RESEARCH PAPER

Multiset Analysis of Consequences of Natural Disasters Impacts on Large-Scale Industrial Systems

Igor Sheremet

Financial University under the Government of Russian Federation, Moscow, Russian Federation, RU
sheremet@rfbr.ru

Paper is dedicated to the new approach to distributed industrial systems (IS) sustainability/vulnerability assessment. This approach is based on the unitary multiset grammars (UMG) as a flexible and convenient tool designed specially for large systems analysis and optimization. UMG description of IS technological base as well as multiset representation of order completed by the IS, its resource base and impact on the IS are presented. Criterion for recognition of IS sustainability to the impact is formulated. UMG extension for natural disasters impacts (NDI) representation is introduced, and criterion for recognition of IS sustainability to the NDI is also presented. The solution of the reverse problem, concerning part of the order, which may be completed by the affected IS, is described. Implementation issues are considered.

1 Introduction

Modern industrial systems (no matter, global, regional, national, transnational etc.) are complicated, distributed, strongly interconnected networks of local facilities, producing, transporting and utilizing various products and resources. That's why, – first of all, of the mentioned strong interconnectivity and associated with it multiple chain effects, – these systems are often vulnerable to natural disasters in such a way, that the one only facility destruction may cause consequences far beyond area or place this facility is located. So one of the most actual, important and at the same time difficult problems of the modern system analysis is development of the widely available mathematical toolkits, models and computer technologies, which would be able to provide the decision makers from the government and corporate stuffs with comprehensive and precise prognostic information about such consequences – first of all, would be system, affected by natural disaster impact (NDI), vulnerable or sustainable to it.

Industrial system (IS) in general case contains technological base (various industrial complexes and devices) producing (manufacturing) various objects (cars, computers, buildings etc.) and utilizing various resources (materials, microchips etc.), necessary for this process (**Fig. 1a**). Natural disaster destroys some segments of the technological and resource bases, that's why amounts of objects produced are decreasing (**Fig. 1b**).

By this we may formulate criterion for IS robustness/vulnerability assessment (**Fig. 2**).

If amounts of objects, produced by IS before and after natural disaster impact, are equal, then IS is sustainable to the NDI; otherwise it is vulnerable to the impact. In the last case there is also one more important "reverse" problem: what part of the work may be executed by vulnerable IS, which facilities are affected by NDI?

Let us consider formulated problems in more details.

As was said higher, IS include industrial (manufacturing) devices (complexes). Each such device may be represented as a "black box" B with m inputs and one output (**Fig. 3**). Every i-th input is marked by a_i – name of object (item, resource) type, – and n_i – amount (volume, quantity) of this object. So n_1 objects (of type) $a_1, \ldots, n_m$ objects (of type) a_m are required for one object (of type) a manufacturing.
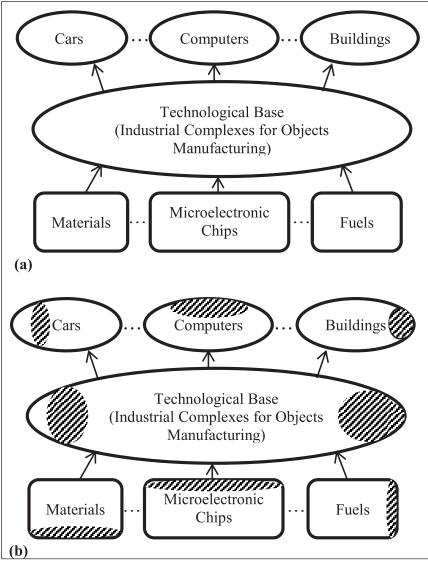
Figure 1: Industrial system: **(a)** in normal state; **(b)** after impact.
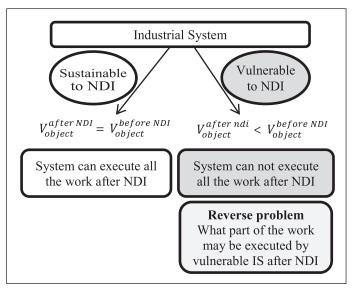

Figure 2: Possible consequences of natural disaster impact on industrial system.
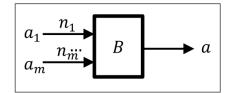
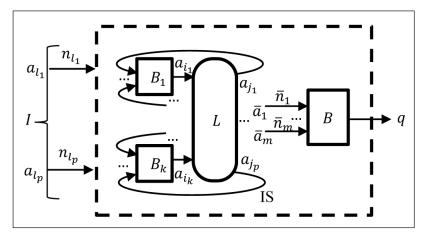**Figure 3:** "Black box" representation of manufacturing device.

Figure 4: "Black box" representation of industrial system and order to be completed.

IS as a whole may be represented by k such "black boxes" $B_1, \ldots, B_k$ interconnected by the "logistical ring" L, providing transport of objects, manufactured by devices, from their outputs to inputs of another devices, thus forming integrated manufacturing process (**Fig. 4**).

This process, however, is driven by orders, which sources are external systems or persons, consuming objects, produced by IS. Each order q in general case defines types and quantities of objects, which would be manufactured ($\bar{n}_1$ objects of type $\bar{a}_1, \ldots, \bar{n}_l$ objects of type $\bar{a}_l$ at **Fig. 4**). From the other side, IS itself consumes resources, which are represented at **Fig. 4** as set I containing n_{l_1} objects (of type) $a_{l_1}, \ldots, n_{l_p}$ objects (of type) a_{l_p}. By this, IS along with order also may be considered as "black box", i.e. device producing object q after being applied to initial resources set I. Any impact on IS eliminates some initial resources and devices entering this system, thus reducing its producing capability.

Let us underline, that there are no static interconnections between devices $B_1, \ldots, B_k$, and structure, representing IS in the described manner, is not graph, usually considered as a canonical form of such systems description and modelling (Burkart, 1997; Mills and Dabrowski, 2006; Hespanha et al., 2007; Levin et al., 2009; Dabrowski and Hunt, 2011; Mills et al., 2012; Carreras et al., 2005; Mills et al., 2011; Lade and Gross, 2012; Scheffer, 2009; Sheffer et al., 2009; Dabrowski et al., 2011; D'Andrea and Dullerud, 2003; Dullerud and Paganini, 2005; Goh and Yang, 2002; Horn and Johnson, 1991; Jadbabiae et al., 2003; Klavins et al., 2006; Mesbahi and Egerstedt, 2010; Mills and Dabrowski, 2008; Olfati-Saber et al., 2007; Stewart, W 1994). Here every order completion provides its own tree, including manufacturing operations and transfers of their results among devices, incorporated to this process. Moreover, there may be a lot of variants of each order completion, by reason IS may contain devices manufacturing similar objects in different alternative ways. So, every order completion process generates, in fact, each own cooperation, or contracts set, providing necessary items manufacturing.

Described formalization is basic for direct and reverse problems, verbally formulated higher, primary consideration. But it is not sufficient for strict mathematical description and, further, necessary algorithms design. There would be some constructive mathematical toolkit providing solutions of mentioned problems. But multiple attempts of well-known general-purpose tools, based on vector-matrix calculus, graphs theory, Markovian chains, Petri nets etc. (Burkart, 1997; Mills and Dabrowski, 2006; Hespanha et al., 2007; Levin et al., 2009; Dabrowski and Hunt, 2011; Mills et al., 2012; Carreras et al., 2005; Mills et al., 2011; Lade and Gross, 2012; Scheffer, 2009; Sheffer et al., 2009; Dabrowski et al., 2011; D'Andrea and Dullerud, 2003; Dullerud and Paganini, 2005; Goh and Yang, 2002; Horn and Johnson, 1991;

Jadbabiae et al., 2003; Klavins et al., 2006; Mesbahi and Egerstedt, 2010; Mills and Dabrowski, 2008; Olfati-Saber et al., 2007; Stewart, W 1994), application in the described area were not successful by reasons of problem's dimension, difficulties of IS manufacturing process precise representation and computational complexity. This lead us to the new approach, which is, by our opinion, more flexible and simple in application to the large-scale IS representation, analysis and synthesis, as well as more efficient from the computational complexity point of view, especially in high-parallel computing environments. This new approach is strongly based on the recursive multisets (MS) theory (Sheremet, 2010, 2011) – more precisely, multiset grammars (MG), – that's why is called "multigrammatical". Multiset grammars, namely, one of their possible dialects, – constraint multiset grammars (CMG) – were at first proposed in (Marriott, 1994; Marriott and Meyer 1997; Marriott, 1996) as a tool for recognition of visual objects with complex structure. CMG may be also considered as one of the problem-oriented constraints logical programming languages (Marriott and Stucky, 1998; Apt, 2003; Fruhkwirth and Abdennadher, 2003).

Authors' main contribution to this area are so called unitary multiset metagrammars (UMMG) (Sheremet, 2010, 2011; Sheremet and Zhukov, 2016), which are specific knowledge representation model, providing deep integration and convergence of classical optimization theory and modern knowledge engineering. As shown in (Sheremet, 2010, 2011; Sheremet and Zhukov, 2016), various subsets (subclasses) of UMMG provide efficient solution of various problems from system analysis and classical optimization areas.

Problem considered in this paper was announced in (Sheremet, 2016), and it is solved mainly by applying one of the simplest classes of UMMG family – unitary multiset grammars (UMG). Also general form multigrammars are applied to the mentioned reverse problem study.

We consider main results of UMG/MG application to the problem verbally formulated higher in a following way. Section 2 is dedicated to basic notions and definitions. Main elements of the approach (technological base, resource base, impact on industrial system) multiset/multigrammatical representation as well as IS sustainability (vulnerability) conditions are described in the Section 3, while Section 4 is dedicated to the NDI representation and mentioned conditions transformation to the corresponding form. Reverse problem concerning recognition of the abilities of the vulnerable IS is considered in Section 5. Inplementation issues are discussed in short in Section 6. Further directions of the development of multigrammatical approach practical applications are described at the conclusion.

2 Basic Notions and Definitions

According to (Calude et al., 2001; Petrovskiy, 2003; Banatre and Le Metoyer, 1993; Singh et al., 2007; Red'ko et al., 2015), **multiset** is set of so called multiobjects of the form $n \cdot a$, that means there are n identical objects (of type) a, and that is written as

$$\nu = \{\, n_1 \cdot a_1, \ldots, n_m \cdot a_m \,\}, \tag{1}$$

where ν is multiset name, $n_1 \cdot a_1, \ldots, n_m \cdot a_m$ are multiobjects entering ν, and integer numbers $n_1, \ldots, n_m$ are called multiplicities of $a_1, \ldots, a_m$ objects, all of which are different. (1) means that there are n_1 objects (of type) $a_1, \ldots, n_m$ objects (of type) a_m in multiset ν.

Unitary multiset grammar (UMG) is couple $S = <a_0, R>$, where a_0 is title object, and R is scheme – set of the so called **unitary rules** (UR) of the form

$$a \rightarrow n_1 \cdot a_1, \ldots, n_m \cdot a_m, \tag{2}$$

where a object is called head and list $n_1 \cdot a_1, \ldots, n_m \cdot a_m$ – body of the UR.

UMG were designed specially for the representation of hierarchical systems and objects, so the most valuable in the considered problem area is so called **structural interpretation** of the unitary rules. According to the structural interpretation, UR (2) means that object a consists of n_1 objects $a_1, \ldots, n_m$ objects a_m.

Example 1. Let $S = <car, R>$, where R contains two URs:

> *car* → 1 · *frame*, 1 · *engine*, 4 · *door*, 4 · *wheel*.
> *engine* → 1 · *motor*, 1 · *accumulator*, 1 · *transmission*.

As seen, car consists from one frame, one engine, four doors and four wheels. Engine, in turn, contains motor, accumulator as well as transmission. ∎

Technological interpretation of unitary rules is extension of the structural one in such a way that UR

$$a \rightarrow n_1 \cdot a_1, \ldots, n_m \cdot a_m, n'_1 \cdot b_1, \ldots, n'_k \cdot b_k, \tag{3}$$

represents not only structural components (spare parts) of the object a, which are multiobjects $n_1 \cdot a_1, \ldots, n_m \cdot a_m$, but also resources, necessary for assembling object a from these components, and being multiobjects $n'_1 \cdot b_1, \ldots \ldots, n'_k \cdot b_k$.

Example 2. Let $S = <car, R>$, where R contains one UR:

$$car \rightarrow 1 \cdot frame, 1 \cdot engine, 4 \cdot door, 4 \cdot wheel,$$
$$400 \cdot kW, 300 \cdot USD, 10 \cdot mnt - asm - line.$$

Here first four multiobjects of the UR body are the same, as in the first UR in the example 1, while the last three multiobjects define amounts of electrical energy (400 *kilowatt*), money (300 *dollars*) and time (10 *minutes of the assemblying line operation*), which are necessary for assembling car from its spare parts (*frame, engine*, 4 *doors*, 4 *wheels*). ∎

Unitary multiset grammars define systems (devices, processes etc.) in the easily understood top-down manner, and result of UMG application is set of multisets, each containing multiobjects with multiplicities defining total amounts of specific elementary (non-decomposed) objects having place in the system (device) or utilized while its manufacturing. Degree of decomposition is regulated by the analyst applying this tool while problem solving.

To define formally UMG semantics, i.e. process of generation of set of multisets V_S represented by UMG S, we shall use some operations and relations on multisets (Sheremet, 2010, 2011; Petrovskiy, 2003). Lower +, – and ∗ are symbols of multisets addition, subtraction and multiplication operations correspondingly, which are defined as follows:

$$\nu + \nu' = \left(\bigcup_{\substack{n \cdot a \in \nu \\ n' \cdot a \in \nu'}} \{(n + n') \cdot a\} \right) \cup$$
$$\cup \left(\bigcup_{\substack{n \cdot a \in \nu \\ a \notin \nu'}} \{n \cdot a\} \right) \cup \tag{4}$$
$$\cup \left(\bigcup_{\substack{n' \cdot a' \in \nu \\ a' \notin \nu}} \{n' \cdot a'\} \right).$$

$$\nu - \nu' = \bigcup_{\substack{n \cdot a \in \nu \\ n' \cdot a \in \nu' \\ n > n'}} \{(n - n') \cdot a\} \tag{5}$$

$$n \ast \nu = \bigcup_{m \cdot a \in \nu} \{(m \times n) \cdot a\} \tag{6}$$

Here symbols + , – and × denote usual arithmetic operations. In (4) we assume, that object a absence in multiset ν is equivalent to it's zero-value multiplicity.

Lower we shall designate by $\beta(\nu)$ set of all objects having place in multiset ν (it is called multiset basis):

$$\beta(\nu) = \{a \mid a \in \nu\}. \tag{7}$$

Symbol "$\subseteq$" denotes multisets inclusion relation, and, if $\nu' \subseteq \nu$, then ν' is called submultiset of ν. Formally

$$\nu' \subseteq \nu, \tag{8}$$

if

$$(\forall n' \cdot a \in \nu')(\exists n \cdot a \in \nu)\, n' \leq n. \tag{9}$$

Also we shall use multisets intersection operation denoted by bold symbol "$\cap$" and defined as follows:

$$\nu \cap \nu' = \bigcup_{\substack{n \cdot a \in \nu \\ n' \cdot a \in \nu'}} \{\min(n, n') \cdot a\}. \tag{10}$$

As in (4), $a \notin \nu$ and $0 \cdot a \in \nu$ are equivalent.

By A_s we shall designate set of all objects having place in UMG $S = <a_0, R>$, while by $\overline{A_s}$ – set of all terminal objects having place in S, i.e. objects which are not heads of unitary rules $r \in R$ and present only in URs bodies. As seen,

$$\overline{A_s} \subseteq A_s. \tag{11}$$

Multiset $\nu \in V_s$ is called **terminal multiset** (TMS), if there is no one UR in the scheme R, which may be applied to ν, i.e. it contains only terminal objects. Set of terminal multisets (STMS) $\overline{V_s} \subseteq V_s$ is subset of V:

$$\overline{V_s} \subseteq V_s. \tag{12}$$

Strict mathematical definition of the UMG application, i.e. set of multisets generation process, which we shall use here, is as follows (Sheremet, 2010, 2011; Sheremet and Zhukov, 2016):

$$V_{(0)} = \{\{1 \cdot a_0\}\}, \tag{13}$$

$$V_{(i+1)} = V_{(i)} \cup \left(\bigcup_{\nu \in V(i)} \bigcup_{r \in R} \{\pi(\nu, r)\} \right), \tag{14}$$

$$\pi(\nu, <a \rightarrow n_1 \times a_1, \dots, n_m \times a_m>) =$$
$$= \begin{cases} \nu - \{n \times a\} + n * \{n_1 \times a_1, \dots, n_m \times a_m\}, \text{if } n \times a \in \nu, \\ \{\varnothing\} \text{ otherwise}, \end{cases} \tag{15}$$

$$V_s = V_{(\infty)}, \tag{16}$$

$$\overline{V_s} = \{\nu \mid \nu \in V_s \,\&\, \beta(\nu) \subseteq \overline{A_s}\}, \tag{17}$$

where UR $a \rightarrow n_1 \cdot a_1, \dots, n_m \cdot a_m$ for unambiguity is represented in angle brackets, i.e. as $<a \rightarrow n_1 \cdot a_1, \dots, n_m \cdot a_m>$.

As seen, multisets generation is implemented by application to set of MS $V_{(i)}$, created at previous i steps, all unitary rules $r \in R$. In turn, every such UR is applied to all multisets $\nu \in V_{(i)}$ by special function π of two arguments, first being ν, and second – UR r in the form $a \rightarrow n_1 \cdot a_1, \dots, n_m \cdot a_m$. If ν contains multiobject $n \cdot a$, it is replaced by multiset $n * \{n_1 \cdot a_1, \dots, n_m \cdot a_m\}$ (this, of course, is followed by summarizing multiplicities of identical objects in the MS sum). Otherwise result is empty multiset. The described iterative process in general case is infinite, and set of multisets, defined by UMG $S = <a_0, R>$, is it's fixed point $V_{(\infty)}$, while set of terminal multisets, defined by S, is subset of V_s, defined by (17).

Let us note, that from computational complexity point of view (14) may be transformed to

$$V_{(i+1)} = V_{(i)} \cup$$
$$(\bigcup_{\nu \in V_{(i)}} \bigcup_{\substack{n \cdot a \, \exists \in \, \nu \\ <a \rightarrow n_1 a_1, \dots, \\ n_m \cdot a_m> \, \in R}} \{\nu - \{n \cdot a\} + n * \{n_1 \cdot a_1, \dots, n_m \cdot a_m\}\}), \tag{18}$$

where $\exists \in$ means selection of any one multiobject $n \cdot a$ from multiset ν. That provides sharp reduction of generation computational complexity (Sheremet, 2010, 2011).

Here we shall consider only so called **finitary UMG**, which generate finite STMS (Sheremet, 2010, 2011).

UMG $S = <a_0, R>$ is finitary, when there exists i such, that $V_{(i+1)} = V_{(i)}$, and if so, $V_{(i)} = V_S$. Problem of UMG finitarity recognition is algorithmically decidable (Sheremet, 2010, 2011).

Let us continue multigrammatical formalization of basic notions concerning the main subject of the paper.

Industrial system may be represented now by UMG $S = <tb, R>$, where tb (acronym from "technological base") is title object, and R is set of unitary rules in technological interpretation. **Order** completed by industrial system may be represented by multiset $q = \{n_1^q \cdot a_1^q, \dots \dots, n_l^q \cdot n_l^q\}$, and, evidently, resources amount, necessary for order q completion, is set of terminal multisets generated by UMG

$$S_q = <tbq, R \cup \{<tbq \rightarrow n_1^q \cdot a_1^q, \dots, n_l^q \cdot a_l^q>\}>, \tag{19}$$

i.e. $\overline{V}_{S_q}$ (for simplicity we shall use $\overline{V}_q$ instead of $\overline{V}_{S_q}$ lower). Because of possibility of multiple ways of some objects manufacturing, i.e. so called internal alternativity of S_q (Sheremet, 2010, 2011), there may be more than one TMS in $\overline{V}_q$.

Example 3. Let S be the same, as in the previous example, and $q = \{3 \cdot car\}$, i.e. IS must manufacture 3 cars. Then

$$S_q = <tbq, \{<tbq \rightarrow 3 \cdot car>, <car \rightarrow 1 \cdot frame, 1 \cdot engine, 4 \cdot wheel, 4 \cdot chair,$$
$$400 \cdot kW, 300 \cdot USD, 10 \cdot mnt-asm-line>\}>,$$

(here UR are written in the angle brackets for unambiguity) and, as seen,

$$\overline{V}_{sq} = \{\{3 \cdot frame, 3 \cdot engine, 12 \cdot wheel, 12 \cdot chair, 1200 \cdot kW, 900 \cdot USD, 30 \cdot mnt - asm - line\}\}. \blacksquare$$

Resource base of industrial system may be represented by multiset, which multiobjects define amounts of objects, being available for technological base. Evidently, resource base ν is **sufficient** for order q completion by IS $S = <tb, R>$, if there exists at least one set of resources amounts, necessary for this completion, being submultiset of ν:

$$(\exists \overline{v} \in \overline{V}_q) \, \overline{v} \subseteq \nu. \tag{20}$$

Example 4. Resource base

$$\nu = \{4 \cdot frame, 3 \cdot engine, 15 \cdot wheel, 12 \cdot chair, 1500 \cdot kW, 1200 \cdot USD, 50 \cdot mnt - asm - line\}$$

is sufficient for order q from the example 3 completion, while resource base

$$\nu = \{1 \cdot frame, 3 \cdot engine, 15 \cdot wheel, 12 \cdot chair, 1500 \cdot kW, 1200 \cdot USD, 50 \cdot mnt - asm - line\}$$

is not sufficient, because number of frames having place in the resource base, i.e. one, is less than it is necessary for 3 cars assemblying, i.e. three, although all other resources amounts are sufficient (they even exceed necessary values). $\blacksquare$

Impact on industrial system may be represented as multiset $\Delta\nu$, which defines resources amount eliminated from the resource base of IS, so the last after impact would be $\nu - \Delta\nu$.

All introduced notions and definitions provide formulation of the condition of IS sustainability/vulnerability to impact.

Let resource base ν is sufficient for order q completion by IS $S = \langle tb, R \rangle$, i.e. it satisfies (20). Then IS S completing order q with resource base ν is **sustainable** to impact $\Delta\nu$, if

$$(\exists \bar{\nu} \in \bar{V}_q)\; \bar{\nu} \subseteq \nu - \Delta\nu. \tag{21}$$

Otherwise IS S is **vulnerable** to impact $\Delta\nu$.

Example 5. IS $S = \langle tb, R \rangle$ from example 3, completing order $q = \{3 \cdot car\}$ with resource base from example 4, sufficient for this order completion, is sustainable to impact

$$\Delta\nu = \{3 \cdot wheel, 2 \cdot chair, 300 \cdot kW, 100 \cdot USD, 5 \cdot mnt - asm - line\}$$

because resource base after impact

$$\nu - \Delta\nu = \{4 \cdot frame, 3 \cdot engine, 12 \cdot wheel, 10 \cdot chair, 1200 \cdot kW, 1100 \cdot USD, 45 \cdot mnt - asm - line\}$$

is sufficient for completion. At the same time this IS is vulnerable to impact

$$\Delta\nu = \{1 \cdot engine\},$$

because resource base after impact

$$\nu = \{4 \cdot frame, 2 \cdot engine, 15 \cdot wheel, 12 \cdot chair, 1500 \cdot kW, 1200 \cdot USD, 50 \cdot mnt - asm - line\}$$

is not sufficient for completion (number of engines is less than necessary). ∎

3 Natural Disasters Impacts on Industrial System

Until now we have considered impacts without their specific features. Natural disaster impacts most general feature is their localization, i.e. connection with some fixed areas (points) affected by the NDI. However, any information about technological base, as well as resource base, elements locations, in the considered higher UMG representation of industrial systems is not included.

Let us extend unitary rules in technological interpretation by geospatial information in such a way, that every object, having place in UR, would have form a/z, where z is locator defining area (point), where this object presents, and "/" is divider, which is not used anywhere in a and z strings. So UR would have the following form:

$$a/z \rightarrow n_1 \cdot a_1 / z_1, \ldots, n_m \cdot a_m / z_m, \tag{22}$$

that means object a at location z may be produced (assembled) if there are n_1 objects a_1 at location $z_1, \ldots, n_m$ objects a_m at location z_m. (Objects like $a/z, a_1/z_1, \ldots, a_n/z_n$ are called "composite objects", or "composits").

Similarly, we shall extend resource base and order representation, which would become respectively

$$\nu = \{n_1 \cdot a_1 / z_1, \ldots, n_k \cdot a_k / z_k\}, \tag{23}$$

$$q = \{n_1^q \cdot a_1^q / z_1^q, \ldots, n_l^q \cdot a_l^q / z_l^q\}. \tag{24}$$

After that there is a simplest way for NDI representation, namely, by the set $Z = \{z_1, \ldots, z_k\}$, of locations destroyed by this NDI completely. Corresponding relation for NDI in the multiset form is constructed directly:

$$\Delta\nu(Z) = \{n \cdot a \mathbin{/} z \mid n \cdot a \mathbin{/} z \in \nu \ \& \ z \in Z\}, \tag{25}$$

and

$$\nu - \Delta\nu(Z) = \{n \cdot a \mathbin{/} z \mid n \cdot a \mathbin{/} z \in \nu \ \& \ z \notin Z\}. \tag{26}$$

From (23)–(24) and Z definition we may write the condition of IS sustainability/vulnerability, which is evident generalization of (21). If

$$(\exists \bar{\nu} \in \bar{V}_q) \ \bar{\nu} \subseteq \nu - \Delta\nu(Z), \tag{27}$$

then IS $S = <tb, R>$, completing order with resource base ν, is **sustainable** to NDI Z. Otherwise IS is **vulnerable** to Z.

Example 6. Let us consider IS, which resource base is

$\nu = \{10 \cdot a_1/z_1, 5 \cdot a_2/z_2, 19 \cdot a_3/z_3, 7 \cdot a_4/z_4\},$

technological base is represented by scheme R containing two unitary rules

$a/z_1 \rightarrow 3 \cdot a_1/z_1, 2 \cdot a_2/z_2, 7 \cdot a_3/z_3,$
$a/z_1 \rightarrow 2 \cdot a_2/z_2, 3 \cdot a_4/z_4,$

and order q completed by this IS is $q = \{2 \cdot a_1/z_1\}$.

Natural disaster impact $Z = \{z_3\}$ causes destruction of subset of the resource base, located at z_3, so according to (25),

$\Delta\nu(Z) = \{19 \cdot a_3/z_3\},$

and

$\nu - \Delta\nu(Z) = \{10 \cdot a_1/z_1, 5 \cdot a_2/z_2, 7 \cdot a_4/z_4\}.$

As seen,

$\bar{V}_q = \{\nu_1, \nu_2\},$

where

$\nu_1 = \{6 \cdot a_1/z_1, 4 \cdot a_2/z_2, 14 \cdot a_3/z_3\},$
$\nu_2 = \{4 \cdot a_2/z_2, 6 \cdot a_4/z_4\},$

and

$\nu_1 \not\subseteq \nu - \Delta\nu(Z),$
$\nu_2 \not\subseteq \nu - \Delta\nu(Z),$

that's why IS is sustainable to NDI Z. ∎

Note, that NDI may destruct facilities not always completely, but very often partially. In this case some objects located at the area, affected by the NDI, may remain in the undestructed state and thus may be used elements for manufacturing some products.

This case may be represented in the direct manner:

$$\Delta\nu(Z) = \{\Delta n_{i_1} \cdot a_{i_1} \mathbin{/} z_{i_1}, \dots, \Delta n_{i_p} \cdot a_{i_p} \mathbin{/} z_{i_p}\}, \tag{28}$$

that provides use of condition (27) for the IS sustainability/vulnerability assessment.

Example 7. Let us consider IS from the previous example 7, and NDI, partially destructing locations z_3 and z_4, so that

$$\Delta\nu(Z) = \{12 \cdot a_3/z_3, \, 2 \cdot a_4/z_4\},$$

and

$$\nu - \Delta\nu(Z) = \{10 \cdot a_1/z_1, \, 5 \cdot a_2/z_2, \, 7 \cdot a_3/z_3, \, 5 \cdot a_4/z_4\}.$$

As seen, because of

$$\nu_1 \not\subseteq \nu - \Delta\nu(Z)$$
$$\nu_2 \not\subseteq \nu - \Delta\nu(Z)$$

IS is vulnerable to NDI $\Delta\nu(Z)$. ∎

4 Reverse Problem

Let us consider the case, when IS R, completing order q with resource base ν, is vulnerable to NDI (Z or $\Delta\nu(Z)$).

The question is as follows: what part of order q may be completed by IS R affected by NDI Z?

Simple, but, however, non-evident approach to this question consideration is based on the general form multiset grammars use for constructing the solution.

Multiset grammar is couple $S = \,<\nu_0, R>$, where ν_0 is multiset called kernel, and R is, as in the UMG, scheme, which elements are called rules, which structure is

$$\nu \rightarrow \nu', \tag{29}$$

where ν and ν' are multisets. Semantics of multigrammars is similar to UMG semantics (Sheremet, 2010, 2011):

$$V_{(0)} = \{\nu_0\}, \tag{30}$$

$$V_{(i+1)} = V_{(i)} \cup \left(\bigcup_{\nu \in V_{(i)}} \bigcup_{r \in R} \{\pi(\nu, r)\} \right), \tag{31}$$

$$\pi(\nu, \bar{\nu} \rightarrow \bar{\nu}') = \begin{cases} \nu - \bar{\nu} + \bar{\nu}', & \text{if } \bar{\nu} \subseteq \nu, \\ \{\varnothing\} & \text{otherwise}, \end{cases} \tag{32}$$

$$V_s = V_{(\infty)}, \tag{33}$$

$$\bar{V}_s = \{\nu \,|\, \nu \in V_s \, \& \, (\forall r \in R)\, \pi(\nu, r) = \{\varnothing\}\}. \tag{34}$$

As seen, MG provides generation of new multisets from already generated, beginning from the kernel, by replacement of their submultisets in accordance with rules having place in the scheme. Generation is executed until it is possible, so, in general case, V_s and $\bar{V}_s$ may be infinite. However, if there exists i such, that $V_{(i)} = V_{(i+1)}$, then $V_{(i)} = V_s$, then both V_s and $\bar{V}_s$ are finite sets, and MG is finitary.

Let us consider UMG $S = \,<tb, R>$, representing IS with technological base R, and this IS resource base ν. Multiset grammar $S_\nu^* = \,<\nu, R^*>$ will be called **dual** to unitary multiset grammar $S = \,<tb, R>$, if

$$R^* = \{\{n_1 \cdot a_1, \ldots, n_m \cdot a_m\} \rightarrow \{1 \cdot a\} \,| \\ < a \rightarrow n_1 \cdot a_1, \ldots, n_m \cdot a_m > \in R\}, \tag{35}$$

i.e. every unitary rule $a \rightarrow n_1 \cdot a_1, \ldots, n_m \cdot a_m$ having place in the UMG S scheme R is substituted by one and the only one "mirror" rule in MG S_ν^* scheme R^*.

As may be seen, set of multisets $V_{S_\nu^*}$, generated by MG S_ν^*, contains all variants of production, which may be manufactured by IS S, beginning from the resource base ν. Set of variants of order q partial completion is, thus, subset of set generated by MG S_ν^*, each TMS of which contains at least one object from q order (here and lower we denote mentioned set of variants as V (R, ν, q)):

$$\text{V}(R, \nu, q) = \{\bar\nu \mid \bar\nu \in V_{S_\nu^*} \, \& \, q \cap \bar\nu \neq \{\varnothing\} \}. \tag{36}$$

As a consequence, total set of solutions of the reverse problem is

$$\begin{aligned} &\text{V}(R, \nu - \Delta\nu\,(Z),\, q) = \\ &= \{\bar\nu \mid \bar\nu \in V_{S_{\nu-\Delta\nu(z)}^*} \, \& \, q \cap \bar\nu \neq \{\varnothing\} \}. \end{aligned} \tag{37}$$

Example 8. Let us consider IS with the same scheme R and resource base ν as in the previous examples 7 and 8, completing order $q = \{ 3 \cdot a/z_1 \}$, and being affected by the same NDI $\Delta\nu(Z)$ as in the example 8. Kernel ν_0 of MG $S^*= <\nu_0, R^*>$, dual to UMG $S = < tb, R>$, is

$$\nu_0 = \nu - \Delta\nu(Z) = \{10 \cdot a_1/z_1,\, 5 \cdot a_2/z_2,\, 7 \cdot a_3/z_3,\, 5 \cdot a_4/z_4\},$$

while rules are

$$r_1 \equiv \{3 \cdot a_1/z_1,\, 2 \cdot a_2/z_2,\, 7 \cdot a_3/z_3\} \rightarrow \{1 \cdot a/z_1\},$$
$$r_2 \equiv \{2 \cdot a_2/z_2,\, 3 \cdot a_4/z_4\} \rightarrow \{1 \cdot a/z_1\},$$
$$r_2 \equiv \{3 \cdot a/z_1\} \rightarrow \{1 \cdot tbq\}.$$

According to (36),

$$\text{V}(R, \nu - \Delta\nu(Z), q) = \{\nu_1, \nu_2, \nu_3\},$$

where

$$\nu_1 = \{7 \cdot a_1/z_1,\, 3 \cdot a_2/z_2,\, 5 \cdot a_4/z_4,\, 1 \cdot a/z_1\},$$
$$\nu_2 = \{5 \cdot a_1/z_1,\, 3 \cdot a_2/z_2,\, 2 \cdot a_4/z_4,\, 2 \cdot a/z_1\},$$
$$\nu_3 = \{10 \cdot a_1/z_1,\, 3 \cdot a_2/z_2,\, 7 \cdot a_3/z_3,\, 2 \cdot a_4/z_4,\, 1 \cdot a/z_1\}.$$

As seen, ν_1 is result of application of r_1 to ν_0; ν_2 – result of application of r_2 to ν_1; while ν_3 is result of application of r_2 to ν_0.

So, a valuable part of the order (2 of 3 objects a located at z_1) may be completed by the affected IS, and even some valuable part of the resource base would remain after following this way of order completion. ∎

However, in general case some of multisets entering set V $(R, \nu - \Delta\nu(Z), q)$ may be of no practical use (as ν_1 and ν_3 in the previous example 8), because the only purpose of the assessment is to get the best ("valuable" in the common sense) variants of order completion, which are not improvable from the practical point of view.

To filter STMS, constructed in accordance with (37), we shall define one useful function **max**, which value **max**(V) is subset of V including only so called non-dominated multisets entering V:

$$\textbf{max}(V) = \{\nu \mid \nu \in V \, \& \, (\not\exists \nu' \in V)\, \nu \subset \nu'\}, \tag{38}$$

i.e. there is no one multiset in **max**$(V) \subseteq V$, which is submultiset of some other multiset entering V; **max**(V) is thus set of maximal elements of V.

Example 9. Let $V = \{\nu_1, \nu_2, \nu_3, \nu_4\}$, where

$\nu_1 = \{3 \cdot a_1, 2 \cdot a_2, 1 \cdot a_3\}$,
$\nu_2 = \{1 \cdot a_1, 2 \cdot a_2\}$,
$\nu_3 = \{2 \cdot a_1, 1 \cdot a_3\}$,
$\nu_4 = \{1 \cdot a_1, 3 \cdot a_2\}$.

As may be seen, according to (38), because of $\nu_2 \subseteq \nu_1$, $\nu_3 \subseteq \nu_1$, $\mathbf{max}(V) = \{\nu_1, \nu_4\}$. ∎

From here set of valuable ("precise") solutions of reverse problem, denoted $\overline{V}(R, \nu - \Delta\nu(Z), q)$ may be constructed in a following evident way:

$$\overline{V}(R, \nu - \Delta\nu(Z), q) = $$
$$= \mathbf{max}(\{\overline{\nu} \cap q \mid \overline{\nu} \in V(R, \nu - \Delta\nu(Z), q)\}). \tag{39}$$

As seen, set of valuable solutions includes those only intersections of elements of $V(R, \nu - \Delta\nu(Z), q)$ and q order, which are non-dominated multisets.

Example 10. Consider $V(R, \nu - \Delta\nu(Z), q) = \{\nu_1, \nu_2, \nu_3\}$ and $q = \{3 \cdot a/z_1\}$ from Example 9. According to (39),

$\overline{V}(R, \nu - \Delta\nu(Z), q) = \mathbf{max}(\{\nu_1 \cap q, \nu_2 \cap q, \nu_3 \cap q\}) =$
$\quad = \mathbf{max}(\{\{1 \cdot a/z_1\}, \{2 \cdot a/z_1\}, \{\varnothing\}\}) = \{\{2 \cdot a/z_1\}\}$. ∎

Note, that general form multigrammars application instead of unitary MG provides generation of all possible variants of work distribution between alternative facilities (devices), not only monopolial ones. However, this opportunity as a consequence leads to sensitive increasing of computational complexity of generation mentioned. Theory and implementation of this complexity reduction would be considered in separate publications.

5 Implementation issues

Software implementation of IS sustainability assessment ("direct problem") employs knowledge base (R set of unitary rules) representation and storage in a form of data base, which elements are selected by means of NoSQL-like (McCreary and Kelly, 2013; Vaish, 2013) query language.

Data base mentioned is set of records being couples of the form $< a, B >$, where a is object (key of the record) while B is set of bodies of unitary rules, which head is a:

$$B = \{B_1, \ldots, B_m\}, \tag{40}$$

where

$$B_i = \{< a_1^i, n_1^i >, \ldots, < a_{k_1}^i, n_{k_1}^i >\}, \tag{41}$$

so that (40)–(41) represent m unitary rules

$$a \rightarrow a_1^1, n_1^1, \ldots, a_{k_1}^1, n_{k_1}^1,$$
$$\ldots \tag{42}$$
$$a \rightarrow a_1^m, n_1^m, \ldots, a_{k_m}^m, n_{k_m}^m.$$

While generation, set B is selected from data base by query, which processing is implemented by special function $READ(a)$. Similarly, insertion of new unitary $a \rightarrow w$ to the knowledge base is implemented by function $INSERT(a, w)$, as well as UR $a \rightarrow w$ deletion from KB (DB) is implemented by function $DELETE(a, w)$. Also there is function $DELALL(a)$, eliminating from KB all unitary rules with head a, i.e. deletion from data base record $<a, B>$, where B is current set of bodies of URs with head a.

Associative internal organization of the described data base along with physical blocks cash, supporting DB management via virtual memory space, provides fast execution of functions mentioned without redundant search (Sheremet, 2013).

Software implementation of assessment of part of the order, which may be completed by the affected vulnerable IS ("reverse problem") employs knowledge base (scheme R^* of MG, dual to the initial UMG) representation and storage in a form of data base similar to the considered higher.

However, this data base contains records of two types:

1) triples of the form $<r, w, a>$, where r is unique identifier (key of the record) of the "mirror" rule $w \rightarrow \{1 \cdot a\}$, where w multiset is represented as in (41);
2) couples of the form $<a, w>$, where a is object (key of the record), while w is set of couples $<r, n>$, where r is identifier of rule, which left part contains multiobject $n \cdot a$.

Example 11. Let dual UMG R^* contains following "mirror" rules:

$r_1: \{3 \cdot a_1, 2 \cdot a_2\} \rightarrow \{1 \cdot a_3\}$,
$r_2: \{2 \cdot a_1, 5 \cdot a_3\} \rightarrow \{1 \cdot a_0\}$.

Then corresponding data base will contain following records:

$< r_1, \{<a_1, 3>, <a_2, 2>\}, a_3>$,
$< r_2, \{<a_1, 2>, <a_3, 5>\}, a_0>$,
$< a_1, \{<r_1, 3>, <r_2, 2>\} >$,
$< a_2, \{<r_1, 2>\} >$,
$< a_3, \{<r_2, 5>\} >$. ∎

As may be seen, quantity of records of the first type is $|R^*|=|R|$ while quantity of records of the second type is equal to number of objects having place in left parts of rules entering R^*.

Records of the first type are used while generation of the TMS (i.e. solutions) set in accordance with (36)-(38) by application of "mirror" rules. If this would be done without any improvements, there would be full search of all $|R^*|$ rules at every generation step in order to apply some of them to the current multiset, generated while previous steps execution. So for practically interesting knowledge bases with $|R^*|$, beginning from $10^5 - 10^6$, direct implementation of generation algorithmics is unreal. To avoid this principal difficulty, records of the second type are introduced. They are stored and selected by key (object a) in such a way, that by one access to data base identifiers of all "mirror" rules, which possibly may be applied to current multiset by reason there is multiobject $n \cdot a$ in their left parts, and $n \leq m$, where $m \cdot a$ enters current multiset. This techniques provides cut-off all the rest "mirror" rules, non-applicable to this MS by reason of object a absence in their left parts. There are several implementations of this basic idea, and they would be considered separately.

Described approach to software implementation of the proposed algorithmics is somewhat efficient from the practical point of view, that is verified by real software-intensive IS management experience (Sheremet, 2005; Sheremet, 2005; Karasev and Sheremet, 2008). Such software operates CALS-originated knowledge bases, which contain structural descriptions of all types of objects, manufactured by various facilities, entering distributed IS. Due to "granularity" of the described knowledge representation, all accumulation and correction of such KB from the distributed local sources is performed without any difficulties in the online regime. Assessment of the possibility of orders completion at moments, when they enter IS, is performed in soft real time mode (minutes per message) on knowledge bases containing 10–12 mln. unitary rules with 2–3 mln. type of objects (items) used and manufactured by the industrial system, and this mode does not require special-purpose or high-parallel hardware. Reverse problem software is activated, when various impacts like manufacturing equipment malfunctions and necessary resources delays occur, and is solved on the same hardware in practically the same time scale. More detailed description of the UMG toolkit software implementation in various hardware environments may be object of the special paper.

6 Conclusion

Multigrammatical approach shortly presented in the paper may be efficiently used for the assessment of consequences of natural disasters impacts (as well as human-implemented impacts) on large-scale industrial systems, no matter what production they are manufacturing, what infrastructure they use, what kind of producing devices they contain etc., due to the generality and flexibility of MG/UMG toolkit used for the assessment.

The main four directions of further development of the approach, in our opinion, are:

1) application of so called temporal MG/UMG with parallel processes modelling features to more deep and adequate assessment of NDI consequences;
2) implementation of "what-if" regimes for decision makers;
3) minimization of computational complexity of multisets generation as well as high-parallel generation algorithms development for grid/cloud hardware;
4) development of "Big Knowledge" paradigm and unconventional computational models for its hardware implementation.

The first direction background is following. As it is easy to see, multigrammatical approach is strongly based on the additivity of quantities of objects represented by their multiplicities. However, time is additive only in consideration to one processing (manufacturing) unit (for example, assemblying line): if there are two or more such units, they may operate in parallel, that's why total work may be completed by the system faster than in the sequential mode with additive time operating. Temporal MG/UMG are such generalization of the considered higher mathematical tools, that provide simple description of industrial systems and their elements extended by time intervals which are necessary to these elements to execute operations. Basic construction of temporal unitary multiset grammar is so called temporal unitary rule of the form

$$a \rightarrow n \cdot t, \; n_1 \cdot a_1, \; \ldots, \; n_m \cdot a_m,$$

(43)

where t is fixed object denoting time measurement unit (for example, minute) while n multiplicity is duration of time interval (number of t objects), necessary for assemblying a object by utilizing n_1 units of resource $a_1, \ldots, n_m$ units of resource a_m. This extension provides as a final purpose construction of Gantt diagrams charts of manufacturing (production) processes evolving all devices which are necessary for order completion, that's why temporal MG/UMG algorithmics is in fact algorithmics of optimal (rational) scheduling (Conway et al., 2003; Hermann, 2006). It is much more deep and sophisticated in comparison with MG/UMG, however, providing practically useful generalization of well known scheduling problems and their solutions.

The second of the listed directions is from the practical point of view very important, because it provides end users (decision makers) with the opportunity to prepare to possible NDI before they really occur.

The third direction, being quite evident, follows from the non-procedural, declarative representation of the knowledge about industrial systems and their operation items. "Granularity" of multigrammatical knowledge bases provides easy, in fact, "additive" accumulation of knowledge as well as its correction.

However, like any other knowledge representation model, MG/UMG need special algorithmics providing minimization of redundant search while multisets generation, especially for the so called filtering MG/UMG (Sheremet, 2010, 2011; Sheremet and Zhukov, 2016; Sheremet and Zhukov, 2016). This direction is, perhaps, most critical, because it eliminates or mitigates sharp growth of redundant generation steps, which reason is well-known combinatorial explosion while knowledge base volume expansion. There would be combination of two basic approaches for such kind of problems solution: redundant steps elimination by some smart cutting-off conditions exploitation at maximal early stages of multisets generation, and "brutal force" application by parallel generation of alternative branches on asynchronously operating processor units. State of the art in this area is described in (Sheremet 2010, 2011).

The most general is, in our opinion, the fourth direction. As may be easily estimated, real large-scale industrial systems technological bases and IS production (manufactured objects) would be represented by UMG knowledge bases containing millions and millions unitary rules. Creating, maintaining and utilizing such amounts of knowledge is a great problem being the next step of computer technologies application after Big Data, which are in fact everyday reality, however, not yet understood (Roberts, 2016) (as may be seen from section V, when UMG are used, then **Big Knowledge** is implemented by **Big Data** tools). In many considerations, it would be a new way of thinking, which must lead us to the new understanding of the global technosphere as an interconnected set of devices, joined with one another by information infrastructure (that's already "Internet of Things") and logistical networks. Such understanding, in turn, may optimize humanity behavior and its relations with the nature.

Acknowledgement

Competing Interests

Author Information

Igor Sheremet, Doct. Eng. Sci., Professor, Corresponding Member of Russian Academy of Science (RAS). Head of the department of Financial University under the Government of Russian Federation, deputy director for science of Russian Foundation for Basic Research. Works at areas of theoretical and applied computer science, systems analysis, data and knowledge engineering, robotics, cybersecurity. Vice-chairman of RAS Board on Robotics and Mechatronics, member of RAS Systems Analysis Committee, official member of the Scientific Advisory Committee of International Institute of Applied Systems Analysis (IIASA) at Laxenburg (Austria).

References

Apt, K 2003 Principles of Constraints Programming. Cambridge University Press, p. 420. DOI: https://doi.org/10.1017/CBO9780511615320

Banatre, J-P and **Le Metoyer, D** 1993 Programming by Multiset Transformation. *Communications of the ACM*, 36(1): 98–111. DOI: https://doi.org/10.1145/151233.151242

Burkart, O 1997 *Automatic Verification of Sequential Infinite-State Processes. Lecture Notes in Computer Science*, Vol. 1354, p. 163. DOI: https://doi.org/10.1007/3-540-69678-4

Calude, C S, Păun, G, Rozenberg, G and **Salomaa, A** 2001 *Multisets Processing: Mathematical, Computer Science and Molecular Computing Points of View – Lecture Notes in Computer Science*, Vol. 2235. NY: Springer, p. 359. DOI: https://doi.org/10.1007/3-540-45523-X

Carreras, B, Lynch, V, Dobson, J and **Newman, D** 2005 Critical Points and Transitions in Electric Power Transmission Model for Cascading Failure Blackouts. *Chaos*, 12(4): 985–994. DOI: https://doi.org/10.1063/1.1505810

Conway, R W, Maxwell, W L and **Miller, L W** 2003 Theory of Scheduling (Second Edition). Dover Publications, p. 304.

D'Andrea, R and **Dullerud, G** 2003 Distributed Control Design for Spatially Interconnected Systems. *IEEE Transactions on Automatic Control*, 48(9): 1478–1495. DOI: https://doi.org/10.1109/TAC.2003.816954

Dabrowski, C and **Hunt, F** 2011 Using Markov Chain and Graph Theory Concepts to Analyze Behaviour in Complex Distributed Systems. In: *Proc. European Modelling and Simulation Conference* (EMSS). Rome, Italy, pp. 658–669.

Dabrowski, C, Hunt, F and **Morrison, K** 2011 Improving the Efficiency of Markov Chain Analysis of Complex Distributed Systems. *National Institute of Standards and Technology Interagency Report* 7744, 2011. p. 81.

Dullerud, G and **Paganini, F** 2005 *A Course in Robust Control Theory: A Convex Approach*. NY: Springer Verlag, p. 398.

Fruhkwirth, T and **Abdennadher, S** 2003 *Essentials of Constraint Programming*. Berlin, Springer–Verlag, p. 147. DOI: https://doi.org/10.1007/978-3-662-05138-2

Goh, C J and **Yang, X Q** 2002 *Duality in Optimization and Variational Inequalities*. NY: Taylor & Francis, p. 330. DOI: https://doi.org/10.1201/9781420018868

Hermann, J W 2006 *Handbook of Production Scheduling*. New York: Springer, p. 318. DOI: https://doi.org/10.1007/0-387-33117-4

Hespanha, J, Naghstabrizi, P and **Xu, Y** 2007 A Survey of Recent Results in Networked Control Systems. *Proceedings of the IEEE*, 95(1): 138–162. DOI: https://doi.org/10.1109/JPROC.2006.887288

Horn, R A and Johnson, C R 1991 *Matrix Analysis*. NY:Cambridge University Press, p. 287.

Jadbabiae, A, Lin, J and **Morse, A** 2003 Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules. *IEEE Transactions on Automatic Control*, 48(6): 988–1001.

Karasev, R S and **Sheremet, I A** 2008 Assessment of the Goal-Driven Program Duration with Shifted Terms of Its Contracts. *Defense Technology Issues*, 1: 31–40 (In Russian).

Klavins, E, Christ, R and **Lipsley, D** 2006 A Grammatical Approach to Self-Organizing Robotic Systems. *IEEE Transactions on Automatic Control*, 51(6): 949–962. DOI: https://doi.org/10.1109/TAC.2006.876950

Levin, D A, Peres, Y and **Wilmer, E L** 2009 *Markov Chains and Mixing Times.* American Mathematical Society, p. 371.

Lade, S J and **Gross, T** 2012 Early Warning Signals for Critical Transitions: A Generalized Modeling Approach. *Computational Biology*, 8(2): e1002360. DOI: https://doi.org/10.1371/journal.pcbi.1002360

Marriott, K 1994 Constraint Multiset Grammars. *Proc. IEEE Symposium on Visual Languages.* IEEE Computer Society Press, pp. 118–125. DOI: https://doi.org/10.1109/VL.1994.363633

Marriott, K 1996 Parsing Visual Languages with Constraint Multiset Grammars. In: *Programming Languages: Implementation, Logic and Programs. – Lecture Notes in Computer Science*, Vol. 1292, p. 419.

Marriott, K and **Meyer, B** 1997 On the Classification of Visual Languages by Grammar Hierarchies. *Journal of Visual Languages and Computing*, 8(4): 375–402. DOI: https://doi.org/10.1006/jvlc.1997.0053

Marriott, K and **Stucky, P G** 1998 *Programming with Constraints: An Introduction.* MIT Press, p. 482.

Mesbahi, M and **Egerstedt, M** 2010 *Graph-Theoretic Methods in Multiagent Networks.* Princeton University Press, p. 424. DOI: https://doi.org/10.1515/9781400835355

McCreary, D and **Kelly, A** 2013 *Making Sense of NoSQL: A Guide for Managers and the Rest of Us.* Manning Publications, p. 312.

Mills, K and **Dabrowski, C** 2006 *Investigating Global Behaviour in Computing Grids. Lecture Notes in Computer Science*, Vol. 4124. Springer, pp. 120–136.

Mills, K and **Dabrowski, C** 2008 Can Economics-Based Resource Allocation Prove Effective in a Computation Marketplace? *Journal of Grid Computing*, 6(3): 291–311. DOI: https://doi.org/10.1007/s10723-007-9094-4

Mills, K, Filliben, J, Cho, D-Y and **Schwartz, E** 2011 Predicting Macroscopic Dynamics in Large Distributed Systems. *Proc. ASME 2011 Conference on Pressure Vessels & Piping.* Baltimore, MD, July 17–22, 2011.

Mills, K, Filliben, J and **Dabrowski, C** 2012 *Predicting Global Failure Regimes in Complex Information Systems.* DoE COMBINE Workshop, September 19, 2012.

Olfati-Saber, R, Fax, J A and **Murray, R M** 2007 Consensus and Cooperation in Networked Multiagent Systems. *Proceedings of the IEEE*, 95(1): 215–233. DOI: https://doi.org/10.1109/JPROC.2006.887293

Petrovskiy, A B 2003 *Spaces of Sets and Multisets.* Moscow: Editorial URSS, p. 248 (In Russian).

Red'ko, V N, Bui, D B and **Grishko, Yu** 2015 A. Modern State of Multisets Theory from the Entity Point of View. *Cybernetics and System Analysis*, 51(1): 171–178.

Roberts, F 2016 *What is Big Data and How has it Changed. – Invited talk at International Conference on Data Intensive Systems Analysis for Geohazard Studies.* Sochi, Russia, July 18–21, 2016.

Scheffer, M 2009 *Critical Transitions in Nature and Society.* Wiley, p. 400.

Sheffer, M, Bascompte, J, Brock, W A, Brovkin, V, Carenter, S R, Dakos, V, Held, V, Van Nes, E H, Rietkerk, M and **Sugihara, G** 2009 Early Warning Signals for Critical Transitions. *Nature*, 461(3): 53–59. DOI: https://doi.org/10.1038/nature08227

Sheremet, I A 2005 *On the Approach to Goal-Driven Planning in Defense Acquisition Area. Defense Technology Issues*, 5: 5–14 (In Russian).

Sheremet, I A 2005 *Goal-Driven Planning Based on Recursive Generation of Contracts Chains.* Moscow: FDAS, 42 p. (In Russian).

Sheremet, I A 2010 *Recursive Multisets and Their Applications.* Moscow: Nauka, p. 293 (In Russian).

Sheremet, I A 2011 *Recursive Multisets and Their Applications.* Berlin: NG Verlag, p. 249.

Sheremet, I A 2013 *Augmented Post Systems: The Mathematical Framework for Knowledge and Data Engineering in Network-Centric Environment.* Berlin: EANS, p. 395.

Sheremet, I A 2016 Multiset Approach to the Estimation of Consequences of Natural Disaster Impacts on Industrial Systems. *Geoinformatics Research Papers*, 4: BS4002, DOI: https://doi.org/10.2205/2016BS01Sochi

Sheremet, I and **Zhukov, I** 2016 Optimizing Multiset Metagrammars. Formal Definitions. *International Journal of Computer Engineering and Information Technology*, 8(7): 106–114.

Sheremet, I and **Zhukov, I** 2016 Optimizing Multiset Metagrammars. Multigrammatical Representation of Classical Optimization Problems. *International Journal of Computer Engineering and Information Technology*, 8(7): 115–124.

Singh, D, Ibrahim, A M, Yohanna, T and **Singh, J N** 2007 An Overview of Applications of Multisets. *Novi Sad Journal of Mathematics*, 37(2): 73–92.

Stewart, W 1994 *Introduction to the Numerical Solutions of Markov Chains.* Princeton University Press, p. 293.

Vaish, G 2013 *Getting Started with NoSQL.* Packt Publishing, p. 142.