Fuzzy Extension of the Petri Box Calculus for Modelling of Manufacturing Systems

ALEXANDER LAVROV

Department of Computer Science University of Oldenburg D-26111 Oldenburg GERMANY

Abstract: - This paper outlines a fuzzy extension of the Petri Box Calculus (a process algebra with Petri net semantics) and its application to modelling of automated manufacturing systems. Two aspects are considered as subject to fuzzification: involvement level of argument nets' in the scope of a composition operator, and inter-dependencies of values assigned to tokens. Corresponding to these aspects, the introduced fuzzy-based features enable a flexible representation as well as uncertainty treatment in describing and analysing, first, system configuration and operation sequences, second, material and control flows between components.

Key-Words: - Fuzzy sets, high-level Petri nets, process algebras, automated manufacturing

1 Introduction

Automated manufacturing systems [14] are a standard example of a general class of discrete-event systems (DES) [16]. In modelling, specification, and analysis of DES such means as process algebras (PA's) [11] and Petri nets (PN's) [12] are effectively used as primary means [1, 13] or semantical models for other techniques [7, 10].

Both PN's and PA's possess features which make them superior, in certain respects, to each other (e.g., in PA's: compositionality and association with logics, and in PN's: effective treatment of true concurrency issues, natural graphical description). Aimed at utilizing the advantages of both means, various combined models have been developed, in particular the Petri Box Calculus (PBC)[3]. In the PBC, a PN is associated with each algebraic expression, and manipulating by expressions unambiguously implies corresponding net transformations. The approach proved to be effective both for theoretical and practical applications, and has given rise to a special tool [2].

The current PBC uses crisp descriptions. It ca-

pabilities can further be extended by introducing means for handling uncertainty [17, 18], very frequent in real DES. In the context of DES control [15], an idea of fuzzy synchronisation has originally been proposed in [9]. Now we develop a general fuzzy-based extension of the PBC framework. There are no principal obstacles to applying PBC connectives to fuzzy PN's [6]. Challenging and promising, however, remains a fuzzy treatment of the composition operators themselves (i.e. *how* the nets are composed, rather then which kind of nets is used) and of the way the system dynamics is analysed in this case. These are the issues on which this paper concentrates.

We proceed as follows. Section 2 motivates the problem statement. In section 3, some basic notions are presented, and the idea of the proposed approach is sketched. Then, fuzzy extensions of PBC composition operators are defined (section 4), and a type construction operator, adjustable via a fuzzy parameter set, is introduced (section 5). Explanations are accompanied by application-oriented examples.

2 Problem Description

<u>Fuzzifying the compositions of components</u>. A description of a system configuration or a process plan may require, besides statements like "y always precedes z", also ones of that kind "preferably, y preceeds z", whereby an alternative is not excluded. In fig. 1: the overall direction of the part flow is from C_1 to C_3 . Operation b requires completing a beforehand, whereas the completion of c before b may be desired (e.g. for the reason of joint transportation of the outputs) but not obligatory; as for operation d, it can be performed only after the completion of the others -a, b, and c. The above situation implies the possibility of



Figure 1: Motivating fuzzy ordering of operations

different levels of involvement of arguments into the scope of a composition operator. To enable treatment of the above situation in a PN-based description of a system is the first goal of this paper.

Fuzzifying the dependencies of flow elements.

Even performing a crisp composition of two nets, one may still have to be able to differently handle the *token flow* between the component nets. This may be the case, for example, if the general part flow is always from workcell C_1 to C_2 (i.e. C_1 and C_2 are combined via a crisp sequence operator, fig. 2), however the flow discipline depends on



Figure 2: Independent and mutually dependent flows

parts: some of them have to wait for other parts processed on C_1 because these parts are used together in C_2 (it is a so-called strong discipline; this is the case with parts 2 and 3), while other parts need not wait and can proceed to C_2 just after C_1 , which is a so-called weak discipline; it is valid w.r.t. part 1. A fuzzy extension of the weakness notion w.r.t. composition operators was introduced in [9]. The second goal of the present paper is a formalization of uncertainty treatment of the "weak versus strong"-aspect in the token flow. (Notice the difference w.r.t. the first goal: now we operate on the level of token flow between already "cripsly" combined nets, while the first goal deals with fuzzifying the notion of net compositions).

3 Basic Notions and Approach Outline

3.1 Notation

If X is a (crisp) set then a fuzzy set A in X is a set of ordered pairs $A = \{(x, \mu_A(x)) \mid x \in X\}$, where $\mu_A(x) \in [0, 1]$ is the membership grade of x in A. The sets $A_\alpha = \{x \mid x \in X \land \mu_A(x) \ge \alpha\}$ and $A_{\widetilde{\alpha}} = \{x \mid x \in X \land \mu_A(x) > \alpha\}$ are called the α -cut of X and the strong α -cut of X, respectively. Set $\{x \mid \mu_A(x) > 0\}$ is called the support of A. The set of all fuzzy sets in X is denoted by $\mathcal{F}(X)$. N denotes the set of natural numbers. A multiset over a crisp set X is a function $M : X \to \mathbb{N}$.

3.2 M-nets and Petri Box Calculus

<u>Petri nets.</u> A PN consists of places and transitions, connected by arcs. A system state is represented in a PN by the distribution of tokens over the places; system dynamics is modelled by changing this distribution via transition firings: when fired, a transition consumes certain tokens from its input places and delivers tokens in its output places. For modelling complex and large-scale systems, high-level PN's [8] are used, in which the places, transitions, arcs and tokens are assigned with additional attributes.

<u>*M*-nets.</u> M-nets [4] are a representative of the highlevels PN's. In this paper, a relational form of Mnets representation is used (for a formal definition refer to [5]). The set of all M-nets is denoted by \mathcal{N}_M .

An M-net is a tuple (S, T, ι) , where S and T are the sets of places and transitions, respectively $(S \cap T = \emptyset)$, and ι is the *inscription function* which assigns special attributes to places, transitions, and their pairs (arcs). To each $s \in S$, function ι assigns a status -e, x, or *i* (which means, respectively: input, output, or internal), and a type - $\alpha(s)$, which specifies the tokens (values) allowed on s; a type is a subset of the set VAL of all possible values; v will be used as a generic name for values. For simplicity, we will identify VAL with **N**. With each $t \in T$, ι associates a *label* (an element of the action set Act, and a type $\alpha(t)$, which specifies the *modes* in which t can fire; a generic notation for a mode is m. Each arc is assigned with a multiset of elements of the kind (v, m) or (m, v), depending on the arc's direction. These multisets determine when and how the transitions fire.

Example 3.1 In fig. 3(a), the status of s_1 is e and the type is $\{1,2\}$; the label of t is a and its modes are m_1 and m_2 . The inscription $\iota((s_1,t))$ of the arc between s_1 and t is a multiset containing two occurrences of $(1, m_1)$ and one occurrence of $(2, m_2)$. Place s_1 contains three tokens, symbolically represented by circles with numbers (values) 1 and 2 inside. Transition t can fire in two modes,



Figure 3: Transition firing in an M-net

 m_1 and m_2 . The precondition of firing in mode m_1 is the presence of at least two 1-tokens in place s_1 and one 3-token in place s_2 . Firing of t results in removing exactly two 1-tokens from s_1 and one 3-token from s_2 , and delivering one 4-token into s_3 (fig. 3(b)). Fig. 3(c) shows the outcome of the subsequent (i.e. starting in the net in figure (b)) firing of t in the mode m_2 . $\blacksquare 3.1$

<u>Petri Box Calculus.</u> Due to space limits, we consider an abridged version of the PBC: only operators of parallel (\parallel), sequential (;), choice (\square) composition, as well as a synchronisation operator **sy**¹. The syntax of (simplified) Petri Box expressions is defined by:

$$E ::= \varepsilon \mid E \parallel E \mid E; E \mid E \square E \mid E \mathbf{sy} \ Q,$$

where ε is a basic action (it corresponds to an elementary "place-transition-place" net as one of those four fragments of net N_1 in fig. 4(a)), and $Q \subseteq Act$.

In (a high level net-version of) the PBC, each expression is in a one-to-one correspondence to some M-net. On the semantical level, application of composition operators is performed via combining entry and exit places of the argument nets in a special way, usually a cartesian product-like (fig. 4).



Figure 4: Combining interface places under sequential composition of N_1 and N_2

3.3 Approach outline

Relation of PBC operators to a real system.

PBC operators have their natural counterparts in the primitives by which the material/information/control flows in a manufacturing system are configured (fig. 5): (a) independent operations can be described as parallel (composed via \parallel), (b) if *a* precedes *b*, this corresponds to their sequential composition (;), (c) availability of alternative resources (e.g. if an operation can be performed on either of machines *a* and *b*)

¹For simplicity, we use a synchronisation operator which differs from that in [3]: it synchronises the actions of the same name (rather then those with conjugate names), and assumes an embedded elimination of the participating transitions.

is represented by the choice operator (\Box) , (d) synchronisation **sy** corresponds to joint activity of machines: e.g., loading (denoted by a) of an automated quided vehicle by a robot requires simultaneous participation of both the vehicle and the robot.



Figure 5: Relation between structural characteristics of a system and basic compositions in the Petri Box Calculus

Problem solution outline To achieve the first goal (see section 2), we introduce fuzzy extensions of the above operators. Binary fuzzy compositions operate on fuzzy sets of nets: given a pair of such sets, a composition yields a fuzzy set of nets, whose elements are crisp compositions of elements from the supports of the argument sets. A fuzzy set of nets encompasses possible situations hidden behind uncertain statements. For example, the situation with workcells C_1 and C_2 from fig. 1, can be handled via a fuzzy description of how much c(which is a constituent of C_1) is impacted by the ordering " C_2 after C_1 ". In fig. 6, this is represented by fuzzy set Φ_1 , which contains two nets: the first one (with membership 0.7) assumes independency of c, and the second one (with membership 0.3) induces the completion of both actions,

a and c, in workcell C_1 before starting C_2 .

The second goal, i.e. treatment of different levels of mutual dependency of components in inter-module information and material flows, is achieved via introducing a fuzzy type construction operator. This operator is parameterised by a fuzzy set which contains independent values; these values remain unchanged (as a consequence, so do the corresponding coefficients in arc inscriptions), hence, freely proceed through new combined places; dependent values become tied to the others, and thus must wait for their arrival. Fuzziness of the parameter set results in the possibility of different treatment of the same token and thus in the multifariousness of the represented flow disciplines.

4 Fuzzy Composition Operators

We give a generic definition which encompasses fuzzy versions of each of operators $\|$, ;, and \Box . Let \circ be a generic notation for a composition operator, and $\stackrel{F}{\circ}$ denote a corresponding fuzzy version.

Definition 4.1 (Generic fuzzy composition) Let $\Phi_1, \Phi_2 \in \mathcal{F}(\mathcal{N}_M)$. Then $\Phi_1 \stackrel{F}{\circ} \Phi_2 = \{(N, \mu) \mid \mu = \max\{\min\{\mu_{\Phi_1}(N_1), \mu_{\Phi_2}(N_2)\} \mid N_1 \circ N_2 = N\}.$ $\blacksquare 4.1$

Example 4.2 Figure 6 shows an example of a *PN*-representation of compositions from fig. 1: Φ_1

$$\Phi_{1} = \begin{cases} \begin{pmatrix} e_{\bigcirc} & a_{\bigcirc} & b_{\bigcirc} & \ddots & \ddots & 0.7 \end{pmatrix} \\ e_{\bigcirc} & e_{0} &$$

Figure 6: Outcome of fuzzy compositions: fuzzy sets of nets

represents a fuzzy set of nets which corresponds to fuzzy sequence of the first two centers C_1 and C_2 ; Φ_2 is a set corresponding to C_3 ; Φ_3 represents the outcome of sequential composition of Φ_1 and Φ_2 , which relates to the whole situation in fig. 1.

4.2

In fuzzy synchronisation operator $\mathbf{s}_{\mathbf{y}}^{F}$, fuzziness relates to the description of which actions (represented in PN's by transitions) really have to be performed in a joint manner (i.e. are influenced by the crisp operator \mathbf{sy}):

Definition 4.3 (Fuzzy synchronisation of transitions) Let $A \in \mathcal{F}(Act)$, $\Psi \in \mathcal{F}(\mathcal{N}_M)$. Then $\Psi \text{ sy } A = \{(N, \mu) \mid \mu = \max\{\min\{\alpha, \mu_{\Psi}(N')\} \mid N' \text{ sy } A_{\alpha} = N\}\}.$ $\blacksquare 4.3$

Example 4.4 Let $\Psi = \{(N, 1.0)\},$ where N is as in fig. 7(a), and $A = \{(a, 0.7), (b, 0.2)\}.$ Then $\Psi \operatorname{sy}^{F} A = \{(N_{1}, 0.7), (N_{2}, 0.2)\},$ where $N_{1} = N \operatorname{sy} A_{0.7}$ (fig. 7(b)) and $N_{2} = N \operatorname{sy} A_{0.2}$ (fig. 7(c)) $\blacksquare 4.4$



Figure 7: Illustrating transition synchronisation

5 Fuzzy type construction

<u>Basic type construction</u>. We present a fuzzy extension of the type construction operator \bowtie_{ψ} proposed in [5]. Operator \bowtie_{ψ} discriminates between independent (weak) and dependent (strong) values for the type of a new place resulting from interface places of the to-be-composed nets. Type construction is parameterised by set ψ , which contains the weak values. By varying ψ , one can adjust the level of mutual dependency of values. Fig. 8 illustrates the underlying idea: if place s_3



Figure 8: Origin of a new type

results as a combination of places s_1 and s_2 , then the type of s_3 is defined as follows: the weak part is the intersection of the weak parts of the types of s_1 and s_2 (shadow area; value 1); the strong part is produced by taking all pair-wise combinations of the strong values from the contributing types (e.g. 7 results from 2 in $\alpha(s_1)$ and 5 in $\alpha(s_2)$). (For transparency, in the following we denote each new value by a combination of the contributing values: e.g. 23 in fig. 9(b) is resulted as a combination of 2 from $\alpha(s_1)$ and 3 from $\alpha(s_2)$, fig. 9(a).)

Coefficients related to new values in the annotations of the arcs adjacent to new places are inherited as those related to the corresponding contributing values (fig. 9).



Figure 9: Illustrating arc inscriptions: (a) before and (b) after sequential composition

<u>Fuzzy type construction.</u> A fuzzy extension \bowtie_{D}^{F} of operator \bowtie_{ψ} is based on a fuzzy description (D) of the parameter set. We present two approaches to fuzzy type creation: the first one (definition 5.1) treats new types as fuzzy sets of crisp types, according to the second approach (definition 5.3), a new type itself is a fuzzy set of values. A relation between the two approaches follows from equation 1.

Definition 5.1 (Fuzzy type construction-I)² Let $D \in \mathcal{F}(VAL)$, $\xi_1, \xi_2 \subseteq VAL$. Then $\underset{D}{\overset{F_1}{\bowtie}} (\xi_1, \xi_2) = \{(\xi, \mu) \mid \mu = \max\{\alpha \mid \underset{D_{\alpha}}{\bowtie} (\xi_1, \xi_2) = \xi\}.$

Note that in a fuzzy set obtained according to the above operator, any two different elements (which are crisp types) with a nonzero membership grade have *different* membership grades.

Example 5.2 Let s_1 and s_2 are combined into s_3 . Let $\alpha(s_1) = \{1, 2, 3\}, \ \alpha(s_2) = \{1, 2, 4\},$ and $D = \{(1, 0.9), (2, 0.3)\}.$ Then $D_{0.3} = \{1, 2\},$ $D_{0.9} = \{1\}.$ Further, $\bowtie_{D_{0.3}} (\alpha(s_3), \alpha(s_4)) = \{1, 2, 34\} \ (=\xi') \text{ and}$ $\bowtie_{D_{0.9}} (\alpha(s_3), \alpha(s_4)) = \{1, 22, 24, 32, 34\} \ (=\xi'').$ Finally, $\bowtie_{D}^{F_1} (\alpha(s_3), \alpha(s_4)) = \{(\xi', 0.3), (\xi'', 0.9)\}$ $= \{(\{1, 2, 34\}, 0.3), (\{1, 22, 24, 32, 34\}, 0.9)\}. \blacksquare 5.2$

Definition 5.3 (Fuzzy type construction-II)

$$\begin{split} Let \quad D \in \mathcal{F}(VAL), \quad \xi_1, \xi_2 \subseteq VAL. & Then \\ \stackrel{F_2}{\bowtie} (\xi_1, \xi_2) = \Psi, & where \quad \Psi \in \mathcal{F}(VAL) \quad is \quad obtained \\ via \ injection \ \gamma \ : \ VAL \times VAL \rightarrow \mathcal{F}(VAL) \quad such \ that \\ \forall (v_1, v_2) \in \xi_1 \times \xi_2 : \\ \gamma((v_1, v_2)) = \begin{cases} (v, \max\{\mu_D(v), 1 - \mu_D(v)\}), \\ if \ v_1 = v_2 = v, \\ (v, \min\{(1 - \mu_D(v_1)), (1 - \mu_D(v_2))\}), \\ v \in VAL \setminus (\xi_1 \cup \xi_2), \quad if \ v_1 \neq v_2. \end{cases}$$

Example 5.4 For the same initial data as in example 5.2, we get: $\underset{D}{\overset{F^2}{\bowtie}}(\alpha(s_3), \alpha(s_4)) = \{(1, 0.9),$ The behaviour of a system modeled by a PN N can be characterised by the set $\chi(N)$ of possible firing sequences in N. If N is an outcome of some net composition, with fuzzy type construction applied therein, then the behaviour of N is analysed for individual crisp types which comprise the support of the (fuzzy) outcome of $\bowtie_{D}^{F_1}$, or for fixed crisp types which correspond to α -cuts of the fuzzy types resulted from $\bowtie_{D}^{F_2}$.

Let Ψ be a result of $\overset{F_1}{\boxtimes}$. Let $N_{\alpha}(\overset{F_1}{\boxtimes})$ correspond to N where instead of Ψ a crisp type ξ was used such that $\mu_{\Psi}(\xi) = \min \{ \mu_{\Psi}(\xi') \mid \mu_{\Psi}(\xi') > \alpha \}$. Analogously, $N_{\alpha}(\overset{F_2}{\boxtimes})$ corresponds to N where fuzzy types, resulted from $\overset{F_2}{\boxtimes}$, are substituted with their α -cuts. Let $N(\underset{\psi}{\boxtimes})$ denote the same net N where instead of fuzzy operator $\overset{F}{\boxtimes}$, basic operator \bowtie was used. Let $\psi = \underset{\Sigma}{\boxtimes} \bowtie$. Then

operator $\underset{\psi}{\bowtie}$ was used. Let $\psi = \underset{D_{(1-\alpha)}}{\bowtie}$. Then the following holds:

$$\chi(N(\underset{\psi}{\bowtie})) = \chi(N_{\alpha}(\underset{D}{\overset{F_1}{\bowtie}})) = \chi(N_{\alpha}(\underset{D}{\overset{F_2}{\bowtie}})) \quad (1)$$

In the above equation, the nets underlying the first two components are the same, while the one whose firing sequences are represented by the third component, has different type(s) and incidences.

Conclusions

With the intention to treat uncertainty in modelling and analysing of material/information/control flows in automated manufacturing, we introduced a fuzzy extension of the Petri Box Calculus. The proposed means

²We assume that the new places obtain internal status, which prevents them from further participation in compositions, which allows us to assume that the types of the interface places are crisp, and thus simplify the notation. In the general case, definitions 5.1 and 5.3 can easily be extended to fuzzy type construction from *fuzzy* argument types, in a way analogous to the treatment of fuzzy sets of nets in definitions 4.1, 4.3.

enable one to represent and treat different levels of involvement of system components, represented by Petri nets, into the scope of composition operators which comprise system's specification, and to analyze system's behaviour in the scope of a corresponding fuzzy description. Further research will include, in particular, fuzzy treatment of the refinement in high-level Petri nets, and the development of application-oriented techniques for maintaining tractable fuzzy descriptions in compositional modelling of systems.

Acknowledgements: The author's special thanks goes to Eike Best. The support from Alexander von Humboldt Foundation is also greatly appreciated.

References

- J.C.M.Baeten (ed.), Applications of Process Algebra, Cambridge Univ. Press, 1990
- [2] E. Best, Partial Order Verification with PEP, Proc. POMIV'96 (Partial Order Methods in Verification), G. Holzmann, D. Peled and V. Pratt (eds.). American Mathematical Society, 1997, pp. 305-328.
- [3] E. Best, R. Devillers, J. Hall, The Box Calculus: a New Causal Algebra with Multi-label Communication, Springer-Verlag, LNCS Vol. 609, 1992, pp. 21–69.
- [4] E. Best, H. Fleischhack, W. Fraczak, R. P. Hopkins, H. Klaudel, and E. Pelz, A Class of Composable High Level Petri Nets, Application and Theory of Petri Nets 1995, Springer-Verlag, LNCS Vol. 935, 1995, pp. 103-120.
- [5] E. Best, A. Lavrov, Generalised Composition Operations for High-Level Petri Nets, (Submitted; a longer version is available from http://parsys.informatik.unioldenburg.de/persons/alexander.lavrov)
- [6] J. Cardoso, R. Valette, D. Dubois, Fuzzy Petri Net: An Overview, 13th IFAC World Congress, San Francisco, USA, 30 June-5 July 1996, Vol.J, pp. 443-448.

- [7] H. Fleischhack, B. Grahlmann, A Compositional Petri Net Semantics for SDL. Proc. of ICATPN'98 (International Conference on Application and Theory of Petri Nets), Springer-Verlag, LNCS, Vol. 1420, 1998, pp. 144-164.
- [8] K. Jensen, G. Rozenberg (eds.), Highlevel Petri Nets: Theory and Application, Springer-Verlag, 1991
- [9] A. Lavrov, Models of Fuzziness and Nondeterminism: Joint Use in Discrete Event System Control, Proc. International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA'99), Vienna, Austria, February 1999, pp. 359-365.
- [10] F. Maraninchi, Operational and compositional semantics of synchronous automaton composition. *Proc. CONCUR'92*, Springer-Verlag, LNCS, Vol. 630.
- [11] R. Milner, Communication and Concurrency, Prentice Hall, 1989
- [12] C. A. Petri, Kommunikation mit Automaten, Schriften des Institutes für instrumentelle Mathematik, Bonn, 1962
- [13] J.-M. Proth, X. Xie, Petri Nets. A Tool for Design and Management of Manufacturing Systems, John Wiley & Sons, 1996
- [14] J. A. Rehg, Computer-Integrated Manufacturing. Englewood Cliffs, NJ: Prentice-Hall, 1994
- [15] R. S. Sreenivas, On the Existence of Supervisory Policies that Enforce Liveness in Discrete-Event Dynamic Systems Modeled by Controlled Petri Nets. *IEEE Transactions* on Automatic Control, Vol. 42, 1997, No. 7, pp. 928–945.
- [16] A. Tornambe, Discrete Event System Theory, World Scientific Publishing Co Pte Ltd, 1995
- [17] L. Zadeh, Fuzzy Sets, Inform. and Control, Vol. 8, 1965, pp. 338–353.
- [18] H.-J. Zimmermann, Fuzzy Set Theory and Its Applications, Boston: Kluwer Acad., 1992