Diagnostic statement simulations of process systems

S A Nikishchenkov
Samara State University of Transport, 2, Svoboda Ave., Samara, 443066, Russia
E-mail: nikishchenkovs@mail.ru

Abstract. The article examines the application of diagnostic schemes for technological processes. On the basis of the Bernstein-Russell-Narinyani theorem, we obtain assumptions about correct and defective technological processes.

1. Introduction
Simulations, methods, and solutions in theoretical parallel programming may be successfully applied not only for solving tasks of process systems (PS) productivity increase, but also for PS monitoring and diagnostics [1-3]. This is due to the fact that parallel simulations are associated with a detailed study of a processes information basis with the PS monitoring, in turn, being aimed at prevention of material (in general case, of any other) resources loss. Some of the monitoring methods used (network schedules, methods of interlock and prevention of conflict and dangerous situations, regulation and ranking of operations, etc.) are applied before the parallel computers have appeared, but the formal justification and development are obtained from the parallel programming grounds.

In general, the process description includes: the final product type and characteristics; transient state at the process stages; operations composition, characteristics, and procedure; equipment and resources (material, information, time resources, etc.). The operation characteristics include the type, pertinence to the workplace, type of conversion (function), input and output data and tangible media, time parameters, relations (connections) with other operations, control parameters (procedure, order as well as start and completion attributes).

We will define the flow of operations as a theoretical and multiplex representation of the aggregate of operations, belonging to the process, distributed in the set of PS subsystems and examined in a monitored period of time:
- $A_{t}$ is a subset of operations of the t-th process, being performed at the moment t;
- $A_{i,t}$ is the i-th conversion operation;
- $A_{i,t}(P_j)$ is the operation for logical condition check;
- $nA_i$ and $outA_i$ are input and output operation arguments;
- $fA_i$ and $tA_i$ are conversion and time of operation;
- $A^s_{t}$ is a subset of operations being started in the moment t;
- $A^r_{t}$ is a subset of operations being “ready” for run under conditions of all arguments and resources availability.

2. Results and Discussion
In such arrangement, the process operational monitoring reduces to local or centralized monitoring of relations between events in the workflow. It seems efficient to use parallel process simulations in the
form of statement schemes, as well as their paralleled forms (parallel graph schemes of algorithms, trigger functions, information and logic schemes, counter networks) [4-9] using the following technology:

- selection of diagnostic model, i.e. the aggregate of the process scheme and the list of defects in the formal relations form;
- development of defects-detection algorithms;
- development of a monitoring system (selection of method for diagnostic indicators readout, organizational and technical implementation, and diagnostic-processor programming);
- implementation of measures to ensure the monitoring effectiveness (identification of the monitoring zone, modification of the diagnostic simulation, etc.).

Parallel (independent, asynchronous) run of statements (operations of single of different processes) is defined by the Bernstein-Russell-Narinyani theorem on the absence of competition dependencies [6]:

\[(inA_i \cap outA_j) \cup (inA_j \cap outA_i) \cup (outA_i \cap outA_j) = 0.\]

Unlike computing systems, allowing simultaneous reading of certain data by different statements without memory damage, this kind of competition dependence should also be considered in the PS, and this results with the supplemented theorem:

\[(inA_i \cap inA_j) \cup (inA_i \cap outA_j) \cup (inA_j \cap outA_i) \cup (outA_i \cap outA_j) = 0.\]  \(1\)

It is promising to examine the theorem (1) from the diagnostician point of view, i.e. as a correct functioning condition (CFC), the transition from which to the defect attributes (DA) is carried out with the exception-to-the-rule principle. In this case the evaluation of consequences of non-fulfillment of conditions for input and output variables independency for two scheme statements running in parallel is performed. For the process, expression (1) means asynchronous unordered (including simultaneous) use of elements of information and material basis with a possible conflict situation, recognition and avoidance of which is one of the monitoring tasks.

See Table 1 for possible conditions for the intersection of n-tuples of variables for each of four elements in formula (1) as well as the corresponding defects content. For example, the option with the number 1000 describes the case when only input n-tuples of statements intersect.

A typical production analogy is that operations at different work places use the same hardware (or other element of information and material basis), and it is being intercepted by the operation that starts to be performed first. Thus, the second operation is delayed (is not performed), etc., which results in the correct process distortion.

Non-fulfillment of conditions according to (1) results in combinations of defects such as interception, substitution (of one intermediate result with another), and the argument absence. The last defect is due to the fact that the type of dependence described by two middle elements of expression (1) requires the mandatory sequence of one statement after another, since the result of the first one is the argument of the second one.

To describe the entire process or its fragment, simulations in the form of schemes, describing the statements with their connections (Figure 1) are used:

The statement scheme represents in a symbol form the aggregate of operations \(A_1-A_{15}\) (two of these, \(A_8\) and \(A_9\), are checking the value of \(P_1\) and \(P_2\) logical conditions), n-tuples of input and output variable statements, for example: \(inA_6 = a, b\); \(outA_6 = f\), transfer of variables between statements and statements run logic under their cyclic repetition and alternative execution. For example, statements \(A_1-A_5\) are corresponding to the input acceptance of parts and components \(a, b, c, d, e\); \(A_6, A_9, A_{12}, A_{14}\) are corresponding to the cyclic processing of the \(f\) assembly with check of condition \(P_2\) and transfer of \(f\) to the final assembly of the unit \(A_{13}\); \(A_8\) checks \(e\) for \(P_1\) condition and provides an alternative transmission to \(A_{13}\); \(A_{15}\) operation corresponds to the assembly technical assurance.

The statement scheme is the basis for algorithmization—the structural and temporal distribution of operations in the PS resources. A parallel graph scheme of the algorithm in Fig. 1, 2 specifies the sequence of assembly operations performance at two workplaces.
Table 1. Interpretation of defects in parallel process operations using Bernstein-Russell-Narinyani theorem

|       | 0 0 0 0 | 1 0 0 0 |
|-------|---------|---------|
|       | No defects | Interception |
|       |          |          |

|       | 0 0 0 1 | 1 0 0 1 |
|-------|---------|---------|
|       | Substitution | Interception with Substitution |
|       |          |          |

|       | 0 0 1 0 | 1 0 1 0 |
|-------|---------|---------|
|       | Absence | Interception and Absence |
|       |         |          |

|       | 0 0 1 1 | 1 0 1 1 |
|-------|---------|---------|
|       | Substitution and Absence | Interception, Absence and Substitution |
|       |         |          |

|       | 0 1 0 0 | 1 1 0 0 |
|-------|---------|---------|
|       | Absence | Interception and Absence |
|       |         |          |

|       | 0 1 0 1 | 1 1 0 1 |
|-------|---------|---------|
|       | Substitution and Absence | Interception, Absence and Substitution |
|       |         |          |

|       | 0 1 1 0 | 1 1 1 0 |
|-------|---------|---------|
|       | Mutual Absence | Interception and mutual Absence |
|       |         |          |

|       | 0 1 1 1 | 1 1 1 1 |
|-------|---------|---------|
|       | Substitution and mutual Absence | Interception, mutual Absence and Substitution |
|       |         |          |

Trigger functions [4] are the ultimate predicate over multiple statements and logical conditions in the scheme (including, in general, additional ones, which consider the process history) and specifies the maximally parallel asynchronous process (for the class of so-called “free” schemes), to ensure statements start as far as their input arguments are ready. For example, for the $A_{13}$ statement the start condition will be as follows: $A_{13}$ will be started in case $A_9$ is executed with $P_2 = 1$ value and $A_{10}$ is executed, and either $A_8$ with a value of $P_1 = 1$ or $A_{11}$ is executed:

$$C_{13} = A_9(P_2^1) & A_{10} & (A_8(P_1^1) \oplus A_{11}) \rightarrow A_{13}. \quad (2)$$

The trigger functions-based control assumes the availability of an appropriate PS architecture and control mechanisms. For the purpose of diagnostics of reconfigurable PS, use of monitoring rather than control properties of trigger functions is effective [3, 9]. In this case, the correctness of the statement start is checked by the condition of the trigger function truth:
4

3

and otherwise it is recognized as defective. The process data and logic scheme (PDLS) as shown on
the Fig. 1, 3, describes such control with two types of logic used: AND and Exclusive OR. On the
Figure symbols AND are accepted by default, Exclusive OR is denoted with ⊕.

3. Conclusion
As shown in [10], for PDLS the tier-parallel form (TPF) may be set, where (1) is fulfilled for all
statements within each tier.

Here are some affirmations on correct and defect processes:

1. Any process may be represented in the statement scheme form, and for each statement a trigger
and monitoring function (2) and (3) can be set, and for the correct process in any moment of time and
for each operation the CFC is fulfilled:

K_t,i := A_{i,t} \leftarrow A_{0, P_{2}^t} \& A_{10} \& (A_{0, P_{1}^t} \oplus A_{t,i}).

2. For each process there is a maximally parallel asynchronous pipelined workflow, corresponding
to trigger functions-based control:

∀ t ∀ A_{i,t} \subseteq A_{i}' \subseteq A_i^t \subseteq A_i^t,

3. The rearrangement of operations independent according to (1), transition from one process
paralleling degree to another, or dynamic paralleling of operations, results in its reconfiguration, i.e. to
the structural and temporal redistribution of operations.

4. For parallel process with arbitrary control method:
CFC: \( \forall t \exists A_t^i \subset A_t^j \land \left( \left| A_t^i \right| \geq 1 \right) = 1 \); DA: \( \exists t \exists A_t^i \not\subset A_t^j \lor \left( \left| A_t^i \right| = 0 \right) = 1 \).

In such case, all defects such as “false start,” “no start,” and “statement replacement” representing a complete system of algorithmic defects are detected.

1. PDLS is a diagnostic invariant of reconfigurable PS [3].
2. For the development of defects detection algorithms, the interpretation of diagnostic simulations with counter networks is feasible [9].
3. For inspectable processes, the CFC-based monitoring method [10] is applicable.

In assessment of possibilities of monitoring tools implementation using parallel simulations, PS inspectability, engineering tasks complexity, and the degree of monitoring-methods possible automation should be considered.

References

[1] Czichos H 2013 Handbook of Technical Diagnostics. Fundamentals and Application to Structures and Systems (Berlin: Springer-Verlag Berlin Heidelberg)
[2] Ju F, Lee H K, Osarogiagbon R U, Yu X, Faris N, Li J 2015 Computer modeling of lung cancer diagnosis-to-treatment process. Translational Lung Cancer Research 4(4) 404-414
[3] Nikishchenkov S A 2002 Functional diagnostic of reconfigurable information and control systems at macro level Information technologies at the railway transport (Infotrans-2002). The 7-th International Scientific and Practical Conference (Saint Petersburg, Russia)
[4] Kotov E V 1978 Introduction to the theory of program schemes (Novosibirsk: Nauka)
[5] Glushkov M, Letichevskii A A 1969 Theory of algorithms and discrete processors Advances Inform. Systems Sci 1
[6] Padua D 2011 Encyclopedia of Parallel Computing (New York: Springer)
[7] Krystof R A 2003 Principles of Constraint Programming (Cambridge: Cambridge University Press)
[8] Kasyanov V N, Evstigneev V A 2000 Graph Theory for Programmers. Algorithms for Processing Trees (Boston : Kluwer Academic)
[9] Balakin V N, Barashenkov V V, Kazak V V, Nikishchenkov S A 1988 Device for control units monitoring A.s. No. 1365086
[10] Balakin V N, Barashenkov VV, Kazak V V, Nikishchenkov S A 1989 Device for control of parallel computing system A.s. No. 1451697