ARCHIVES of FOUNDRY ENGINEERING
DOI: 10.1515/afe-2016-0029
Published quarterly as the organ of the Foundry Commission of the Polish Academy of Sciences
DE GRUYTER OPEN
ISSN (2299-2944)
Volume 16
Issue 2/2016
75 – 78

Application of ALSV(FD) logic and XTT knowledge representation in the range of ADI properties

K. Regulski [b], D. Wilk-Kołodziejczyk [a,b *], G. Rojek [b], S. Kluska-Nawarecka [a], W.T. Adrian[b]
[a] Foundry Research Institute, Cracow, Poland;
[b] AGH University of Science and Technology, Cracow, Poland;
*Corresponding author. E-mail address: dorota.wilk@iod.krakow.pl

Received 27.01.2016; accepted in revised form 15.03.2016

Abstract

The objective of studies presented in this publication was structuring of research knowledge about the ADI functional properties and changes in these properties due to material treatment. The results obtained were an outcome of research on the selection of a format of knowledge representation that would be useful in further work aiming at the design, application and implementation of an effective system supporting the decisions of a technologist concerning the choice of a suitable material (ADI in this case) and appropriate treatment process (if necessary). ALSV(FD) logic allows easy modelling of knowledge, which should let addressees of the target system carry out knowledge modelling by themselves. The expressiveness of ALSV (FD) logic allows recording the values of attributes from the scope of the modelled domain regarding ADI, which is undoubtedly an advantage in the context of further use of the logic. Yet, although the logic by itself does not allow creating the rules of knowledge, it may form a basis for the XTT format that is rule-based notation. The difficulty in the use of XTT format for knowledge modelling is acceptable, but formalism is not suitable for the discovery of rules, and therefore the knowledge of technologist is required to determine the impact of process parameters on values that are functional properties of ADI. The characteristics of ALSV(FD) logic and XTT formalism, described in this article, cover the most important aspects of a broadly discussed, full evaluation of the applicability of these solutions in the construction of a system supporting the decisions of a technologist.

Keywords: Computer-aided foundry production, Application of information technologies in the field of foundry, ADI, the ALSV (FD) logic of attributes, Rule-based XTT format

1. Introduction

The main objective of this study is to develop a computer system that would support the technologist in his decision on the choice of ADI and its possible variations ([1], [2], [3]) in all those cases where this cast iron is expected to offer certain functional properties (e.g. tensile strength Rm, yield strength Rp0.2). The knowledge collected in such a system must be represented in an appropriate format and connected to an inference engine, which is suitable for the applied knowledge representation. That is why the choice of format and logic used by the format is of crucial importance (this problem was also discussed in [4], [5]). In making this choice it is indispensable, first of all, to consider the power of expression (expressivity) provided by a given format of the knowledge, which should allow to a minimal degree saving the knowledge without losing any information obtained in the conducted experiments or empirical research. Another criterion considered in this selection is the degree of difficulty in knowledge modelling, which directly determines whether

- 10.1515/afe-2016-0029
Downloaded from PubFactory at 07/27/2016 10:49:25AM
via free access

knowledge in the system can be complemented by domain experts (technologists), or whether it is necessary to engage skilled knowledge engineers, the fact that will certainly restrict the use of the system. To choose an appropriate representation, it is also recommended to take into account the tool support and the ability to integrate the existing tools (such as editors or inference engines) with the target system supporting the work of the technologist.

2. ALSV(FD) logic

ALSV(FD) logic (Attributive Logic with Set Values over Finite Domains) is the logic of attributes [6], and as such it allows for knowledge modelling using objects and their attributes. This logic supports attributes that can accept simultaneously more than one value (these are the, so called, attributes of generalization) and works for finite attribute domains [7], [8].

The basic elements of ALSV(FD) logic are the names of attributes and their values. Let A be a finite set of attribute names and D a set of possible values of attributes (their domain). Let A = {A1, A2, ..., An} be a set of attributes whose values define the state of the system under consideration. It is assumed that the set D is the sum of n sets (disjoint or not) D = D1 ∪ D2 ∪ ... ∪ Dn such that Di is the domain of attribute Ai for i = 1, 2, ..., n. It is further assumed that each domain Di is a finite set.

Let Ai be an attribute of the set A and Di – a subdomain associated with this set. Let Vi represent a subset of the set Di and let d ∈ Di be a single element of the domain. The correct formulas of logic along with their semantics are presented in Tables 1 and 2 for simple attributes and generalized attributes, respectively.

Table 1.
The syntax of formulas for simple attributes

Syntax	Semantics
Ai = d	The attribute value is equal to d
Ai ∈ Vi	The attribute value belongs to the set Vi
Ai ≠ d	Shorthand behaves in the same way as Ai ∈ Di \ {d}.
Ai ∉Vi	Shorthand behaves in the same way as Ai ∈ Di \ Vi

Table 2.
The syntax of formulas for generalized attributes

Ai = Vi	The value of attribute Ai is equal to the set Vi
Ai ≠ Vi	The value of attribute Ai differs from Vi in at least one element
Ai ⊆ Vi	The value of attribute Ai is a subset of Vi
Ai ⊇ Vi	The value of attribute Ai is a superset of the set Vi
Ai ~ Vi	The value of attribute Ai has a non-empty intersection with the set Vi
Ai /~ Vi	The value of attribute Ai has an empty intersection with the set Vi

If V is an empty set, i.e. the attribute does not accept any value, it uses the notation Ai = ∅. If the attribute value is vague, it uses the notation Ai = NULL (database convention). If the attribute value is insignificant, it uses the notation: A = _ (Prolog convention). More complex formulas are created using Boolean operators of conjunction (∧) and disjunction (∨) in their classical interpretation.

2.1. Example of application

As an input material, a table with the results of experiments conducted at the Foundry Research Institute in Cracow was used. All the experiments were related with ADI. The ALSV(FD) logic was applied to determine the names of individual attributes, their type, an attribute group and attribute domain.

The first attribute presented here is an attribute named **ChemicalCompositionC**. This attribute expresses the content of carbon in the composition of cast iron. It is a simple type attribute, because it accepts only one value. It belongs to a group **ChemicalCompositionAfter1StageTreatment**, and the attribute domain is numeric. The same group also includes other attributes with the following names (these are also simple attributes with a numeric domain):

- ChemicalCompositionSi,
- ChemicalCompositionMn,
- ChemicalCompositionP,
- ChemicalCompositionS,
- ChemicalCompositionMo,
- ChemicalCompositionNi,
- ChemicalCompositionCu,
- ChemicalCompositionMg,
- ChemicalCompositionV,
- ChemicalCompositionW,
- ChemicalCompositionCr,
- ChemicalCompositionSb.

Attributes relating to the functional properties of ADI include: **Rm, R0,2, A and Z**. These are also simple attributes with a numeric domain. Since functional properties of ADI are altered by the treatment, these four attributes belong to two groups: **Chemical Composition After 1 Stage Treatment** and **Chemical Composition After 2 Stage Treatment**.

To the description of cast iron were also added attributes describing its microstructure. The domain of all those attributes is a symbolic description; they are also simple attributes. **Matrix Type, Pearlite Content, Dispersion** are attributes belonging to the groups: **Microstructure Description After 1 Stage Treatment-Matrix, Microstructure Description After 2 Stage Treatment-Matrix,** By contrast, **Shape, Size and Distribution** belong to the groups: **Microstructure Description After 1 Stage Treatment-Precipitates**, **Microstructure Description After 2 Stage Treatment-Precipitates**.

2.2. Applicability assessment

The difficulty in knowledge modelling is next to none, since the formalism is simple and intuitive. Modelling relies mainly on identification of attribute groups, single attributes and their domains, and in assigning specific attributes to groups. The force of the expression of the ALSV(FD) logic corresponds to the expressiveness of the propositional calculus, but owing to permissible syntactic structures, the record is more compact. ALSV (FD) logic does not allow for the formulation of rules, so its usefulness in the construction of a system supporting decisions seems small. Prospective seems the use of ALSV (FD) logic

- 10.1515/afe-2016-0029
Downloaded from PubFactory at 07/27/2016 10:49:25AM
via free access

formalism as a basis for XTT rule-based format. In studies associated with the definition of the description in ALSV (FD) logic, a QHEd editor may prove to be of some use [9]. Support in the form of this editor is rated as minor, because the editor as such is a tool both inconvenient and non-intuitive in use (it is necessary to perform multiple steps to define simple things).

3. XTT format

XTT format (EXtended Tabular Trees) is a representation of knowledge created for rule-based databases. XTT allows for structured writing of rules using tables for grouping of related rules and relationships between tables, which support the inference flow [10]. Rules in the XTT use attribute-expressive language, XTT2 uses ALSV (FD) logic.

In a set of n attributes A = {A1, A2,. . . , An}, the rule in XTT format takes the following form:

$$(A\,1 \propto 1\,V\,1\,) \wedge (A\,2 \propto 2\,V\,2\,) \wedge \ldots \wedge\ (A\,n \propto n\,V\,n\,) \rightarrow RHS$$

where $\propto i$ is one of the acceptable relational symbols in ALSV (FD) logic and RHS (Right Hand Side) is the right part of the rule containing conclusions. In practice, the conclusions take the form of specific new values assigned to the attributes.

3.1. Example of use

As in the previous section, also now analysis covered the process of ADI manufacture and a table with data derived from the experiments carried out by the Foundry Research Institute in Cracow. Since XTT is not suitable for the discovery of rules (as it is the case with e.g. the decision trees [11]), but only for modelling of the ready relationships, it has been assumed that the process includes the following relationships:

- **Recommended chemical composition** and **Components used** and **Parameters 1. stage of treatment (spheroidization, inoculation)** → affect **Chemical composition after 1. stage of treatment** and **Properties of cast iron after 1. stage of treatment**.
- **Properties of cast iron after 1. stage of treatment** and **Parameters of 2. stage of treatment (austenizing, heat treatment / austempering)** → affect **Properties of cast iron after 2. stage of treatment**.

The above indicated dependencies have been modelled using HQEd editor. The result of this work is shown in Figure 1. From this result it follows that the examined dependencies could be successfully modelled in a manner consistent with their meaning.

The whole XTT diagram (Figure 1) is the structure consisting of tables (including rules) and of the relationships between these tables, which indicate the sequence in which the tables are analyzed by the inference engine. The inference engine analyzes the tables from top to bottom. The analyzed rule can be executed if its conditional part is fulfilled (according to the main paradigm of rule-based knowledge bases). Then, the inference engine advances to the next rule in a table, or moves in accordance with the direction of the relationship to another table.

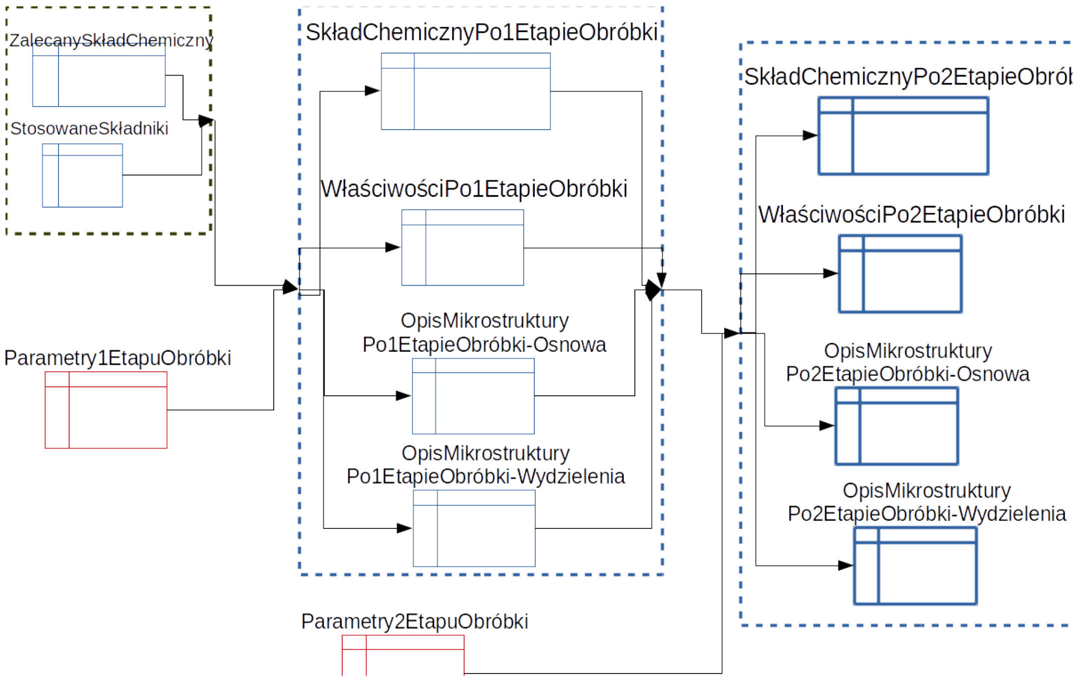

Fig. 1. Sketch of XTT tables with parameters affecting the properties of ADI

- 10.1515/afe-2016-0029
Downloaded from PubFactory at 07/27/2016 10:49:25AM
via free access

3.2. Applicability assessment

XTT format allows for intuitive modelling of knowledge, but in practice the use of a dedicated tool (HQEd) is difficult due to its unstable operation, and therefore the use of a dedicated editor can be described as tedious and time-consuming.

The expressiveness of XTT format follows the rules of production based on propositional calculus. This allows the formulation of rules, their networks and relationships that model the dependencies regarding ADI and its treatment. Once a knowledge base has been created in the form of such rules, it becomes possible to send queries to the database, where the queries can have the following form:

- Which process parameters and chemical composition should be chosen to obtain a particular property?

A unique feature of the XTT format compared with other rule-based knowledge bases is visualization of the modelling process - the rules are grouped into decision tables and presented in graphic form. This way of representation introduces transparency to a model, even in the case of a large number of rules. Additionally, it forces model systematization and structuring, both of which largely contribute to an improvement of its readability. This helps the designer to eliminate from the model many of its anomalies still in the design phase [12], [13].

The rule-based knowledge base can be modelled in the XTT, when we know for sure which process parameters are affected by which values. Formalism can then help prepare a suitable cast iron on the basis of accumulated cases (use case studies), but one should remember that it is not suitable for the discovery of rules. It can only use cases modelled in the database.

The possibilities of process formalization are limited. One can try to represent different phases as attributes necessary for the occurrence of the next phase, but it is not the solution that would well fit the primary application of XTT format.

3. Summary

The difficulty of knowledge modelling with ALSV (FD) logic and XTT format is quite insignificant, which should allow for independent construction of the knowledge base by an expert - technologist, who has readily available expert knowledge in the form of experimental results, and other information resources on the ADI. The expressiveness of XTT format is high - it allows knowledge recording in the form of production rules that operate based on formulas of propositional calculus. A set of rules allows for forward and backward reasoning, which can, among others, also give answer to the question which process parameters and chemical composition will ensure the required performance of ADI. As regards the construction of rules, a support is provided in the form of HQEd editor which, however, is quite troublesome in practical use.

The ALSV(FD) logic of attributes and rules operating in the XTT format constitute a comprehensive tool for modelling of hierarchical rule-based database. Its creation is, however, time-consuming and the obtained benefit unclear. These representations allow only for modelling of the existing rules, since automatic discovering of relationships based on the description of experiments is not possible.

Acknowledgements

Financial support of The National Centre for Research and Development LIDER/028/593/L-4/12/NCBR/2013 is gratefully acknowledged.

References

[1] Kowalski, A., Kluska-Nawarecka S. & Regulski, K. (2013). ADI after austenitising from intercritical temperature. *Archives of Foundry Engineering.* 13(1), 81–88.

[2] Myszka, D. & Bombiński, S. (2014). Preliminary evaluation of the applicability of F, V and AEsignals in diagnosis of ADI machining process. *Archives of Foundry Engineering.* 14(1), 91–96.

[3] Krzyńska, A. (2013). Searching for Better Properties of ADI. *Archives of Foundry Engineering.* 13(1), 91–96.

[4] Mrzygłód, B. & Regulski, K. (2011). Model of knowledge representation about materials in the form of a relational database for CAPCAST system. *Archives of Foundry Engineering.* 11(3), 81–8.

[5] Kluska-Nawarecka S., Mrzygłód B., Durak J. & Regulski K. (2010). Analysis of the applicability of domain ontology in copper alloys processing. *Archives of Foundry Engineering.* 10(2), 69-74.

[6] Ligęza, A. (2006). *Logical Foundations for Rule-Based Systems*. (2nd ed.). Springer-Verlag

[7] Nalepa, G., Ligęza, A. (2008). XTT+ Rule Design Using the ALSV(FD). In the 2nd East European Workshop on Rule-Based Applications (RuleApps 2008) at the 18th European Conference on Artificial Intelligence, 23 July 2008 (pp. 11-15). Patras, Greece.

[8] Nalepa, G., Ligęza, A. (2009). On ALSV Rules Formulation and Inference. In the Twenty-Second International Florida Artificial Intelligence Research Society Conference, 19-21 May 2009. 396 – 401, Sanibel Island, Florida, USA

[9] Nalepa, G., Kaczor, K. (2015). *HQEd.* Retrieved June 5, 2015, from http://ai.ia.agh.edu.pl/wiki/hekate:hqed

[10] Ligęza, A., Nalepa, G. (2007). Knowledge Representation with Granular Attributive Logic for XTT-Based Expert Systems. In the Twentieth International Florida Artificial Intelligence Research Society Conference, 7-9 May 2007 530 – 535, Key West, Florida, USA.

[11] Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning.* 1, 81 – 106.

[12] Ligęza, A., Nalepa, G. (2009). Rules verification and validation. In Giurca, A., Gašević, D., Taveter, K., (Eds.) *Handbook of research on emerging rule-based languages and technologies: open solutions and approaches* 273-301 Hershey, New York, IGI Global.

[13] Ligęza, A., Nalepa, G. (2005). Visual design and on-line verification of tabular rule-based systems with XTT. In Marktplatz Internet: Von e-Learning bis e-Payment: 13. Leipziger Informatik-Tage, LIT 2005, 21-23 September 2005 303-312. Bonn

- 10.1515/afe-2016-0029
Downloaded from PubFactory at 07/27/2016 10:49:25AM
via free access