Transplant International

Transplant International ISSN 0934-0874

ORIGINAL ARTICLE

Decision tables and rule engines in organ allocation systems for optimal transparency and flexibility

Murk Schaafsma,[1] Wilfred van der Deijl,[1] Jacqueline M. Smits,[1] Axel O. Rahmel,[1] Pieter F. de Vries Robbé[2] and Andries J. Hoitsma[3]

1 Eurotransplant, Leiden, The Netherlands
2 Medical Informatics, Department of Primary and Community Care, Radboud University Nijmegen Medical Centre, Nijmegen, The Netherlands
3 Division of Nephrology, Radboud University Nijmegen Medical Centre, Nijmegen, The Netherlands

Keywords
decision tables, flexibility, organ allocation, rule engine, transparency.

Correspondence
Mr. M. Schaafsma MD, IT Department Development & Information, Eurotransplant, PO box 2304, 2301 CH Leiden, The Netherlands. Tel.: +31 71 5795794; fax: +31 71 5790057; e-mail: m.schaafsma@eurotransplant.org

Conflicts of Interest
The authors have declared no conflicts of interest.

Received: 25 November 2010
Accepted: 1 January 2011
Published online: 3 February 2011

doi:10.1111/j.1432-2277.2011.01221.x

Summary

Organ allocation systems have become complex and difficult to comprehend. We introduced decision tables to specify the rules of allocation systems for different organs. A rule engine with decision tables as input was tested for the Kidney Allocation System (ETKAS). We compared this rule engine with the currently used ETKAS by running 11 000 historical match runs and by running the rule engine in parallel with the ETKAS on our allocation system. Decision tables were easy to implement and successful in verifying correctness, completeness, and consistency. The outcomes of the 11 000 historical matches in the rule engine and the ETKAS were exactly the same. Running the rule engine simultaneously in parallel and in real time with the ETKAS also produced no differences. Specifying organ allocation rules in decision tables is already a great step forward in enhancing the clarity of the systems. Yet, using these tables as rule engine input for matches optimizes the flexibility, simplicity and clarity of the whole process, from specification to the performed matches, and in addition this new method allows well controlled simulations.

Introduction

As in other Organ Exchange Organizations, Eurotransplant (ET) uses organ allocation programs ('the allocation systems') to assign donor organs to the most suitable candidates on the waiting lists. During the last decades the allocation systems evolved from relatively simple systems to a complex set of rules, not only because of medical science but also because the number of countries joining ET has grown. These allocation systems are adapted regularly [1] and are completely renewed after a period of time, as in the Eurotransplant Kidney Allocation System (ETKAS) [2,3] and Liver Allocation System [4].

This paper describes the project of the introduction of decision tables to make organ allocation systems more understandable and the implementation of a rule engine for the ETKAS to reach optimal flexibility, transparency and accountability.

Methods

Design outline

In the process of implementing new rules in an allocation system, the first step was translating the specified recommendation (the specification) in such a way that programmers could translate it into programming code. Both the specification and its translation had to be unambiguous. To maintain the necessary level of transparency and accountability in this complex process, ET started a project with the following points as design criteria:

1. Optimize the clarity of specification in allocation systems:

- Create a standard for the specification of allocation rules that is unambiguous and understandable for users.

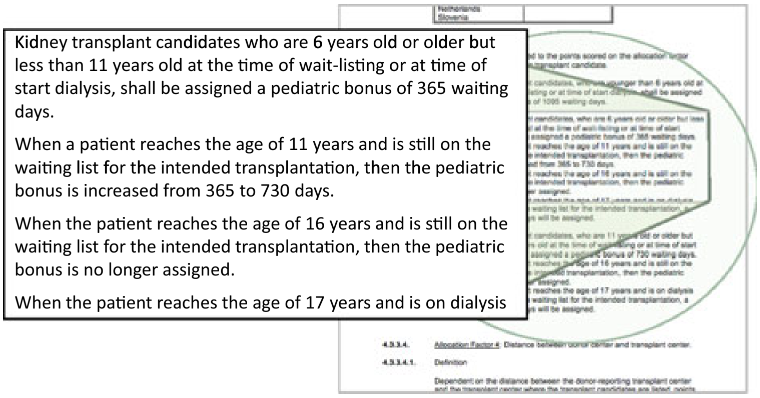


Figure 1 Specification of the calculation of paediatric bonus points in ETKAS, as free text.

2. Create optimal transparency and flexibility in organ allocation systems:

- Standardize the programming.
- Minimize the translation steps of the specified (new) rules into computer language.
- Separate the computer program (the algorithm) and the rules.
- Save the input, the output, and the decisions made in each particular organ match:
  - **(i)** all data used: all patients on the waiting list, donor data and so on;
  - **(ii)** the applying rules at the time of matching;
  - **(iii)** the outcome of the organ match and all results per patient.
- Create the possibility for reproducible testing of adaptations of the rules.
- Optimize the flexibility in changing the rules.

Optimize the clarity of specifications in allocation systems: the use of decision tables

Since the start of organ allocation, the allocation systems are documented and specified in free format text (Fig. 1), usually comprising dozens of pages of description. This kind of documentation is normally readable and understandable text, at least for nephrologists and the medical staff of ET. Where each separate line describing a rule is understandable; several lines with rules can generally be explained, but will be difficult to understand. However, with dozens of pages of rules of specification, it is not always easy to have a clear view of the whole allocation system.

A standard way of documenting (complex) rules in 'systems' in a surveyable way is the use of decision tables, as described by Vanthienen [5]. A decision table is composed of rows and columns, separated in a condition (mostly the upper) part and an action (lower) part. Each condition row contains the condition alternatives. The action rows contain the action to be performed when all conditions in the corresponding column are met (see Fig. 2). These tables are used in many fields and in a great range of medical topics, from immunization [6,7] and coronary surgery [8] to implementation of a therapy decision system for patients with liver metastases [9]. The tables are also widely used in software engineering for documenting and specifying complex decisions in a simple way, which is easy to check for consistency, completeness and correctness [5]. For this reason, ET chose decision tables as a starting point for documenting the allocation systems and evaluated a user friendly program to define and print decision tables in an easy way. This system not only offers an intuitive user interface but also has all the necessary checks built in to ensure that the defined rules are totally correct. In addition, these decision tables are intrinsically the documentation of the allocation systems.

Create optimal transparency and flexibility in organ allocation systems: the use of a rule engine

After writing the specifications of a match algorithm into decision tables, they still have to be translated into

Conditions			
1. Donor type	DCD		DBD
2. Candidate country	Germany or Croatia	other	–
Actions			
1. Candidate selected	–	x	x
2. Candidate not selected	x	–	–

Figure 2 An example of a decision table. The top half represents the conditions and the possible values. The bottom half contains the actions to be performed (marked with a 'x') when all conditions in the column above are met. In this example, in case of Donation after Cardiac Death (DCD) a candidate on the waiting list from Germany or Croatia will not be selected, a candidate from another country will be selected. In case of a Donation after Brain Death (DBD) donor, candidates from all countries will be selected.

computer language (an example how this translation is done in ETKAS is shown in Fig. 3). Programming language is far from natural language and can be very complex, with again a consequent risk of a lack of transparency. To fulfil the design criteria for the computer program, ET had to look for a solution where a better usage of the decision tables could be made.

Standardize the 'programming' and minimize the translation steps of the specified (new) rules into computer language

In this step the current ETKAS was compared with a program that required only a minimum of translation of the specification, i.e. the decision tables, into computer language. Schwarz *et al.* [9] describes a decision support system, in which decision tables are translated into a computer program. Generally, in optimal circumstances, such a computer program should use prespecified decisions directly. Chin and Kotak [10] describes a concept of a computer program, a so called rule engine that can be fed with prespecified rules, which are executed when all the necessary data are entered. But, as described by Kashyap *et al.* [11] the implementation of such a rule engine still involves translating of the decisions into a very abstract logical language comprehensible for mathematicians but not necessarily for the users of organ allocation systems.

For optimal transparency, decision tables have to be used directly in the computer program (the rule engine) without any translations into computer or other abstract language. The decision tables had to be the input for the decisions by the program as so called 'meta data'. We found only two suitable rule engines that fitted into our IT architecture, with a good user interface, to define the rules in an appropriate way with well defined input and output. Standardization of the 'programming' is automatically enforced because there is no programming any more. By doing so, we obtained a common language between medical users and programmers both at the descriptive and the programming level. The rule engine applied by us was Oracle Business Rules [12], a standard product readily available on the market.

This kind of system offered a possible solution, but the challenge remained. Rule engines are used in many fields [10] and in medical applications [11], but only in online applications with just one data set. In our case that would be just one patient with one donor. We found no reports on the use of a rule engine in an application with hundreds or thousands of rules that had to be executed over thousands of data sets such as, in our case, all patients on the waiting list. Notice that the system used Rete algorithms [13,14] for optimizing the performance, which implies that the additional time needed for donor to recipient matching is minimized, which itself is an implicit design criterion.

Separate the computer program from the rules and save the input, output and the decisions of each particular organ match

In using query languages it is a common practice to perform a lot of decisions for selecting during data access, for instance only candidates with a compatible blood group are selected. After this initial selection, the other part of the rules that determine the allocation will be executed. Hence, rules are split in two pieces of programming code. To have full control of the allocation system, it is necessary to separate data access (reading data of patients on the waiting list and saving results of the organ

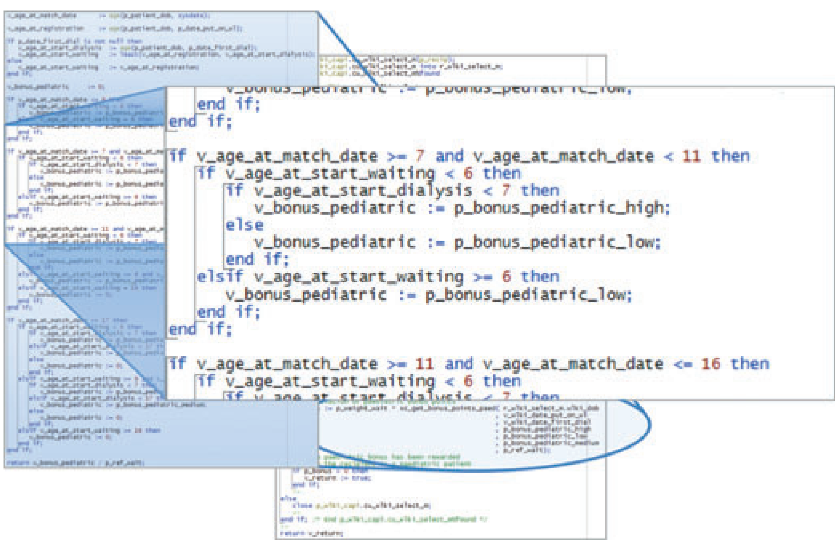


Figure 3 Description of a code in PL/SQL package written for calculating the paediatric points in ETKAS. Points are calculated for patients between 7 and 11 years of age at the date of matching.

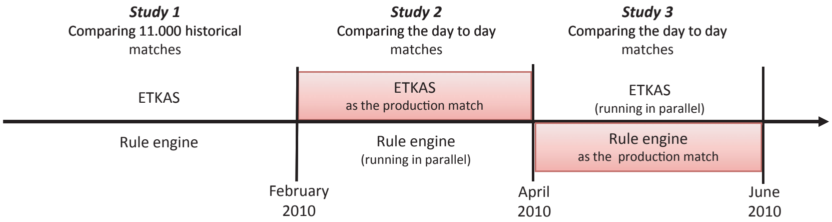


Figure 4 Comparison of ETKAS and the rule engine method in a three step approach.

match) and the execution of all the rules. A three-stage process has to be realized:

1. Read all data from the patients on the waiting list, the available donor and respective organ, and other data.
2. Execute the rules of the allocation system.
3. Save all resulting data and generate the ranking list.

A rule engine by itself cannot access the database directly. Data have to be read from a database and transmitted to the rule engine, together with the applying rules. Subsequently, the rules, as metadata for the rule engine, will be executed on the data and the results will be transferred to the database computer to be used for output and for storage in the database. The data are not only stored in the database, but also archived for accountability. In this way all data used in the match and the results of this particular match are saved for future reference.

Optimize the flexibility in changing the rules

Changing the rules of an allocation system that uses a rule engine with decision tables as metadata, is very easy as only one or more decision tables have to be adapted. No real computer programming has to be done. This possibility of changing the rules on line in a comprehensive and surveyable way offers an enormous flexibility compared with translating it into programming language

Testing a rule engine in implementing the decision tables for kidney allocation system

As first step, a feasibility study with the rule engine that uses decision tables as input for the allocation rules was set up. We assessed the performance of this rule engine by first implementing the kidney allocation rules as described in the decision tables. Both match results, which are a list of ranked transplant candidates, are compared when generated by the two systems. We compared results of the rule engine implementation with the ETKAS (i.e. current production system) by running 11 000 matches from historical donors in both systems. Secondly, a test to determine the quality and the results in real life was undertaken by running the ETKAS and the rule engine 5 months in parallel on our allocation system (shown in Fig. 4). In this phase we used the match list of the ETKAS for the real kidney allocation, where after 3 months of parallel performance, we switched and used the match lists of the rule engine for generating the real match list. In total we tested 18 characteristics of the donor, recipient and matching (described in Table 1), with an average of 783 cases per characteristic.

Table 1. Characteristics tested in the 11 000 kidney match runs.

Donor characteristics
Heart beating/non heart beating
Age
Blood group
Country
Region
Non-ET donor
Donor virology
Extended criteria donor
Euthanasia
Recipient characteristics
Age
High urgency
Transplant Center
Non resident
Outdated screening
Matching characteristics
HLA gradient
Full/incomplete HLA
Number of available antigens

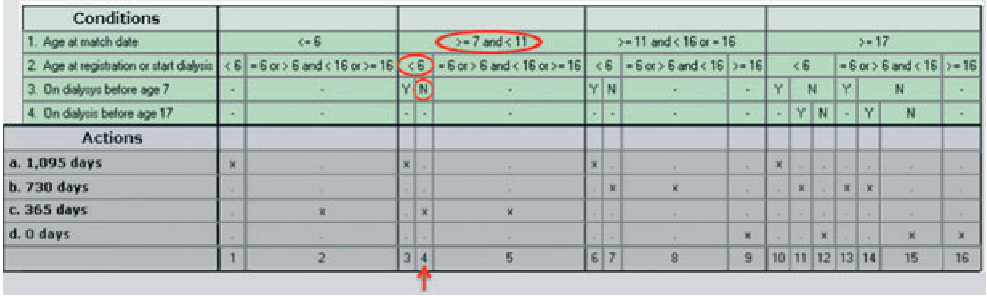
Results

Decision tables

When specifying the ETKAS in decision tables (an example in Fig. 5) we found the free text (Fig. 1) was not as detailed and unambiguous as expected. Details had to be figured out and added because the use of decision tables enforced us to do so. For instance, is the screenings date for determining HLA antibodies, the date it was entered or the sample date? The tables do not accept any ambiguity or noncomplete decisions. After fully accepting the specifications for kidney allocation, the free text documentation of the allocation systems for other organs was translated into decision tables within 2 months. From that time on, all allocation algorithms in ET and all subsequent changes are specified using decision tables.

The rule engine

The difference in the technical implementation between the ETKAS and the rule engine are shown in Fig. 6.



Conditions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1. Age at match date	<= 6		>= 7 and < 11			>= 11 and < 16 or = 16				>= 17						
2. Age at registration or start dialysis	< 6	= 6 or > 6 and < 16 or >= 16	< 6		= 6 or > 6 and < 16 or >= 16	< 6		= 6 or > 6 and < 16	>= 16	< 6			= 6 or > 6 and < 16			>= 16
3. On dialysys before age 7	-	-	Y	N	-	Y	N	-	-	Y	N		Y	N		-
4. On dialysis before age 17	-	-	-	-	-	-	-	-	-	-	Y	N	-	Y	N	-
Actions																
a. 1,095 days	x	.	x	.	.	x	.	.	.	x	.	.	.	.	.	.
b. 730 days	.	.	.	.	.	.	x	x	.	.	x	.	x	x	.	.
c. 365 days	.	x	.	x	x	.	.	.	.	.	.	.	.	.	.	.
d. 0 days	.	.	.	.	.	.	.	.	x	.	.	x	.	.	x	x
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Figure 5 Decision tables used in the ETKAS specification. A patient who is 8 years old at the matching date (older than 7 years of age and younger than 11 years of age), but was registered on the waiting list at 5 years (registered younger than 6 years of age), and whose dialysis started when he was 7 years of age (not on dialysis before age 7), will get a bonus of 365 days waiting time.

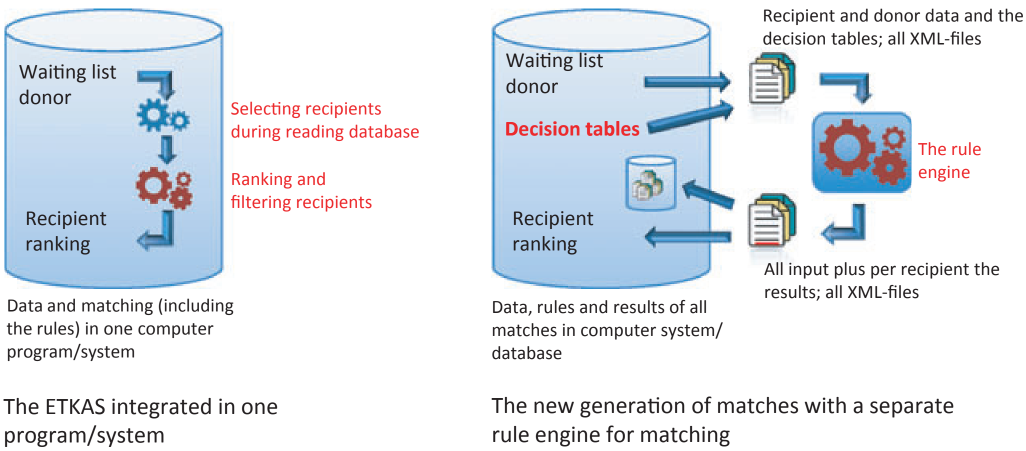


Figure 6 The technical implementation of ETKAS and the rule engine method.

Traditionally the whole allocation system, including data access, is within the same computer and computer program. In a rule engine the execution of the rules is in a separate computer (program). For communication between the database and the rule engine, all data had to be transformed to a standard, readable file format, such as Extensible Markup Language (XML). The rules in the decision table are stored in XML too. For each organ match, the rule engine receives the following data: the available donor and the respective organ data, the candidates on the waiting list, and all rules of the match (the decision tables). During execution of the match, all decisions were added to the candidate data. All input files, with the added decisions per candidate, are stored in the database. These files can be used for future consultation, analysis and, if necessary, for rerunning a particular organ match.

As a result of the ease of use and the flexibility in the rule engine, both, building of the new interface with the database and the implementation of the rules of the decision tables were achieved in 400 h; half the hours spend in 1999 for realizing the (old) ETKAS. As a result of the complexity of the allocation system, the rule engine implementation ended up with a total of 46 decision tables (an example in Fig. 5) with four conditions on average, resulting in approximately 2500 decisions. Programming allocation rules now comes down to defining rules in a decision table; no other programming code has to be written.

The results of the comparison of 11 000 historical matches were identical between ETKAS and the rule engine with regard to the assigned points per patient, and the ranking. Only small, expected differences were found in those cases where detailing in the ETKAS program was lower than required for the decision tables. This implementation of the rule engine showed that the use of a rule engine with decision tables as metadata was reproducible and feasible to use in our organ allocation systems in an easy and flexible way.

As no unexplainable differences were found in the first test, the rule engine was fully accepted for the next phase of testing the rule engine on our allocation system by

Table 2. Design criteria for testing the Eurotransplant Kidney Allocation System (ETKAS) and the rule engine method.

Design criteria	ETKAS	Rule engine
Optimize the clarity of the specification:		
Standard for specification of the allocation system	The rules are defined in decision tables in the specification phase	The rules are defined in decision tables in the specification phase
Optimal transparency and flexibility:		
Standardize the programming	Not enforced, only by a programming quality assurance system	The program, because of the use of decision tables, logically enforces standardization
Minimize the translation steps from specification into computer language	Tables have to be manually translated into programming code	The decision tables are used as input (as the meta data) for the rule engine
Separate computer program and the rules	Not really possible	This is a characteristic of the system
Save the input (including the rules), output, and all decisions of each match	The input can be saved, including the rules in the program. Saving all output and decisions made is not realistic	The system uses standard XML files for input (of all data and rules) and output. As a standard all decisions are part of the output
Possibility for reproducible testing	Test data have to be entered in the database system for each test	The system can be fed with data out of other sources, as spreadsheets and so on
Optimize flexibility in changing the rules	Changes have to be programmed	Only transparent decision tables have to be altered; can even be done on the fly

running in parallel with ETKAS. After the first 3 months of trouble-free use of the rule engine in parallel, the results of both systems were compared. As both ETKAS and the rule engine yielded the same ranking of the candidates, we switched and used the match list generated by the rule engine for allocating the kidneys across the ET countries. For safety reasons and as a final test, we still ran the ETKAS in parallel for a period of 2 months. This time period was a trouble free period too without differences, so we discontinued the ETKAS and ran the kidney allocation system from that time on with the rule engine.

One of the challenges in testing adaptations in any allocation system is to create suitable test cases in the test database, which is an anonymous copy of the actual database. But as the content of the production database is always changing as patients are registered and transplanted, a stable test set is not available. An advantage of the rule engine is that it can run with data from other sources (e.g. Excel sheets), which implies that stable test sets can be obtained. ET is working on a framework for automatic testing of new rules of the allocation systems with a stable and fully known test set. This will be realized within half a year.

Design criteria and the outcomes for both ETKAS and the rule engine system are given in Table 2. It is apparent that the latter system fulfils all our criteria.

Discussion and conclusion

Organ allocation algorithms are complex systems that require multiple decisions and rules. The description of this system is usually only a free text document. In this study, we investigated an alternative system that not only allows unambiguous documentation of the allocation system, but can simultaneously be used as input for the programming of the allocation scheme.

The implementation of decision tables for specifying the rules in all organ allocation systems was shown to be successful. The overall goal was to increase transparency and flexibility. This goal was achieved by standardizing the programming and by minimizing translation steps from specification to actual programming. An inherent characteristic was the separation between the computer program and the rules, which further allowed a clear separation of all input and output data, hence facilitating the reproduction of any historical situation. Finally, the greatest asset of this system is that changes to the allocation algorithm no longer need to be programmed, but can be achieved by a simple adaptation of the decision tables.

Other international organ exchange organizations like UNOS [15], Scandiatransplant and NHSBT (both personal communication) use free text for documenting their allocation systems, whereas in other medical and non-medical fields, decision tables are frequently used, as extensively described by Vanthienen [5].

However, there are only a few reports available that describe the implementation of decision tables in a computer program in such a way that it could be understood by non-IT professionals. Schwarz *et al.* [9] described a system that used decision tables programmed in a standard programming language. This system would not solve our challenge because it would again create a proprietary allocation system. The decision tables would not be used as the metadata for the allocation system. Chin and Kotak [10] described the use of a rule engine in implementing decision making. This rule engine concept was the solution for our challenge; however, rule engines with decision tables as metadata are not widely available yet. For the kidney match, we implemented one of the two suitable products. The implementations fulfilled all design criteria (see Table 2) and the results in testing and in parallel running were the same as with the current production match. Based on this result, the rule engine was used as the ETKAS production match. Rule engines, with decision table as input, are an important improvement in the organ allocation process.

Using decision tables makes specifications comprehensive and readable. Tables can easily be checked for consistency, completeness and correctness. Yet, descriptions of the conditions can be ambiguous. Using decision tables is a great step forward, and the implementation of a rule engine using the decision tables provides optimum clarity. All conditions have to be unambiguous by definition and no free text for explaining some ambiguity in the decision table is possible in the rule engine because the decision tables are the 'programming code' of the rule engine. In case of rule changes, no programming code has to be changed, but only one or more surveyable tables, so optimal flexibility is offered. For input and output, the rule engine needs data files in a standard structure, such as XML. These files can easily be created, read and stored. All performed organ matches are saved along with the rules used, including all data on waitlisted patients, the donors, and the ranking order. The results can be simply retrieved and each performed organ match can be reproduced at anytime. The possibility of creating a real test framework, by using spreadsheets or other standard programs as input for the rule engine, gives an even better quality to the allocation system. The comparison of 11 000 historical matches demonstrates the possibility of testing huge amounts of matches within a surveyable timeframe. The technical solution used for these tests shows that it is feasible to compare different rules in an allocation system in a simulation setting.

In summary, the introduction of the decision tables methodology provided a clear way for specifying organ allocation rules. They were easy to read, easy to change, and user friendly when explaining highly complex allocation systems to the users, which is a requirement for optimal implementation. The whole process from recommendation of a new allocation rule to its technical realization is fully transparent and optimal flexible. All performed organ matches are fully documented and reproducible at any time. Our study showed that the rule engine system fulfilled all design criteria. With its promising simulation potential, the introduction of a rule engine with decision tables is a real great step forward and is for ET the new technical standard in organ allocation systems.

Authorship

MS and WvdD: designed the project. WvdD: performed the technical implementations. MS: wrote the manuscript. WvdD: JMS, AOR, PFdVR. and AJH: reviewed the manuscript.

Funding

The authors have declared no funding.

Acknowledgements

We thank Theo Prins and Wouter Zanen who implemented the decision tables for the organ matches for all organs and the development team of Oracle Business Rules for their cooperation in enhancing the system so it could fulfil all our needs.

References

1. Smits JM, van der Bij W, Rahmel AO, Laufer G, Meiser B. How to maximize the utilization of reported donor lungs in Eurotransplant? Impact of the rescue allocation policy on utilization rates. In: Cecka JM, Terasaki PI, eds. *Clinical Transplants*. Los Angeles: UCLA Immunogenetics Center, 2009: 123.
2. Wujciak T, Opelz G. A proposal for improved cadaver kidney allocation. *Transplantation* 1993; **56**: 1513.
3. De Meester J, Persijn GG, Wujciak T, Opelz G, Vanrenterghem Y. The new Eurotransplant Kidney Allocation System: report one year after implementation. *Transplantation* 1998; **66**: 1154.
4. Jung GE, Encke J, Schmidt J, Rahmel A. Model for end-stage liver disease. New basis of allocation for liver transplantations. *Chirurg* 2008; **79**: 157.
5. Vanthienen J. Ruling the business: about business rules and decision tables. Available at: http://www.econ.kuleuven.be/tew/academic/infosys/members/VTHIENEN/download/Papers/br_dt.pdf (accessed 25 January 2011).
6. Miller PL, Frawley SJ, Sayward FG, Yasnoff WA, Duncan L, Fleming DW. Combining tabular, rule-based, and procedural knowledge in computer-based guidelines for childhood immunization. *Comput Biomed Res* 1997; **30**: 211.
7. Shiffman RN, Greenes RA. Improving clinical guidelines with logic and decision-table techniques: application to

hepatitis immunization recommendations. *Med Decis Making* 1994; **14**: 245.
8. Shiffman RN, Leape LL, Greenes RA. Translation of appropriateness criteria into practice guidelines: application of decision table techniques to the RAND criteria for coronary artery bypass graft. *Proc Annu Symp Comput Appl Med Care* 1993: 248.
9. Schwarz V, Hohenberger P, Köhler CO, Schlag P. Setting up a decision support system with decision tables. *Methods Inf Med* 1989; **28**: 126.
10. Chin AG, Kotak H. Improving debt collection processes using rule-based decision engines: a case study of Capital One. *Int J Inf Manage* 2006; **26**: 81.
11. Kashyap V, Morales A, Hongsermeier T. On implementing clinical decision support: achieving scalability and maintainability by combining business rules and ontologies. *AMIA Annu Symp Proc* 2006: 414.
12. Oracle. Oracle Business Rules. http://www.oracle.com/technetwork/middleware/business-rules/overview/index.html (accessed 25 January 2011).
13. Forgy CL. Rete: a fast algorithm for the many pattern/many object matching problem. *Artif Intell* 1982; **19**: 17.
14. Wikipedia. Rete algoritm. http://en.wikipedia.org/wiki/Rete_algorithm (accessed 25 January 2011).
15. UNOS/OPTN. Allocation of Deceased Kidneys. http://optn.transplant.hrsa.gov/PoliciesandBylaws2/policies/pdfs/policy_7.pdf (accessed 25 January 2011).