A Physical-Effect-Considered Formalized Modeling for Pipelined Manufacturing Machining System

Yu Min-cong[a] [*] Qi De-yu[b]

School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, Guangdong, China

[a]yu.mingcong@mail.scut.edu.cn, [b]qideyu@gmail.com

Keywords: Computer integrated manufacture, IACS; System modeling method, Physical effect relation

Abstract: New stage of global industry is coming with the new requirements to the IACS (Industry Automation Control System), e.g. flexible manufacturing, small batch and customized production. So far, most of IACS modeling methodologies are concerning on the description to specified domain of computing subsystem, e.g. the architecture of computing hardware and software. In this paper, a physical-effect-considered formal modeling method for pipelined manufacturing machine is addressed. The formalism can be used to not only drive valid control and cooperation of different modules, but also validate the solution of sequence of operation planning (SOP) in the design phase.

Introduction

With the manufacturing industry marching towards 4.0, flexible manufacture, small batch manufacture and customized manufacture for pipelined manufacturing are put forward, which means higher requirements for the IACS (Industrial Automation Control System). Most of the existing methods[1] [2] [3] are for computing subsystem, such as AADL and SysML, but seldom focus on the mechanical and electrical subsystem domain, especially the model describing the behavior in a domain-specific way. CAD applied in the mechanical and electrical subsystem field also offers a dynamics simulation environment, but it only focuses on the simulation and analysis of any specific physical parameters, rather than the abstract description about the relationship of the physical effect implemented by the behaviors to the materials, which is too difficult for designers. The MPSG[4][5], MFSM[6] and P-SOP[7] with related works on SOP in [8] [9], illustrating the description of structure and behavior in the mechanical and electrical subsystem domain, which are based on the FSM, did not introduce the concept of physical effects rules or principles between the parts and the materials, namely the basic rules for domain experts to describe the cooperative transmission and processing behaviors of/to materials in the modules, which guarantees the materials can be completely and safety produced. In the absence of introduction of the concept, domain experts have to repeatedly be working on the same description or design, or the complement are done by the domain experts of the computing subsystem for lazy implementation, as a result that a large number of repeated similar modules of codes are scattered in different modules. In all, it lacks either the domain model description, or the calculation model mapping. Pipelined manufacturing machine refers to the manufacturing machinery equipment that combines a series of processing technology with pipeline transmission device on the materials input. Based on the

observation and study of such machines, this paper addresses a formalized modeling method based on the physical effect relations between the material and machine, which introduces the domain specific rules of physical effect and safety manufacturing, and becomes the formalism foundation of the description of module actions and the algorithm of device driving and operation validation.

Formalized representation of physical-effect-considered model

A) Material

M, the collection of materials in the scope of physical effect from machine. In this paper, the material is limited to the one of discrete and rigidity, which occupy a fixed and confirmed space during effect.

B) Parts

$C = \{C_{car} \cup C_{ttc}\}$ The parts make the physical effect to the material directly, where, C_{car} is for the set of parts of carrier type for transfer; C_{ttc} is for the set of parts of tentacle type for process.

C) Physical effect status

$s \in S = \{S_{mov} \cup S_{proc}\}$, where, S_{mov} is for the set of motion status of the material under the physical effect, $S_{mov} = \{pla, cap, mov\}$ is for the three status of placing, dominating and moving, which placing means the part is placing and supporting the material without enforcement, dominating means the part is dominating the material with enforcement and the moving means the part is driving the material. S_{proc} is for the status of interacting, which means the part is interacting properties with material, e.g. the part is reading or writing property from or to the material, e.g., measuring the length, reading the electric data or printing the layout content.

D) Part-material physical effect relation. $E = <C, M, S>$,

$\forall i, j, <c_i, m_i, s_i>, <c_j, m_j, s_j> \in E, \neg \exists c_i = c_j$. In this paper, the focus is

$E_m = \{<c_i, s_i> / \forall i, <c_i, m_i, s_i> \in E\}$, namely, all the physical effect relations of m from the parts related. $\forall <c_i, s_i> \in E_m, s \in S_{mov} \Leftrightarrow c \in C_{car}, s \in S_{proc} \Leftrightarrow c \in C_{ttc}$.

E) Position map

Let t be the unit time, $path(t)$ be the function on t of physical position of material when it is being transferring or processing in the machine, $f_{E_m}(t)$ be the function of E_m on t. Therefore, the discrete set of position of m is

$$P_m(i) = \begin{cases} path(0) & i = 0 \\ path(t_i) & f_{E_m}(t_{i-1}) \neq f_{E_m}(t_i), i > 0 \end{cases}, i = 0, 1, 2, \ldots$$

Then, the position map of machine is defined as a directed graph $MAP_M = <P_M, Dir>$, where, $P_M = \bigcup_m^M P_m$ is for the set of positions of machine, and

$Dir =< P_{start}, P_{end} >= \bigcup_{m}^{M} \{ < P_m(i), P_m(i+1) >| i = 0,1,2...\}$ is the set of directed edges between position which is the material transfer direction, where, P_{start} stands for the start position, and P_{end} for the end position.

F) Carrier location. Define the relation of carrier parts and its position located $L =< C_{car}, P_M >$, where the original status is empty set. Define three operations on relation L, $SetLocation : L \times C_{car} \times P_M \to L$, $GetLocation : L \times C_{car} \to P_M$ and $GetCarriers : L \times P_M \to \{c\} \subset C_{car}$, which is for setting, getting the position of carrier located and getting the carrier set of position specified respectively.

G) The reachable function of carrier parts
$\forall c \in C_{car}, p \in P_m$, the function is defined as

$$Reachable_m(c,p) = \begin{cases} True & \exists t, p = path(t), E_m = f_{E_m}(t), < c, plc >\in E_m \\ False & Otherwise \end{cases}.$$

Namely, the carrier placing the material m can be located in the position p in some time if the function returns *True*, otherwise *False*.

H) The placeable function of carrier parts
$\forall c \in C_{car}, p \in P_m$, the function is defined as

$$Placeable_m(c,p) = \begin{cases} True & \neg \exists < c', s' >\in E_m, c' \in C_{car}, c' \neq c \\ False & Otherwise \end{cases}, < c, pla >\in E_m, E_m = f_{E_m}(t), p = path(t)$$

namely, the carrier part can place the material m independently on position p if the function return *True*, otherwise *False*.

I) The unite relation of parts and reference points
Let the relation $U =< C, Ref >$, where $Ref =< C_{car} \cup P_M \cup M >$, for the carrier or tentacle parts C are relative fixed to the points Ref, e.g. unite relation, where, the Ref can be carriers, positions or material. Since relation U is transitive, define the judgment function of the existence of unite relation between parts and reference points as $Uniteable : C \times Ref \to True | False$, let function $u(c,r,G) = True | False : \forall c \in C, r \in Ref, G \subset U$, then

$$u(c,r,G) = \begin{cases} False, & G = \varnothing \\ True, & < c,r >\in G \\ True, & \exists i \in C_{car}, < c,i >, < i,r >\in G \\ True, & \exists < x,y >\in G, u(c,x,G- < x,y >) = True \wedge u(y,r,G- < x,y >) = True \\ False, & Otherwise \end{cases}$$

then define $Uniteable(c,r) = u(c,r,U)$. Specially, $\forall c \in C, r = m \in M, < c,r >\in U \Leftrightarrow < c,cap >\in E_m$.

J) Material transfer/process physical effect operation
Define two properties of material m, e.g. $p_{cur} \in P_M$, which is for the current position of material, and $E_m \in E$ with initialization as $p_{cur} = nil$ and $E_m = \varnothing$. There are ten operation functions defined below with different parameters but the same return of the validation of operation. If the operation is valid, the post condition will be executed and if it is succeed the two properties of material or the

relation U mentioned above are updated. Before the introduction of the operations, define the subtype of tentacle parts as $C_{ttc} = \{C_t^P, C_t^A, C_t^B\}$, which we call type P, type A and type B. Type P parts perform the process to the material only if the material is captured and fixed in a certain position. Type A parts perform the process when the material is moving by a carrier. Type B parts process material without the pause of moving of material.

1. *OP Allocator.* Create material information and set its location to the entrance of machine.

params	$p \in P_M, c \in C_{car}, s \in \{plc, cap\}$
pre-cond	$\forall E_{m'} \subset E, m \neq m', \neg \exists <c, s'> \in E_{m'} \wedge E_m = \varnothing$
post-cond	$E_m = \{<c, s>\}, p_{cur} = p$

2. *OP Capture.* Effect from carrier to m of placing turns to capture.

params	$c \in C_{car}$
pre-cond	$\forall <c, s> \in E_m, s \neq mov$
post-cond	$<c, cap> \in E_m, <c, m> \in U$

3. *OP StartMove.* Effect from carrier to m of capture turns to moving.

params	$c \in C_{car}, break \in \{True, False\}$
pre-cond	$<c, cap> \in E_m, Uniteable(c, p_{cur}) = False$, $break = True \wedge |GetCarriers(p_{cur})| \neq 0 \vee break = False \wedge |GetCarriers(p_{cur})| = 0$
post-cond	$<c, mov> \in E_m, \forall c \neq c', c' \in C_{car}, Uniteable(c', c) = False, <c', s> \notin E_m$

4. *OP StopMove.* Effect from carrier to m of capture turns to capture.

params	$c \in C_{car}, join \in \{True, False\}, p_{end} \in P_M$
pre-cond	$<c, mov> \in E_m, Reachable(c, p_{end}) = True$, $Placeable(c, p_{end}) = True \vee Placeable(c, p_{end}) = False \wedge |GetCarriers(p_{cur})| \neq 0$, $join = True \wedge |GetCarriers(p_{end})| \neq 0 \vee join = False \wedge |GetCarriers(p_{end})| = 0$
post-cond	$<c, cap> \in E_m, p_{cur} = p_{end}, \forall c' \in C_{car}, GetLocation(c') = p_{end}, <c', plc> \in E_m$

5. *OP StartProcess.* Effect from tentacle to m of interact starts.

params	$c \in C_{ttc}$
pre-cond	$Uniteable(c, m) = True, c \in \{P_t^P, P_t^A\}$
post-cond	$<c, int> \in E_m$

6. *OP StopProcess.* Effect from tentacle to m of interact stops.

params	$c \in C_{ttc}$
pre-cond	$<c, int> \in E_m$
post-cond	$<c, int> \notin E_m$

7. *OP Pin.* Create the unite relation of parts and reference points.

params	$c \in C, r \in \{C_{car} \cup P_M\}$
pre-cond	$((c \in C_{car} \vee c \in C_t^P \vee c \in C_t^B) \wedge r \in P_M) \vee (c \in C_t^A \wedge r \in C_{car})$
post-cond	$<c, r> \in U$

8. *OP Unpin.* Dispose the unite relation of parts and reference points.

params	$c \in C, r \in \{C_{car} \cup P_M\}$
pre-cond	$<c,r> \in U$
post-cond	$<c,r> \notin U$

9. *OP Release.* Effect from carrier to *m* of capture turns to placing.

params	$c \in C_{car}$
pre-cond	$\forall <c,s> \in E_m, s \neq mov$
post-cond	$<c,plc> \in E_m, <c,m> \notin U$

10. *OP Dispose.* Dispose material information and properties.

params	*nil*
pre-cond	$\forall <c,s> \in E_m, c \notin C_{ttc}, s = plc$
post-cond	$E_m = \varnothing, p_{cur} = nil$

Device driving and operation validation

According to the above mentioned the material transfer/process physical effect operation, a flow for device driving and operation validation based on it is given below, shown as Fig 1. Generally, the pipelined machine is divided into different module based on geographical space and the flow of material, and the modules are called components. Components have their own action to perform, e.g. material input, material output, material routing and processing, which is descript by the sequence of physical effect operations. Driven by the request of the material from its own loading, the components submit their actions to the synchronizer, which later combines and generates the available schedule of SOP to validation and executor (V&E). V&E can run in both or either modes of validation and execution, in which the core algorithm is validating the SOP by its precondition via E_m and p_{cur}. Once they are validated, they are translated into the respective implementation to drive the device, and later update the properties of m by their postconditions if succeed.

Conclusions

According to the research and formalism above, the conclusions to the contribution in this paper are given. Firstly, the module division based on geographical space and the flow of material, which is more proper for domain experts to describe their design of control policy and mechanism, is available since the physical effect between the material and machine part is considered and formalized. Secondly, the formalized operation to the material is the essential interface of device driver programming for software engineers, which is effectively taking the part of programming to the behavior cooperation of machine apart from device driver implementation. Finally, both verification and device driving are available in the same architecture of the core algorithm.
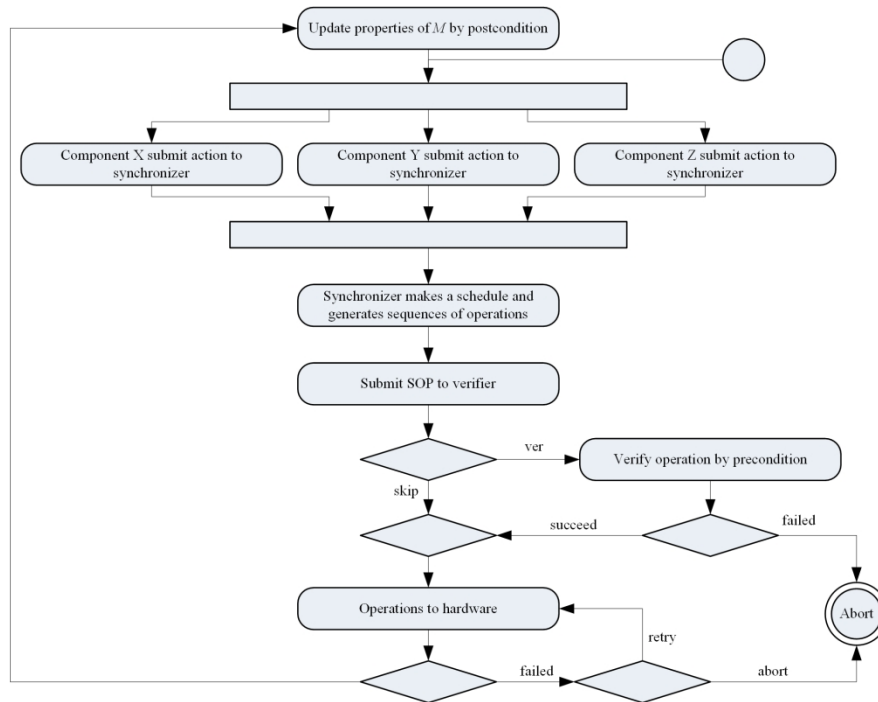
Fig 1. The flow graph of device driving and operation validation

References

[1] Yang, Z.-B., Pi, L., Hu, K., Gu, Z.-H. and Ma, D.-F. (2010) AADL: An architecture design and analysis language for complex embedded real-time systems. Ruan Jian Xue Bao/Journal of Software. 21(5) 899-915.

[2] Jamro, M. and Trybus, B. (2013) An Approach to SysML Modeling of IEC 61131-3 Control Software. 2013 18th International Conference on Methods and Models in Automation and Robotics (Mmar) 217-222.

[3] Angelov, C., Sierszecki, K., Marian, N. and Ma, J. P. (2006) A formal component framework for distributed embedded systems. Component-Based Software Engineering, Proceedings. 4063 206-221.

[4] Smith, J. S., Joshi, S. B. and Qiu, R. G. (2003) Message-based Part State Graphs (MPSG): a formal model for shop-floor control implementation. International Journal of Production Research. 41(8) 1739-1764.

[5] Thapa, D., Jaeil, P., Gi-Nam, W. and Dongmin, S. (2006) Timed-MPSG: A Formal Model for Real-Time Shop Floor Controller. Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on.

[6] Park, E., Tilbury, D. M. and Khargonekar, P. P. (1999) Modular logic controllers for machining systems: formal representation and performance analysis using Petri nets. Robotics and Automation, IEEE Transactions on. 15(6) 1046-1061.

[7] Svensson, B. and Danielsson, F. (2014) P-SOP – A multi-agent based control approach for flexible and robust manufacturing. Robotics and Computer-Integrated Manufacturing(0).

[8] Lennartson, B., Bengtsson, K., Chengyin, Y., Andersson, K., Fabian, M., Falkman, P. and Akesson, K. (2010) Sequence Planning for Integrated Product, Process and Automation Design. Automation Science and Engineering, IEEE Transactions on. 7(4) 791-802.

[9] Miremadi, S., Akesson, K., Fabian, M., Vahidi, A. and Lennartson, B. (2008) Solving two supervisory control benchmark problems using Supremica. Wodes' 08: Proceedings of the 9th International Workshop on Discrete Event Systems 131-136.