evident.

1.1. "Determine Product availability" Rule Base.

Rule 1. "If the alternative distribution is NECESSARY based on some other consideration of the government's economy, then regardless of the best technological readiness and production stability, the value of the product availability cannot be more than COMPLICATED". Note that this rule reduces the number of rules in the mini-ex Rule Base by a factor of 3.

Rules 2 and 3 are self explanatory. Note that using the more or equal, or less and equal notation also reduces the number of rules in the Rule Base and, respectively, the combinations to be calculated. Note also that when applying metrics to the data elements values, exact distinctions are made when the values are a result of some mathematical calculations (for instance payoff/cost factor ⟨1.1 — 1.5, 1.5 — 3,⟩3), whereas when the value is an expert estimation (round-about), the graduation is given with some reasonable interval (for instance technological readiness 75%, 90%, 100%).

1.2. "Determine Transportation Safety" Rule Base.

Rule 1. "If the ship availability is UNCERTAIN, and the two other factors are equal to or less than their average values, then the result cannot be better than BAD".

Rule 2. "If the ship availability is certain, then having in mind that we have to deal with one-per week ship loading, we can assume that even having at least AVERAGE container availability and OK packing service, the value of the transportation safety can be considered GOOD». This reduces 3 rules.

Rule 3. This rule demonstrates just an ordinary "average case" that probably may be reduced in the future training or getting-experienced process of the mini-knowledge base.

1.3. "Determine Risk" Rule Base.

One rule for every possible value is presented requiring no special comments.

1.4. "Determine Stability" Rule Base.

In this structure, the four possible "Worth" values are reduced to three, writing NEGATIVE/LOW into a common lower level case, as there is in fact no need to distinguish between these values according to the reasoning content. This mini-ex system with its 9 rules is also transparent, but we still can reduce the number of rules because of the awareness that the suitability is GOOD only when at least one factor has its extreme best value and the other not the worst.

Conclusions

The above described expert-system shell which was pre-prototyped for our example concerning the Kunda Cement Work export task and fragments of which were presented here, can be easily generalized for rather wide decision classes.

At the beginning of this paper we pointed out that intelligent reasoning systems require a knowledge base, a structured model of the problem domain and an inference engine. We have discussed the first two, whereas the inference engine will hopefully be the subject of the next article.

The main purpose of an expert system is to perform complex decision and planning tasks in the same way as a human expert does. When its knowledge base matures to a truly expert status, one would expect to have the option of replacing the human expert.

REFERENCES

1. *Keller, R.* Expert System Technology. Development and Application. Yourdon Press Computing Series. Prentice-Hall Company, Engelwood Cliffs, New York, 1987.
2. *Keller, R.* The Practice of Structured Analysis. Yourdon Press, New York, 1983.

Received
July 6, 1992

Boris TAMM

FRAGMENTE EKSPERTSÜSTEEMIDE TEHNOLOOGIAST

On kirjeldatud ekspertsüsteemide tehnoloogia põhietappe: teadmushõivet, süsteemide struktuurimist ja prototüpeerimist. Näite baasil on esitatud teadmusbaasi infoloogiline struktuur, otsustuste struktuur, samuti andmesõnastik ja reeglibaas struktuuritud loomulikus keeles.

Борис ТАММ

ФРАГМЕНТЫ ТЕХНОЛОГИИ ЭКСПЕРТНЫХ СИСТЕМ

Рассматриваются основные этапы технологии экспертных систем: сбор знаний, структурирование систем, прототипирование. На основе примера даются инфологическая структура базы знаний, структура решений, а также словарь данных и база правил на структурированном натуральном языке.