International Journal of Computational Intelligence Systems, Vol.2, No. 4 (December, 2009), 332-342

Intelligent Concepts for the Management of Information in Workflow Systems

Georg Peters [a], Roger Tagg [b]

[a] *Department of Computer Science and Mathematics*
University of Applied Sciences - Muenchen
Lothstrasse 34, 80335 Munich, Germany
E-mail: georg.peters@cs.hm.edu

[b] *School of Computer and Information Science*
University of South Australia
GPO Box 2471, Adelaide SA 5001, Australia
E-mail: roger.tagg@unisa.edu.au

Received: 07/01/09
Accepted: 18/05/09

Abstract

Workflow systems are commonly used in industry, commerce and government. They provide computerized support for owners of repetitive, highly standardized business processes, with a means of controlling the execution of instances of those processes according to predefined process templates. However, many real-life business processes are characterized by various forms of unpredictability and uncertainty. For workflow systems to be applicable in these environments therefore, they must incorporate methods of addressing uncertainty, vagueness, variability, exceptional cases and missing information. Methods that have been previously been applied include dynamic instance adaptation, partial completion and case handling - not to mention manual over-riding in the case of exceptions. Intelligent approaches have included stochastic and fuzzy Petri Nets. In this paper, we discuss the further potential of intelligent concepts, in particular rough set theory, for the support of the management of information in workflow systems. Since its introduction in the beginning of the nineteen eighties, rough set theory has gained increasing attention and has established itself as a useful intelligent concept and an important method within soft computing. We show how rough sets can be utilized to set up an early warning system in cases where information is missing in the workflow system. We also show the potential of rough sets to detect excessive or redundant information in a workflow management system's design.

Keywords: Intelligent Concepts, Soft Computing, Degree of Information, Information Management, Workflow Management.

1. Introduction

Today, information has become the fourth production factor supplementing Marx's classic production factors of capital, land and labor. Therefore, information and its effective and efficient management are of strategic importance for any enterprise.

The central challenge of information management is to provide the right information at the right time. To manage this challenge successfully indicators are required that, on the one hand, alert of information shortcomings well in advance and, on the other hand, disclose where information is excessive or redundant.

Presently, the software industry is experiencing a technological shift towards SOA (Service Oriented Architecture)[31]. Basically, the idea of SOA is to design small self-contained software objects that can be quickly and easily combined to support the business processes of a company.

In this context, workflow systems have regained attention as a central layer to glue these software objects together. To design workflow systems several different notations have been suggested and are presently in use (e.g. BPMN (Business Process Modeling Notation)[2;39], eEPC (extended Event-Driven Process Chains)[33] or UML[5]). Unlike these mainly semi-formal modeling languages, Petri Nets[29;20] provide a precise and fully mathematically founded method to design and manage the primary *control flow* perspective of processes of any kind. We will therefore use Petri Nets throughout this paper to illustrate our proposed concepts.

A main challenge in workflow management is to handle the reality of open systems, which is characterized by constant changes as well as vague and uncertain information. To deal with such situations, several concepts to describe uncertainty and vagueness have been suggested. The most established and oldest is probability theory which goes back to the 17th and 18th century when it was introduced by Bernoulli, Laplace, Pascal and others. In 1965 probability theory was joined by fuzzy set theory[41;43]. Recently, Zadeh[42] has been promoting a general theory of uncertainty that provides a holistic framework to describe any kind of uncertainty.

In our paper we utilize intelligent concepts, especially soft computing methods to address this challenge. In particular we show how rough sets, which were introduced by Pawlak[21] some 25 years ago, can be used to manage information in workflow systems more efficiently.

We show how rough sets can be utilized to set up an early warning system when information is missing in the workflow systems. We also show the potential of rough sets to avoid unnecessary information when designing a workflow management system.

The remainder of the paper is organized as follows. In the next section we discuss some principles of flexible workflow management. In Section 3 we introduce to the fundamentals of rough set theory. Section 4 we utilize rough set theory to support the management of information in workflow systems. The paper concludes with a summary in Section 5.

2. Management of Workflows

2.1. The Many Perspectives of Workflow

Jablonski and Bussler[12] point out that workflow management is not in practice limited to the single *perspective* of control flow. Other issues include the:

- *Data Perspective*, namely what information needs to be maintained for each business case, in order to resolve OR-split decisions and termination of repeating loops.
- *Time Perspective*, concerned with the duration of task execution times, delays and deadlines.
- *Resources Perspective*, concerned with the availability of humans, equipment and materials to carry out tasks.

The main concern of our work is with control flow and in particular with the data perspective. However, these perspectives cannot be totally decoupled from the other perspectives, as is discussed below.

2.2. Flexibility in Workflows

The need for flexibility in workflows arises because the assumptions that underly the ideal business process sometimes fail to hold true. Taking the data perspective, the data needed - both to make the workflow routing decisions and to do the tasks themselves - is often missing, conflicting or late. Resource perspective situations can include the unavailability of qualified humans, or the unavailability or shortages of other resources. Time pressures include the pressure of impending deadlines, especially if the remaining tasks are expected to take more time than remains to complete the whole business case; the time pressures may be caused by inability to finish individual tasks within the expected timescale. Finally, it often happens that the business process itself is required to change - often because of changing business environment, competitive pressure, unfavorable customer service feedback etc.

Theoretical approaches to providing flexibility have included Exceptions[14], Dynamic Instance Adaptation[10], Partial Completion[16;24] and Case Handling[37]. Examples of dynamic instance adaptation could include skipping one or more tasks, changing the sequence of tasks, declaring a task sufficiently completed, or relaxing the task dependency rules.

In Case Handling, proposed by van der Aalst et al.[35] the approach is to reduce the number of inter-task dependency rules and let things happen when the data to carry out the tasks, or to resolve any OR conditions, is available. Certain human participants also have special privileges to skip or re-do tasks.

Mixed-initiative Management, as proposed by Rubinstein and Corkill[32] specifically brings in temporal and resource perspectives. As with the suggestion that we propose later in Section 4.3, the style of their solution is to give early warning to the process owner of possible delays or missed deadlines.

Although the common objective here (i.e. efficiently handling process cases) is much the same as that for introducing rough set theory, the emphasis in the two last approaches is more on addressing the "non control flow" perspectives such as data and resource availability for performing tasks and for completing cases in a timely manner. It is an interesting issue as to whether flexibility built in to a workflow model can be good enough for most situations (including emergency handling), or whether the task of adapting instances is really a job best left to human judgment.

2.3. Intelligent Approaches to Workflow Management

In the workflow management literature, there has been some movement towards addressing the problems of handling the variability of individual process instances or business cases, but rough set concepts have not generally been considered to date.

However, there have been several studies based on the application of intelligent concepts such as soft computing methods[19;18], including fuzzy sets[41;43], neural nets[9;4] and genetic algorithms[7]. Although our approach is primarily based on rough sets, some brief comment on these other methods, in particular fuzzy sets, is appropriate at this point.

Zirpins et al.[44] concentrate on fuzzy conditions in workflows, particularly where the condition is compound with simple conditions connected by AND. They try to incorporate measures of probability with the fuzziness. They also use process mining as a means of deriving the workflow models.

Fuzzy business process engineering has been introduced by Huesselmann[11] who suggested a fuzzified eEPC (extended Event-driven Process Chains)[13]. Fuzzy workflows have been proposed by Adam et al.[1] in conjunction with the modeling of workflows using eEPC. An enhanced "fuzzy operator" for the exclusive OR operator is introduced, which uses a min-max inference mechanism.

Wang et al.[38] also apply fuzzy reasoning to a non-Petri model, with a view to matching changed versions of a model. Chan and Zhang[3] have applied fuzzy workflow nets (a variant of Petri nets) in an emergency control application, where dynamic instance adaptation is an essential feature of the application.

3. Fundamentals of Rough Sets

3.1. Basic Properties of Rough Sets

Rough sets were introduced by Pawlak[21;22] in 1982. Since then they have gained increasing importance. Today they can be considered as an important concept within the framework of soft computing. The fundamental idea of rough set theory is that there are two kinds of objects. While some objects are clearly distinguishable from each other some objects are *indiscernible*. The indiscernibility of the objects is normally caused by missing or incomplete information. To deal with such situations, Pawlak suggested the idea of describing a set by two approximations: a lower and an upper approximation of the set. While an object in a lower approximation of a set surely belongs to this set, an object in an upper approximation only *may* belong to the corresponding set. The lower approximation is a subset of the upper approximation of the same set. The area of an upper approximation that is not covered by a lower approximation is often called a boundary area.

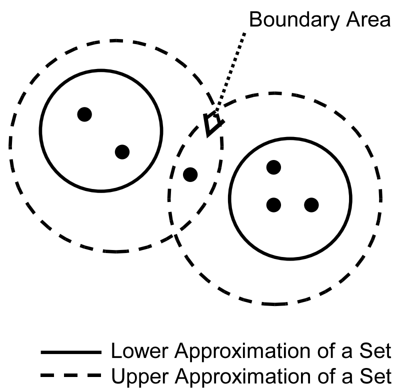

Fig. 1. Lower and Upper Approximations.

Summarized, this leads to three fundamental properties of rough sets:

1. An object can be a member of one lower approximation at most.
2. An object that is a member of the lower approximation of a set is also member of the upper approximation of the same set.
3. An object that does not belong to any lower approximation must be a member of at least two upper approximations.

Many advances have been made in rough set theory over the last 25 years. However, we limit our presentation to these three fundamentals of rough sets. For a basic introduction to rough set theory see[8]. More detailed surveys, especially on its mathematical foundations, can be found for example in Komorowski et al.[15] or Polkowski[30].

It should be noted that the original rough set theory is purely set-based. However, a new interval-based approach has recently been proposed, e.g. Yao et al.[40]. Applications of interval-based rough set theory are in the field of cluster analysis[17;23] and others. However, in this paper we limit our proposals to the set-based version of rough set theory.

3.2. Rough Decision Tables

In the context of our article the application of rough set theory to decision tables is of special importance. Consider the following example[8] dealing with a decision table of eight patients showing different symptoms (Table 1). Four of the patients are well (decision {Flu=no}) while the remaining four patients suffer from flu (decision {Flu=yes}).

Table 1. Patient's Decision Table.

#	Temp.	Headache	Nausea	Decision
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	high	no	no	no
4	high	yes	yes	yes
5	high	yes	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	normal	yes	no	yes

The pair of patients #4 and #5 on the one hand and the pair of patients #6 and #8 on the other hand share the same symptoms {high, yes, yes} and {normal, yes, no} respectively. However, the diagnosis differs, so the decision table does not lead to an unique result. Let us consider patients #4 and #5. While patient #4 suffers from flu patient #5 is well although the patients are indiscernible with respect to their symptoms. Therefore, in the terms of rough set theory, these patients belong to the upper approximations of both the sets {Flu=yes} and {Flu=no}. The same applies to the pair of patients #6 and #8. The diagnoses of the remaining patients do not cause the same problems as described above. Their symptoms lead to a clear diagnosis. While patients #1 and #2 are ill, #3 and #7 are well. So #1 and #2 belong to the lower approximation of the set {Flu=yes} while the patients #3 and #7 are members of the lower approximation of the set {Flu=no}. The implication for the diagnoses of new patients is straightforward; new patients with symptoms equal to #1, #2, #3 and #7 can be treated immediately while patients who have the symptoms {high, yes, yes} and {normal, yes, no} need to have some more detailed physical examination.

3.3. Rough Petri Nets

The potential of rough set theory to Petri Nets has already been investigated by J.F. Peters et al. who suggested rough Petri Nets[27;26;28;25], where the *transitions* function as rough gates. In their example application, Peters et al. applied this idea to sensor and filter models.

4. Managing Information in Workflow Management Systems

4.1. Preliminaries

Currently, several different notations for the design of business processes and workflows are used in practice. Popular notations include BPMN (Business Process Modeling Notation)[2;39]* eEPC (extended Even-Driven-Process Chain)[33] and UML Activity Diagrams[5] †.

A more formal concept with strong mathematical foundations is Petri Net theory[29;20;6] which has been primarily used in technical applications. However, the potential of Petri Nets for business and workflow modeling has been recognized[36], especially since it allows rigid formal analysis of its properties. With some enhancements, Petri Nets are used as the basic modeling notation for some commercial workflow management systems (WfMS), e.g. COSA.

We will therefore use the Petri Net notation to present the intelligent concepts for the management of information in workflow systems.

4.2. Notational Remarks

In Petri Net theory *places* are passive elements, i.e. containers storing tokens. In contrast to that transitions are active in the sense that only transitions can change the state of the net when they "fire"; this means they consume tokens from their input places and produce tokens for their output places. Therefore, transitions are the only elements in a Petri Net that have the capability to make decisions.

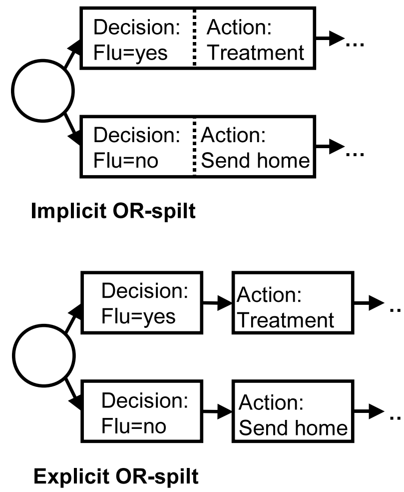

Fig. 2. Implicit and Explicit OR-Splits.

At an OR-split in a workflow the Petri Net diverges and the transitions decide on the further path the case takes. The OR-split can be modeled in an explicit or implicit form (see Figure 2)[36]. In the implicit form the decision and the action are located in the same transition, whereas in the explicit OR-split the decision and the following action are modeled in two separate transitions.

In the following sections we - of course - do not question this concept. However, we graphically mark input places and/or tokens on input places to indicate whether one or more corresponding transitions can fire or not. The decision rules still remain in the only active elements of the Petri Net, the transitions.

4.3. Disclosing Missing Information

4.3.1. Rough Places

Concept of Rough Places. We propose the application of rough sets to OR-split in Petri Nets. If we consider the flu diagnosis example given in the previous section, the rules derived from the decision table can be designed as part of a simple Petri Net

*http://www.bpmn.org
†http://www.uml.org/

representing an OR-split. The patients are symbolized by tokens (see Figure 3 - for simplicity we only show the patients #1 and #2).

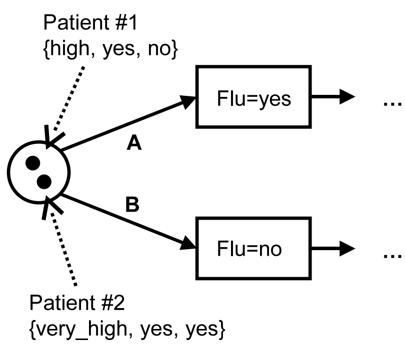

Fig. 3. Diagnosis Decision Tree as Part of a Petri Net.

The firing rules for the transitions and the related assignment of tokens (patients) to the rough approximations are depicted in Table 2.

Table 2. Firing Rules and Rough Approximations.

Patient's Symptoms	Firing	Rough Approximation(s)
{high, yes, no} {very_high, yes, yes}	Flu=yes	Lower Approximation of set {Flu=yes}
{high, no, no} {normal, no, yes}	Flu=no	Lower Approximation of set {Flu=no}
{high, yes, yes} {normal, yes, no}	no firing	Upper Approximations of sets {Flu=yes} and {Flu=no}

Obviously the decision rule at this place is insufficient to deal with all tokens. So some tokens get stuck on the input place of the OR-split. To indicate this we say that the place belongs to the upper approximations of both sets {Flu=yes} and {Flu=yes}. We indicate this by a "dashed circle" place notation as depicted in Figure 4.

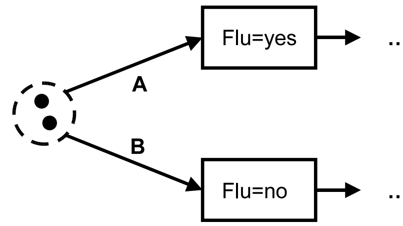

Fig. 4. A Place in Two Upper Approximations.

Potential Applications of Rough Places to WFMS: Incomplete Decision Rule. The appearance of incomplete decision rules (*rough places*) may have two reasons. First, incomplete decision rule, indicated by the appearance of places in upper approximations of both sets {fire=yes} and {fire=no}, can be interpreted as a poorly designed workflow system. The system has to be improved to run properly without any further interruption. Second, a decision rule can intentionally be designed incomplete. Then, for example, the normal cases would pass the decision gate (OR-split) undisturbed. The exceptions would intentionally be "caught" in the upper approximation of a place and presented to the end user for further special treatment.

4.3.2. Rough Tokens

Concept of Rough Tokens. Now consider a patient phoning a General Practitioner (GP). The patients reports that she/he suffers from headache and nausea. However, she/he has not been able to check her/his temperature before phoning the GP. Formally the information provided can be described as: {?, yes, yes}. Since information is missing the GP cannot continue his treatment. In such a case we assign the token to the upper approximation. To graphically distinguish between tokens (patients) belonging to a lower or upper approximations we suggest their representation as shown in Figure 5, namely a "hollow" token for those in the upper approximation.
Potential Applications of Rough Tokens to WFMS: Incomplete Case Information. While the example in 4.3.1 relates to rough places this example is concerned with the information carried by the token

(concept of *rough tokens*). A possible area of application of the proposed method is to provide early warning of potential delays within a workflow system that could be caused by incomplete information in certain business cases. The aim would be to get the workflow system to alert the end user when a choice is waiting on more information. If only the immediate decision is considered, the next transition will be held up. If the complete process including all potential downstream activities is considered, the alert is a warning that further down the track, a transition may be held up.

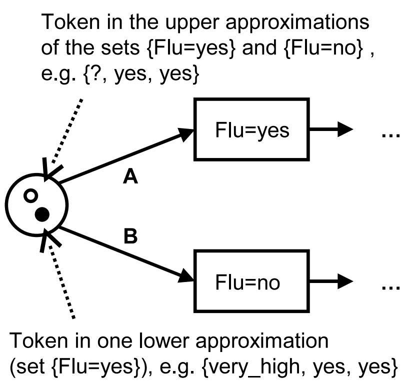

Fig. 5. Tokens in Lower and Upper Approximations.

Ideally, the workflow system should monitor the arrival of the required extra data, so that transitions can be automatically enabled without user intervention. This may well involve facilities to set up software agents that an talk to the applications that manage this data. If, however, it can be seen in advance that certain combinations of case attributes mean that a choice cannot be resolved, the workflow template should probably be altered to allow for a "don't know" branch. The process owner would need to define how long cases can be left in this state, and what should happen to them when time runs out.

4.3.3. Rough Places vs. Rough Tokens

The main difference between the rough places and rough tokens is related to who is responsible when a token gets stuck at a OR-split. In the first case discussed above the token carries all the required information, but the firing rules at the OR-split are insufficient to take a decision. Therefore, the responsibility is at the OR-split. Please note, the definition of a rough place depends on the existence of suitable tokens. Therefore, rough places are not structural properties of Petri Nets.

In the second case the token cannot provide the requested information. Therefore, the token is accountable for its inability to proceed further, so it can be regarded as token in an upper approximation.

4.3.4. Rough Transitions

Concept of Rough Transitions. A token can only proceed when both the token as well as the place the token is assigned to belong to lower approximations. In such a case the decision rule at the OR-split has sufficient information and a transition is enabled to fire.

However, when a token belongs to an upper approximation and/or the place belongs to an upper approximation then the token gets stuck. It is not defined which of the transitions may fire. This leads to the concept of rough transitions.

Let us define the following decision sets {fire=yes} and {fire=no}. Transitions which will surely fire belong to the lower approximation of the set {fire=yes} while transitions that will definitely not fire belong to the lower approximation of the set {fire=no}. The remaining transitions belong to the upper approximations of both sets {fire=yes} and {fire=no}.

As an example, consider the Petri Net given in Figure 6.[‡] Black solid-lined transitions will surely fire. Therefore, they belong to the lower approximation of the set {fire=yes}. The grey transitions surely won't fire[§] consequently they belong to the lower approximation of the set {fire=no}. The sta-

[‡]Since the effects on the capability of making a decision are the same for rough tokens and places (in both cases a token cannot proceed) we will, for simplicity, only display rough places in the example.

[§]The selected path is indicated by a normal arrowhead while the path that is not selected is indicated by a dot in Figure 6.

tus of remaining dashed transition is unclear; they may or may not fire. So they belong to both upper approximations of the sets {fire=yes} and {fire=no}.

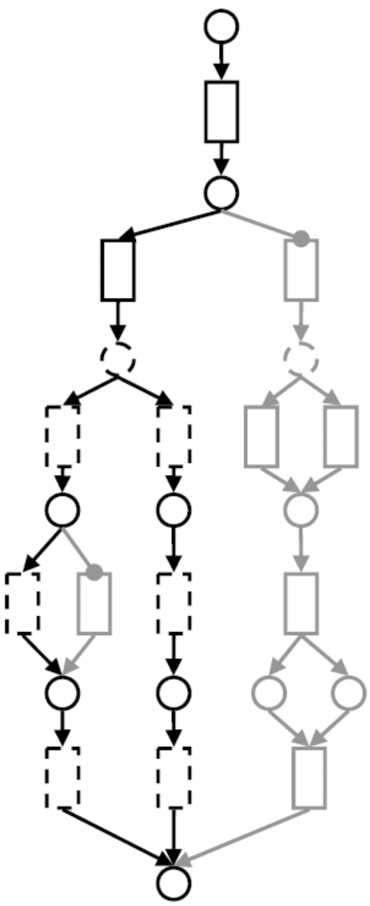

Fig. 6. Rough Transitions.

Please note, that rough transitions, like rough places, depend on the cases (tokens). Therefore, they are not structural properties of Petri Nets.

Potential Applications of Rough Transitions to WfMS: Incomplete Path Information. Resource management is a crucial task in any company. The concept of *rough transitions* supports more efficient management of resources in the following way. As depicted in Figure 6 there are three categories of transition¶: 1) transitions that will be performed surely, 2) transitions that will not be performed, and 3) transitions that may be performed. While in the first case resources have to be allocated to the transitions, in the second case any allocated resources can be released. Uncertainty is reduced to the third case in which it is unclear whether resources are needed to perform the transitions or not.

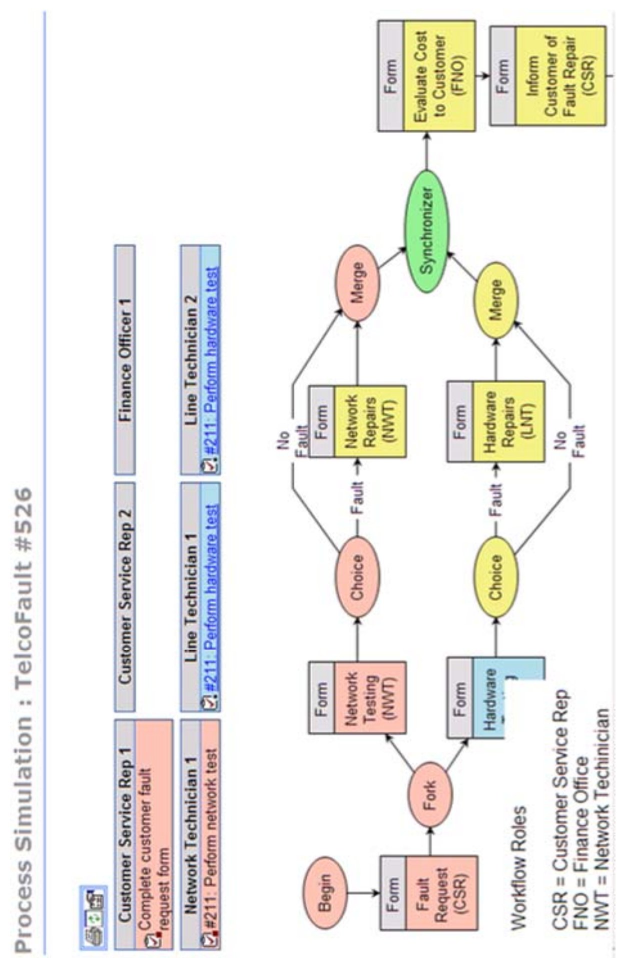

Fig. 7. Color Coding in a Display of a Workflow Process Instance[34].

An example of a user-friendly graphical presentation that could support rough transitions (see Fig-

¶In Petri Nets transitions are only regarded as active entities in the sense that they can change the state of the net. So, generally, business activities can be mapped to transitions and places as well. However, in our context we follow the conventions of leading business process notations, like the EPC[33], where business activities can only be assigned to active entities ("functions").

ure 7) is offered by the WFMS named Chameleon developed by a group based at the University of Queensland, Australia[34]. In the user's view of the process instance, color coding is used to indicate paths that are able to be followed at a given point in time. As this WFMS stands currently, colors can only represent states that follow deterministically from the control flow rules, e.g. in Figure 7, can't be done yet (e.g. yellow), waiting for synchronization (e.g. green), ready to be done (e.g. blue) and already done (e.g. red). There is not yet any consideration of data or resource availability, incomplete rules etc. However, it seems relatively simple to extend this color scheme to include the equivalent of the black, grey and dashed routes in Figure 6.

4.4. Detecting Excessive or Redundant Information

Concept of Reduced Places. To discuss the concept of reduced places we use a variant of the flu diagnosis decision table (see Table 3) as introduced in Section 3.2.

Table 3. Small Patient's Decision Table.

#	Temp.	Headache	Nausea	Decision
1	high	yes	no	yes
3	high	no	no	no
4	high	yes	yes	yes
7	normal	no	yes	no
8	normal	yes	no	yes

Clearly, the decision rule in Table 3 can be reduced since (from the cases shown) the attribute *Nausea* is not necessary for the decision of whether the patient is well or has flu.

In Rough Sets terms the reduced set of attributes $\{Temperature, Headache\}$ is called a *Reduct*. In line with this terminology we call a place that has a decision rule based on a reduct a *Reduced Place*.

Potential Applications of Reduced Places to WFMS: Lean Decision Rules. Reduced places provide a basis to design lean decision rules by applying the concept of reducts. Therefore, it helps to design simple and efficient workflow systems by avoiding overcomplex decision rules.

5. Conclusion

In this paper we have utilized intelligent methods, in particular rough set theory, to manage information in workflow systems. We first presented the concepts of rough places, rough tokens and rough transitions; we then introduced reduced places.

In the first case (rough places, rough tokens and rough transitions), we identified three different occurrences of incomplete information: (i) incomplete decision rules, (ii) incomplete case information and (iii) incomplete path information. In the second case (reduced places), we can detect where there is too much information provided.

In this way, rough sets can help to manage workflow systems efficiently by ensuring that just the right amount of information is provided. The main advantage of using rough sets in workflow applications is that we can draw from a rich theoretical concept to efficiently manage information needs.

References

[1] O. Adam, O. Thomas, and G. Martin. Fuzzy workflows - enhancing workflow management with vagueness. In *Joint International Meeting EURO/INFORMS*, Istanbul, Turkey, July 2003.

[2] P. Briol. *BPMN - The Business Process Modeling Notation Pocket Handbook*. Lulu Press, Morrisville, NC, USA, 2008.

[3] H. Chan and K. Zhang. Application of fuzzy workflow nets in web-based emergency command system. In *Proceed. 2nd IASTED International Conference on Web Technologies, Applications and Services (WTAS 2006)*, pages 67–71, Calgary, Canada, July 2006.

[4] P. de Wilde. *Neural Network Models. Theory and Projects*. Springer Verlag, Berlin, Germany, 2. edition, 1997.

[5] M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Professional, Boston, MA, USA, 2003.

[6] C. Girault and R. Valk. *Petri Nets for Systems Engineering*. Springer Verlag, Berlin, Germany, 20021.

[7] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, Boston, MA, USA, 1989.

[8] J.W. Grzymala-Busse. Introduction to rough set theory and applications. In *Tutorial at KES 2004 - 8th International Conference on Knowledge Based Intelligent Information & Engineering Systems*, Wellington, New Zealand, 2004.

[9] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, N.J., 2. edition, 1998.

[10] S. Horn and S. Jablonski. An approach to dynamic instance adaption in workflow management applications. In *Proceed. Workshop "Towards Adaptive Workflow Systems", ACM 1998 Conference on Computer Supported Cooperative Work*, Seattle, WA, USA, 1998. ACM Press.

[11] C. Huesselmann. *Fuzzy-Geschäftsprozessmanagement*. Josef Eul Verlag, Lohmar, Koeln, Germany, 2003.

[12] S. Jablonski and C. Bussler. *Workflow-Management*. International Thomson Publishing, Bonn, Germany, 1995.

[13] G. Keller, M. Nüttgens, and A. Scheer. Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK). Technical report, Universitaet des Saarlandes, Saarbruecken, Germany, 1992.

[14] M. Klein and C. Dellarocas. A knowledge-based approach to handling exceptions in workflow systems. *Journal of Computer Supported Collaborative Work*, 9:399–412, 2000.

[15] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. *Rough-Fuzzy Hybridization: A New Trend in Decision Making*, chapter Rough Sets: A Tutorial, pages 3–98. Springer-Verlag, Singapore, 1999.

[16] J. Lin and M. Orlowska. Partial completion of activity in business process specification. In *Proceed. IRMA Conference*, pages 186–189, San Diego, CA, USA, 2005.

[17] P. Lingras and C. West. Interval set clustering of web users with rough k-means. *Journal of Intelligent Information Systems*, 23:5–16, 2004.

[18] O. Maimon and L. Rokach, editors. *Soft Computing for Knowledge Discovery and Data Mining*. Springer Verlag, Berlin, Germany, 2007.

[19] S. Mitra and T. Acharya. *Data Mining: Multimedia, Soft Computing, and Bioinformatics*. John Wiley, New York, USA, 2003.

[20] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[21] Z. Pawlak. Rough sets. *International Journal of Computer and Information Science*, 11:341–356, 1982.

[22] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1992.

[23] G. Peters. Some refinements of rough k-means. *Pattern Recognition*, 39:1481–1491, 2006.

[24] G. Peters and R. Tagg. Dimensions of partial completion of activities in workflow management. In *Proceed. Australian Conference on Information Systems*, Sydney, Australia, 2005.

[25] J.F. Peters, S. Ramanna, Z. Suraj, and M. Borkowski. *Rough-Neuro Computing*, chapter Rough Neurons: Petri Net Models and Applications, pages 472–491. Springer-Verlag, Berlin, Germany, 2003.

[26] J.F. Peters, A. Skowron, Z. Suraj, and S. Ramanna. Guarded transitions in rough petri nets. In *Proceedings EUFIT 99 - 7th European Congress on Intelligent Systems & Soft Computing*, pages abstract p. 171, paper on CD, 1999.

[27] J.F. Peters, A. Skowron, Z. Suraj, S. Ramanna, and A. Paryzek. Modeling real-time decision-making systems with rough fuzzy petri nets. In *Proceed. EUFIT98 - 6th European Congress on Intelligent Techniques & Soft Computing*, pages 985–989, 1998.

[28] J.F. Peters, A. Skowron, Z. Suray, and S. Ramanna. Sensor and filter models with rough petri nets. In H.D. Burkhard, L. Czaja, A. Skowron, and P. Starke, editors, *Proceedings of the Workshop on Concurrency, Specification and Programming*, pages 203–211, Humboldt-University, Berlin, Germany, 2000.

[29] C. Petri. Kommunikation mit Automaten. Schriften IIM 2, University of Bonn, Institut für Instrumentelle Mathematik, Bonn, 1962.

[30] L. Polkowski. *Rough Sets*. Physica-Verlag, Heidelberg, Germany, 2003.

[31] M. Rosen, M. Balcer, K. Smith, and B. Lublinsky. *Applied SOA: Service-Oriented Architecture and Design Strategies*. Wiley, New York, USA, 2008.

[32] Z.B. Rubinstein and D.D. Corkill. Mixed-initiative management of dynamic business processes. In *Proceedings 2003 IEEE International Workshop on Soft Computing in Industrial Applications*, pages 39–44, Binghamton, New York, USA, 2003.

[33] A.W. Scheer. *ARIS - Business Process Modeling*. Springer-Verlag, Berlin, Germany, 2000.

[34] N. Shelomanov. Validation and optimisation of workflow processes. Technical report, University of Queensland, St Lucia, Queensland, Australia, 2003.

[35] W. van der Aalst and P. Berens. Beyond workflow management: product-driven case handling. In *Proceed. 2001 International ACM SIGGROUP Conference on Supporting Group Work*, pages 42–51, Boulder, Colorado, USA, 2001. ACM Press.

[36] W. van der Aalst and K. Hee. *Workflow Management - Models, Methods, and Systems*. MIT Press, Cambridge, Massachusetts, USA, 2002.

[37] W. van der Aalst, M. Weske, and D. Grünbauer. Case handling: A new paradigm for business process support. *Data and Knowledge Engineering*, 53(2):129–162, 2005.

[38] X. Wang, R.and Yan, D. Wang, and Q. Zhao. *Global Design to Gain a Competitive Edge*, chapter Flexible Workflow Autonomic Object Intelligence Algorithm Based on Extensible Mamdani Fuzzy Reasoning System, pages 251–260. Springer Verlag, London, UK, 2008.

[39] S. White and D. Miers. *BPMN Modeling and Reference Guide Understanding and Using BPMN*. Future Strategies Inc., Lighthouse Pt, FL, USA, 2008.

[40] Y.Y. Yao, X. Li, T.Y. Lin, and Q. Liu. Representation and classification of rough set models. In *Proceedings Third International Workshop on Rough Sets and Soft Computing*, pages 630–637, San Jose, CA, USA, 1994.

[41] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[42] L. Zadeh. Toward a generalized theory of uncertainty (GTU) an outline. *Information Sciences*, 172:1–40, 2005.

[43] H.J. Zimmermann. *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers, Boston, MA, USA, 4. edition, 2001.

[44] C. Zirpins, K. Schütt, and G. Piccinelli. Flexible workflow description with fuzzy conditions. In *LCS2002 - London Communications Symposium*, London, UK, 2002. University College London.