Multifunctional Remote Laboratories for Real Experiments, Engineering Processes, and Manufacturing Methods

http://dx.doi.org/10.3991/ijoe.v10i5.3726

S.M. Krylov
Samara State Technical University, Samara, Russia

Abstract—**This paper describes the efficient structures of various multifunctional remote laboratories for real experiments, engineering processes, and manufacturing methods. Some theoretical problems concerning functional completeness, universality, flexibility, and programmability of such remote laboratories and facilities are discussed in the framework of a new interdisciplinary scientific approach called general formal technology (GFT). GFT is a branch of general system theory (GST). It correlates with Alexander Bogdanov's Tektology and can be used to investigate formal algorithmic and physical system structures for the synthesis and analysis of various objects.**

Index Terms—**General system theory, flexible remote manufacturing; functional completeness; real remote laboratories; remote experimentation; system universality; system flexibility; Turing machine.**

I. INTRODUCTION

Remote accessible labs for real experiments (RALREs) or real remote laboratories (RRLs) are an important topic of discussion in current scientific and pedagogical papers [1-6]. The possible benefits of such laboratories are as follows:

1. Conducting experiments from any location that has an Internet-connected computer.
2. 24-hour access to remote experimentation facilities.
3. The cost-effective use of remote experimental resources, including resources that would otherwise be prohibitively expensive.
4. The higher reliability of results obtained experimentally (compared to possible unpredictable artifacts in virtual experimentations due to incorrect modeling of investigated effects or objects).
5. The incorporation of new Internet facilities, including Internet libraries, databases, and search engines.

In the above list of RRL benefits, the most important point is #4 because it guarantees us real results that are necessary for suitable education. However, this reliability is also necessary for the remote manufacturing of various real objects and for real experimentation in various technological and scientific areas. Moreover, such RRLs, as well as remote manufacturing plants (remote manufactures for short) and remote experimental facilities should be extremely flexible (i.e., extremely multifunctional, or *universal*) in order to be effectively utilized. We will call such flexible (*universal*) remote manufactures and flexible (*universal*) remote experimental facilities FRMs and FREFs, respectively.

These RRLs, FRMs and FREFs should possess complete distant programmability, including necessary informational feedback to the control of remotely implemented processes.

The use of RRLs could also cause some problems, such as:

a) The "lack of modular approach in designs" [1, 5] or, more generally, the absence of clear and theoretically approved architectural concepts for multifunctional RRLs;

b) The inability to vary the laboratory instruments for a wide range of needs [1, 3, 6];

c) Administrative, pedagogical, and industrial problems [1].

Both case b) and, in part, case a) could also present complications in the use and development of FRMs and FREFs.

We will not investigate item c), as those problems are temporary. We will concentrate on items a) and b) because both are also important for FRMs and FREFs. General formal technology could help us to understand and solve the above problems [7-9, 11-13].

II. GENERAL FORMAL TECHNOLOGY

A. *Theoretical Foundations*

Any formal technology **T** is represented in general formal technology (GFT) frameworks in a similar manner as the formal representation of an algebraic system in mathematics [7-10]:

$$\mathbf{T} = < \mathbf{B}, \mathbf{F_T}, \mathbf{F_A}>, \tag{1}$$

where **B** is a set of physical or abstract objects that are obtained from other objects of **B** and/or objects of a finite set **A,** which is called a basic set. Therefore, objects of a basic set **A** are called basic elements, and $\mathbf{A} \subset \mathbf{B}$. Objects of **B** are also called elements or constructions, i.e., the terms "object", "element", and "construction" are equivalent in GFT frameworks.

The symbol $\mathbf{F_T}$ in (1) denotes a finite set of finite-place "technological operations", and the finite set $\mathbf{F_A}$ is a set of finite-place "operations of analysis" with objects of **B**.

Any physical object $O_i \in \mathbf{B}$ in GFT is described by the following pair:

$$O_i = <\mathbf{G}_i, \mathbf{M}_i>, \quad (2)$$

where $\mathbf{G}_i = \{g_{i1}, g_{i2},..., g_{in}\}$ is a finite set of physical properties of object O_i and $\mathbf{M}_i = \{g_{ij}=f_{ij}(g_{ik}, g_{il},...,g_{im});...\}$ is a finite set of functionalities of object O_i, i.e., a set of functions (or algorithms), that define the value of one (g_{ij}) or many physical properties of object O_i via other properties (for example, g_{il}) of the same object O_i or via properties ($g_{sk},..., g_{qm}$) of other objects ($O_s,..., O_q$), that interact with the properties of object O_i. In object-oriented programming (OOP), those sets that are similar to $\mathbf{M}_i$ are often called "methods". A set $\mathbf{M}_i$ can contain as many similar functions (functionalities) f_{ij} as necessary for an actual representation of O_i [8, 9].

We will also denote objects $O_i,\ldots O_j \in \mathbf{B}$ using traditional denotations such as $x_i,\ldots, y_j \in \mathbf{B}$, if interpretation (2) is not imporetant in the current context.

The description (2) is similar to the descriptions of "program essences" ("objects") in OOP.

In accordance with (1) and (2), the general formula for any technological ($\mathbf{F_T}$) or analytical ($\mathbf{F_A}$) operations in $\mathbf{T}$ is the following:

$$\mathrm{F_i}(x_1,..., x_n, a_1,..., a_t)\rightarrow<y_1,..., y_m, b_1,..., b_s>, \quad (3)$$

where $\mathrm{F_i} \in \mathbf{F_T} \cup \mathbf{F_A}$; $x_1,..., x_n, y_1,..., y_m, \in \mathbf{B}$, $a_1,..., a_t$ are numerical parameters (for example, the temperatures of mixed solutions) or nonnumeric parameters of operation $\mathrm{F_i}$; and $b_1,..., b_s$ are the numerical or nonnumeric parameters of the operation result (for example, the color of the obtained solution); the indices $n, m, s, t \in \mathrm{N}$ (N is a set of natural numbers) denote different objects or parameters. An arrow ("→") indicates the direction of operation $\mathrm{F_i}$ and should not be considered to connote equality. Angle brackets ("<", ">') denote sequences of objects and parameters obtained as a result of operation $\mathrm{F_i}$.

The physical properties of objects $x_1,..., x_n, y_1,..., y_m$ can be present or absent among parameters $a_1,..., a_t$ and $b_1,..., b_s$ in the form (3).

As an example of a simple analytical operation in the form (3), we can write the formal GFT representation of any technological operation $\mathrm{F_w}$ that measures, for instance, the weight of some object x_i:

$$\mathrm{F_w}(x_i)\rightarrow<x_i, b_w>,$$

where b_w is a (numerical) parameter that represents the weight of object x_i (for example, in grams).

Another example of a simple technological operation $\mathrm{F_s}$ (resembles syntheses) that obtains some object y by junction of two objects x_1 and x_2:

$$\mathrm{F_s}(x_1, x_2)\rightarrow<y>.$$

Because any technological or analytical operations of any technology $\mathbf{T}$ could be represented in the form (3), we can formally describe any technological process or procedure as a sequence of proper operations in the form (3), i.e., as a proper GFT algorithm in $\mathbf{T}$ to obtain or analyze any object (or objects) $x\in\mathbf{B}$ of $\mathbf{T}$. These GFT algorithms also include so-called "conditional and unconditional jumps" (or "branches") to change the local sequences of operations in a similar manner to the way numerical results from numeric calculus are usually used to change sequences of calculations in various computing algorithms (programs). Such "conditional jumps" use numerical (or nonnumerical) results (parameters) of proper analytical operations of the set $\mathbf{F_A}$ to perform (or not perform) the jump.

For comparison with (1), any algebraic system $\mathbf{U}$ in mathematics is also defined as a triplet:

$$\mathbf{U} = <\boldsymbol{A}, \boldsymbol{F}, \boldsymbol{P}>, \quad (4)$$

where $\boldsymbol{A}$ is a base set of elements in $\mathbf{U}$; $\boldsymbol{F}$ is a set of mathematical operations; and $\boldsymbol{P}$ is a set of predicates on $\boldsymbol{A}$ [10].

One can easily prove that if an operation of analysis $\mathrm{F_i}\in \mathbf{F_A}$ has only a finite number of results belonging to the finite subset of rational numbers R, then all of these results can be represented by a finite number of proper predicates.

Indeed, if $\mathrm{F_i} \in \mathbf{F_A}$ and $\mathrm{F_i}(x_j) = \mathrm{m_k}$, $x_j \in \mathbf{B}$, $\mathrm{k}\in\mathrm{N}$; $\mathrm{m_k} \in \mathrm{R}$ (R is a finite subset of rational numbers that represent all possible results of analytical operation $\mathrm{F_i}$), then for every $\mathrm{m_k}$, we can define a predicate $\mathrm{P_{ik}}$ such that $\mathrm{P_{ik}}(x_j) = 1$ if $\mathrm{F_i}(x_j) = \mathrm{m_k}$ and $\mathrm{P_{ik}}(x_j) = 0$ if $\mathrm{F_i}(x_j) \neq \mathrm{m_k}$. Therefore, the operation of analysis, $\mathrm{F_i}$, can be replaced with a finite number of predicates similar to $\mathrm{P_{ik}}$. Thus, the formal representation of any technology $\mathbf{T}$ (1) is similar to formal representations of algebraic systems (4), including a system of partially recursive functions.

B. *Theoretical Model of RRL and FRM*

As is widely known in computing technology, the modern programmable extremely multifunctional (universal) devices are computers. Mathematicians agree with the fact that the so-called *computing power* of a computer is equal to the computing power of a Turing machine (TM). Therefore, the TM is a classical mathematical model of a universal computing device that has a maximum computing power for algebraic systems called computational mathematics or computational technology. According to the Church-Turing thesis, human computational technology is equivalent to a system of partially recursive functions. Thus, it is theoretically possible to construct a universal technological system for various technologies described by Equation (1) in a similar manner as a universal TM for the "technology" of partially recursive functions described by Equation (4).

Such a theoretical structure (denoted by symbols $\mathbf{S_U}$) for a universal technological system, which is based on the main technological concepts of TM in accordance with expressions (1) – (3), is shown in Figure 1.

A shift register and pointer (denoted by digit "1" in Figure 1), together with a one-dimensional array of storage cells, perform the role of the one-dimensional tape and shifting mechanism of TMs.

An operational unit containing technological cells $\mathrm{F_1},..., \mathrm{F_m}$, performs technological and analytical operations $\mathrm{F_1},..., \mathrm{F_m}\in \mathbf{F_T}\cup \mathbf{F_A}$. This unit has an input demultiplexer that distributes objects extracted from indicated (by pointer) storage cells to the proper technological cells $\mathrm{F_k}$ in accordance with performed technological or analytical oper-

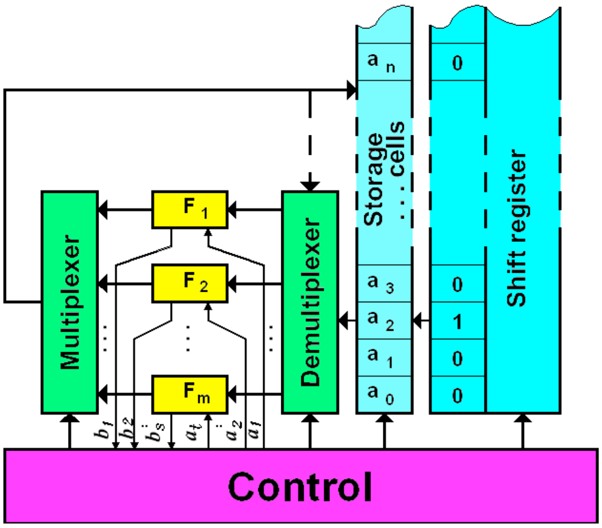

Figure 1. $\mathbf{S}_U$ structure of the theoretical model for the extremely multifunctional (universal) programmable synthesizer-analyzer of various objects for any formal technology **T**

ations. In Figure 1, an output multiplexer directs output objects from ethe outputs of technological cells to the proper storage cells according to the current position of the pointer. All technological and analytical cells are controlled by the control unit in accordance with Equation (3) and a table of rules similar to the table of rules in TMs. Moreover, the control unit can use the parameters (results) of analytical operations to change the sequence of operations; as in the case of the scanned symbol in the current cell of the TM's tape, which can also change the current state of this machine [8, 11].

Thus, the operational and control units perform technological operations in a similar manner to the control, reading and typing operations on the tape of a TM in accordance with its table of rules.

To guarantee the most efficient use of $\mathbf{S}_U$, a "complete" technology should be implemented in $\mathbf{S}_U$; a "complete" technology **T** allows us to i) generate an infinite number of new constructions in **T** and ii) restore the structure and synthesis of any construction that belongs to the set **B** of **T**. There are many ways to achieve completeness of various technologies in GFT [8, 12, 14].

The following theorems can be easily validated for every complete technology:

A. There are many complete technologies **T** that allow us to simulate a TM in $\mathbf{S}_U$.

B. For any object $O_i \in \mathbf{B}$ of any complete technology **T**, there is an algorithm realizable in $\mathbf{S}_U$ for **T** that automatically restores the structure of object O_i, and its synthesis in **T** [8, 13, 14].

C. For any complete technology **T**, there is an algorithm that enumerates syntheses of various objects $O_i \in \mathbf{B}$ in **T** by a regular method so that every such synthesis can be mapped onto the one unique natural number in the sequence of natural numbers. Therefore, all objects in **T** obtained by such a set of syntheses can be restored by the proper natural number. (This theorem is also called the "Theorem of Acquired Knowledge Effectiveness") [8, 13].

The proof of theorem A is based on the idea of interpreting different symbols of any finite alphabet used in TMs as different objects of finite set **A** in **T** implemented in a proper $\mathbf{S}_U$ [7, 8, 12].

The proof of theorem C uses the simple fact that algorithms, which describe various object syntheses, can be constructed in a regular manner as increasing finite sequences of operations belonging to a finite set $\mathbf{F}_T \cup \mathbf{F}_A$.

There are many ways to prove theorem B for different technologies [8, 12]. One of the most general approaches is the use of theorem C to find a proper synthesis to obtain the exact copy of the necessary object O_i. Therefore, theorem B is true even if the object O_i is unknown, i.e., if we have no information about its synthesis and structure aside from the fact that $O_i \in \mathbf{B}$ for **T**. That is, $\mathbf{S}_U$ allows us to analyze and synthesize any object $O_i \in \mathbf{B}$. Moreover, theorem C asserts that we can automatically investigate the possibilities of technology **T** and, when necessary, we can automatically restore the proper or required object or synthesis, which is exactly what is necessary for a flexible RRL, FRM or FREF.

Statement A also confirms the fact that $\mathbf{S}_U$ is a universal programmable system.

III. PRACTICAL ASPECTS

For practical needs, we must improve some components of the $\mathbf{S}_U$ structure; furthermore, computers have an improved structure that differs from that of a TM. Thus, we must change some blocks for the practical structure of the RRL, FRM or FREF compared to their theoretical structure shown in Figure 1.

First, we should replace the complex storage cell units from Figure 1 with a more suitable storage system that uses addressable storage cells (similar to addressable cells in computer memory) to store various objects of the set **B**.

Second, technological algorithms often use "continuous processes" to synthesize objects or substances. For that purpose, a "technological output" of one technological cell is typically immediately connected to the "technological input" of another technological cell. To make such an immediate connection in $\mathbf{S}_U$, we should include a "connecting matrix" that allows us to set (by programming) necessary connections in the array of technological cells according to $\mathbf{F}_T \cup \mathbf{F}_A$.

Third, we should use a common device known as a microcontroller (with program memory) to control $\mathbf{S}_U$.

The structure of $\mathbf{S}_U$ for practical needs in accordance with these features is shown in Figure 2.

The microcontroller in Figure 2 performs interpretations of commands in accordance with Equation (3) for operations in $\mathbf{F}_T \cup \mathbf{F}_A$ and its parameters (see section I-*B*). This unit also performs conditional and unconditional jump commands to change the subsequent command sequences (and corresponding technological operation sequences) according to the results for the analytical operation parameters. That is, any technological algorithms are constructed as the proper sequences of commands on principals that are similar to the principals of computer program constructions for computational algorithms [8, 12].

The programmable connections and interconnections in Figure 2 are used to set the proper transportation systems for the technological operations or performed processes.

Theorems A, B, and C from section II-*B* can be proven for the practical structure of $\mathbf{S}_U$ shown in Figure 2. Moreover, it is proven that so called "specialized technological systems" (i.e., "continuous systems" that allows us to obtain certain types of objects or substances "continuously") have a productivity that is equal to the productivity of the universal technological system $\mathbf{S_U}$ if they both possess the same sets of technological and analytical cells [8, 12].

Alternately, if we want to set various electrical parameters for different technological operations and to measure (electrically) various parameters (results) of analytical operations according to Equation (3), then it is useful to use a universal analog-digital programmable system, such as $\mathbf{S_U}$, for those purposes. Therefore, a RRL, FRM or (especially) FREF as an entire system should have two subsystems that are similar to $\mathbf{S_U}$. The first subsystem allows us to set and measure various electrical parameters. This subsystem implements a proper technology $\mathbf{T_{MS}}$ of mixed signal processing. The second subsystem allows us to conduct various experiments in the investigated technology $\mathbf{T_I}$.

Figure 3 presents a simplified structure of such an improved RRL, FRM or FREF.

The technological cores in Figure 3 represent arrays of proper technological and storage cells together with necessary programmable transportation systems for $\mathbf{T_{MS}}$ and $\mathbf{T_I}$. If the investigated technology $\mathbf{T_I}$ is also $\mathbf{T_{MS}}$, then both technologies can be implemented in the same technological core [13].

The left subsystem (green box in Figure 3) allows us to temporarily construct the necessary generating and measuring devices. Such devices could be called "virtual" devices [13, 15] because, if we generate any concepts to make these devices more suitable, stable, or precise, then we can reconfigure existing virtual devices into such improved devices. Of course, if all parameters in operation (3) are fixed and they have fixed technical implementations, then it is not necessary to vary them, and the left subsystem in Figure 3 can be implemented as a fixed subsystem of the technological cells of $\mathbf{T_I}$.

The most suitable technological areas for RRLs (FRMs or FREFs) that have structures similar to those shown in Figures 2 and 3 are as follows:

1. Technology $\mathbf{T_{DD}}$ for digital data processing.
2. Technology $\mathbf{T_{AS}}$ for analog signal processing.
3. Technology $\mathbf{T_{MS}}$ for mixed (analog and digital) signal processing.
4. Various chemical technologies $\mathbf{T_{ChS}}$ for various chemical solutions.
5. Similar chemical technologies $\mathbf{T_{GSChS}}$ for gaseous substances and chemical solutions.
6. Similar chemical technologies $\mathbf{T_{SSGSChS}}$ for granular solid substances, gaseous substances, and chemical solutions.
7. Biochemical $\mathbf{T_{BCh}}$ and micro-biochemical $\mathbf{T_{mBCh}}$ technologies for the above types of substances.
8. Analytical multifunctional "Programmable Labs-on-a-Chip" (PLOCs) for various technologies [16].

IV. PRACTICAL IMPLEMENTATIONS

A potential structure for a technological core for simple complete investigated technology $\mathbf{T_{ChS}}$ is shown in greater detail in Figure 4.

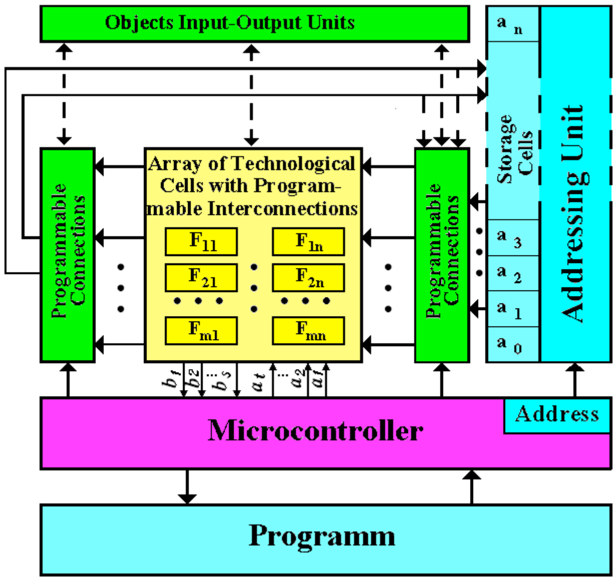

Figure 2. Block diagram of $\mathbf{S_U}$ for practical purposes

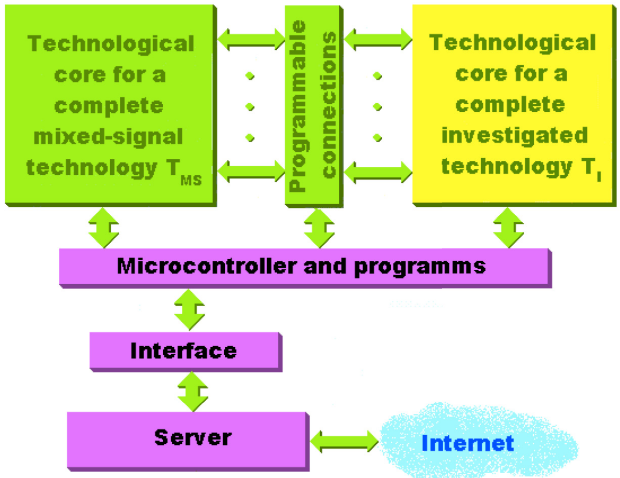

Figure 3. Simplified structure of a modernized RRL, FRM or FREF for any investigated technology $\mathbf{T_I}$

As mentioned above, technology $\mathbf{T_{ChS}}$ is intended for simple chemical operations with (for instance) various water solutions. Therefore, set $\mathbf{F_T}$ of $\mathbf{T_{ChS}}$ has 2 technological operations: for solution batching, F_b, and for solution mixing, F_m, $\mathbf{F_T}$ = <$F_b(x, a)$, $F_m(x_1, x_2)$>, where $F_b(x, a) \rightarrow <(x\text{-}ax), ax>$, and a is a batching parameter that defines a volume ax of water solution obtained in the batching operation; $(x\text{-}ax)$ is a rest of water solution x after separation of volume ax. $F_m(x_1, x_2) \rightarrow <y>$, where y is a mixture of water solutions x_1 and x_2. Operation F_b is implemented in technological cell 1 with the help of a proper electronic device 2 connected to capacitive sensors. This device determines the exact volume ax of the water solutions x that is contained in the technological cell 1 (implemented, for instance, as a narrow glass tube with evenly distributed capacitive sensors).

Operation F_m is implemented in technological cell 3, which simultaneously achieves the operation of fluorescent analysis $P_f \in \mathbf{F_A}$.

To complete technology $\mathbf{T_{ChS}}$, one should guarantee two possibilities: i) obtaining any water solutions ("objects") with various possible concentrations of necessary

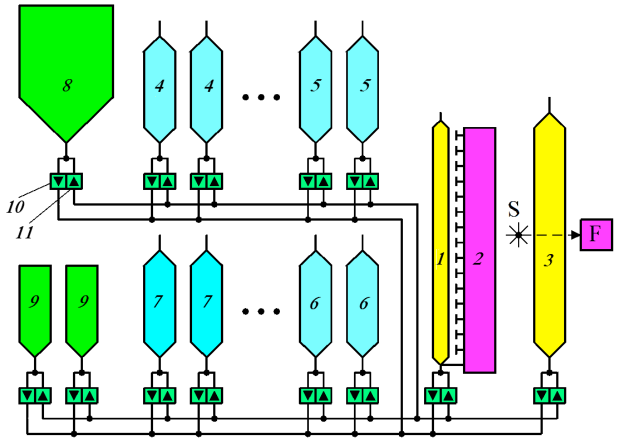

Figure 4. A simple structure of a technological core for an investigated technology $\mathbf{T}_{\mathbf{ChS}}$

ingredients and ii) recreating any possible water solutions with the same ingredients and with the same concentrations [8, 12]. We shall define a finite set of necessary water-soluble ingredients (in the form of concentrated or "pregnant" solutions) as a part of finite "basic" set $\mathbf{A}^{/} = \{a_1, a_2,\ldots, a_n\}$. Set $\mathbf{A}^{/}$, together with a dissolvent a_0 (water), forms a complete "basic" set $\mathbf{A}$ of "basic elements" (the dissolvent and pregnant solutions): $\mathbf{A} = \mathbf{A}^{/} \cup \{a_0\}=\{a_0, a_1, a_2,\ldots, a_n\}$. It is clear that by using the technological operations $\mathbf{F_T}$ = <$F_b(x, a)$, $F_m(x_1, x_2)$>, we can obtain any possible solution with any possible concentration [8, 12].

To define the unknown ingredients of any unknown water solution and to measure their concentrations (i.e., to recreate the syntheses of unknown "objects" in $\mathbf{T}_{\mathbf{ChS}}$), we need the following: a) an appropriate finite set $\mathbf{B}_{\mathbf{F1}}$ of so-called (water-soluble) "indicators" and b) an appropriate finite set $\mathbf{B}_{\mathbf{F2}}$ of so-called (water-soluble) "neutralizers". Sets $\mathbf{B}_{\mathbf{F1}}$ and $\mathbf{B}_{\mathbf{F2}}$ perform specific functions in technology $\mathbf{T}_{\mathbf{ChS}}$ and should not be included in the set $\mathbf{B}$ in formula (1). Elements (solutions) of set $\mathbf{B}_{\mathbf{F1}}$ are used to define each ingredient of set $\mathbf{A}^{/}$ similarly to the way the presence (or absence) of certain molecules is defined in qualitative fluorescence analysis. If such molecules are present in the given solution, they will generate visible light (detected by sensor F in Figure 4) under ultra-violet rays (emitted by ultra-violet source S in Figure 4). Therefore, for each ingredient of set $\mathbf{A}^{/}$, we shall have a corresponding indicator in set $\mathbf{B}_{\mathbf{F1}}=\{b_1, b_2,\ldots, b_n\}$.

The elements (solutions) of set $\mathbf{B}_{\mathbf{F2}}$ play another role. They "neutralize" the emission of visible light in fluorescent analysis by junction with ("neutralizing") molecules of set $\mathbf{B}_{\mathbf{F1}}$. If all such molecules in solution b_j ($b_j \in \mathbf{B}$) are "neutralized", then light is absent and solution b_j has no active molecules of $\mathbf{B}_{\mathbf{F1}}$. If for each type of ingredient in $\mathbf{B}_{\mathbf{F1}}$ we use a proper "neutralizer" $b'_i \in \mathbf{B}_{\mathbf{F2}}$, then $\mathbf{B}_{\mathbf{F2}}=\{b'_1, b'_2,\ldots, b'_n\}$.

The above process is very similar to a titration. Indeed, it should also use solutions of "neutralizers" with fixed concentrations to define the quantity of the ingredients in $\mathbf{B}_{\mathbf{F1}}$ in a given unknown object (unknown solution $b_j \in \mathbf{B}$).

We cannot say objects of sets $\mathbf{B}_{\mathbf{F1}}$ and $\mathbf{B}_{\mathbf{F2}}$ are objects of set $\mathbf{B}$ for two reasons: 1) they might not belong to $\mathbf{B}$ and 2) they expand the technological possibilities of operations $\mathbf{F_T} \cup \mathbf{F_A}$.

Storage cells (for instance, glass flasks) denoted in Figure 4 by the digit "4" are intended for solutions of set $\mathbf{A}$. Storage cells denoted by the digit "5" are intended for solutions in set $\mathbf{B}_{\mathbf{F1}}$, and storage cells denoted by digit "6" are intended for set $\mathbf{B}_{\mathbf{F2}}$. Storage cells denoted by digit "7" are empty. They are intended for the realizations of various technological processes in technology $\mathbf{T}_{\mathbf{ChS}}$.

All transport communications among technological cells ("1", "3"), input-output cells ("8" and "9"), and storage cells ("4", "5", "6", "7") could be implemented using glass pipes (for instance). Every cell is connected to the pipe by input ("11") and output ("10") valves with proper pumps.

The more detailed structure of a technological core unit for the technology $\mathbf{T}_{\mathbf{ChS}}$, including a longer explanation, is described in [8, 12].

The same principles could be used for a technological core design in the above cases 5 - 7.

The structure shown in Figure 4 has a strong mathematically approved statement about its technological universality (or maximum flexibility). To solve the same problem for a so-called "programmable laboratory on a chip" (PLoC) called "Aquacore", authors have tried to use a model of TM [16]. It is well known that TM is intended to solve various tasks in calculation technology that differ substantially from simple chemical technology $\mathbf{T}_{\mathbf{ChS}}$ [8, 12].

In comparison with the above cases 1 – 7 for certain simple different technologies, other technological areas require specific transportation and positioning units to implement structures of multifunctional RRLs or flexible programmable remote manufacturing plants (FRMs), such as $\mathbf{S}_{\mathbf{U}}$ [8, 11, 12]. However, the entire structure of a proper RRL or FRM is easier than in the case when we use robots or complex manipulators for object transportations and positioning, as is performed in modern flexible manufacturing systems (FMSs). Such specific technological areas for RRLs and FRMs are as follows:

9. Machine-building technology $\mathbf{T}_{\mathbf{MB}}$ [8, 12].
10. Nano-technologies for nano-objects $\mathbf{T}_{\mathbf{NT}}$ [11, 12].

Cases 1-10 cover the main technologies used by human civilization. For every such technology, it is possible to automatically create various experiments in accordance with theorems B and C from section II-*B*. These experiments may include the following tasks:

a) Automatic searching for new objects having specific properties in $\mathbf{T}$;

b) Automatic optimization of existing technological algorithms to obtain objects that have specific properties;

c) Automatic searching for algorithms for syntheses or/and analyses of objects that have specific properties in $\mathbf{T}$;

d) Other tasks.

At the moment, such a remote laboratory has been developed for technology $\mathbf{T}_{\mathbf{MS}}$ (see case 3 above). Its structural realization is shown in Figure 3.

This RRL uses programmable mixed signal "system-on-a-chip" PSoC-1 (CY8C27443) by the Cypress Semiconductor Corporation as a core for the left measuring subsystem $\mathbf{T}_{\mathbf{MS}}$ (green box) and for the right investigated technology subsystem $\mathbf{T}_{\mathbf{I}}$ = $\mathbf{T}_{\mathbf{MS}}$ (yellow box), with the structure shown in Figure 3 [13]. A PCB for this RRL is shown in Figure 5.

The RRL is used to conduct real laboratory experiments in the following courses:

1. "Analog Interfaces of Computers".
2. "Microprocessor Systems".

Figure 6 provides an example of graphical data obtained from the RRL in the course "Analog Interfaces of Computers". The conducted experiment uses non-typical connections among analog and digital blocks. Figure 6 shows digital (top) and analog (bottom) signals obtained for the simultaneous outputs of such blocks.

The internal structure of PSoC-1 is close to the theoretically based architecture of $\mathbf{S}_U$ shown in Figure 2 and the approved RRL architecture shown in Figure 3. The standard set of virtual measuring and generating devices (four-channel digital analyzer and two-channel digital oscilloscope) that are necessary for experimentation in the above courses uses only a small part of the configurable digital blocks (DCBs) and analog blocks (ACBs) of PSoC-1. In total, such virtual devices use only 25% of the DCBs and 33% of the ACBs. Thus, the majority of the DCB and ACB arrays are free and can be utilized by users for their real experiments.

Users can explore the remaining DCBs and ACBs. Moreover, users can replace the existing set of virtual devices with more suitable devices, and they can change the collected data and its graphical interpretations. Such modifications do not require an alteration of the RRL hardware. Instead, they require only the appropriate reprogramming of RRL and its computer software. Thus, in comparison with the "classical" approach [17], such RRLs have greater flexibility.

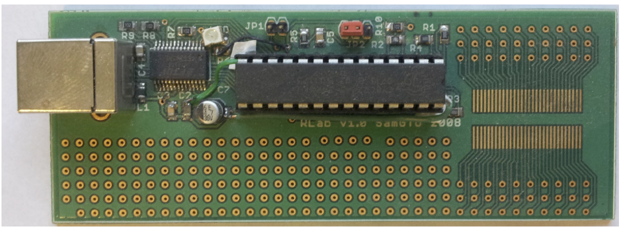

Figure 5. The RRL PCB for $\mathbf{T}_{MS}$ designed in accordance with the structures shown in Figures 2 and 3

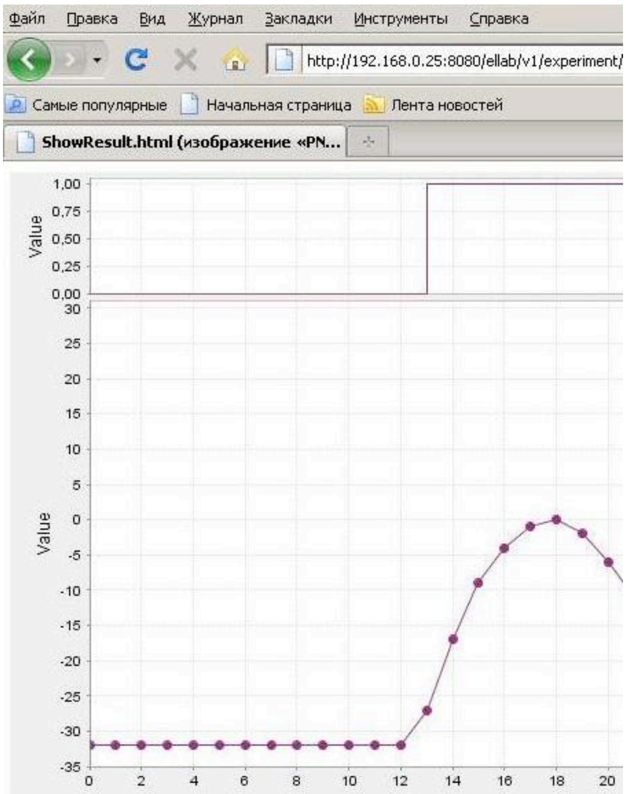

Figure 6. Example of real digital and analog signals obtained in experiments for the technology $\mathbf{T}_I = \mathbf{T}_{MS}$

V. CONCLUSIONS

Interpretations of real world objects and operations with them in the metamathematical forms (1) and (2) allow us to use different formalisms that are equivalent to the formalisms used in very important structures in mathematics called algebraic systems (4).

First, these formalisms allow us to construct various formal algorithms to investigate the theoretical possibilities of such formal technological processes and systems. For example, we can show that there are many complete technologies in GFT instead of one complete technology in computational mathematics called partially recursive functions.

Second, for every complete technology, we can construct a formal model based on principles similar to the functional organization of TMs. That is, we can construct a universal fully programmable technological device that can automatically perform any correct technological algorithm to obtain any possible object in the given complete technology. This result is very important and very useful for designing RRLs, FRMs and FREFs.

Third, we can show that many research works have regular structures and can therefore be performed automatically by such universal programmable technological devises. These possibilities are highly important for RRLs and especially for FREFs.

Fourth, the possibility of obtaining various real ("physical") objects at a distance will allow us to launch an "Internet of Things" that is not based only on 3D-printer technologies [18]. Various FRMs can make very complex objects by using more suitable technological operations than the "layer by layer" principles used in most 3D printers.

Regarding the "lack of modular approach in design", the suggested formalisms also can help with that aspect. Indeed, to achieve universality and remote programmability for various technological systems, we should define a set of technological operations $\mathbf{F}_T \cup \mathbf{F}_A$ (for any complete technology $\mathbf{T}$). This set is finite, and therefore, we should have only a finite number of proper technological cells (technological modules), which could be standardized. To increase the possibilities of technology $\mathbf{T}$, it is usually necessary to add a few new technological cells (modules) to the existing technological ones [8, 12]. This addition could also be created by standard "technological interfaces" that are similar to the standard system interfaces used in computers. Therefore, we could start with a very cheap and simple universal distantly programmable device, and then, step by step, we can expand their technological possibilities by regular and standard means.

However, as a standard part of the control unit, we could use a flexible and reconfigurable architecture of $\mathbf{S}_U$ in the testing and measuring subsystems shown in Figure 3 (green box). It allows us to vary testing and measuring devices and their methods (algorithms) over a wide range.

The described theoretical approach to the designe of RRLs, FRMs and FREFs has rigorous mathematical foundations. Therefore, in comparison with traditional approaches, it guarantees the maximal flexibility, full remote programmability, and complete remote control of designed RRLs, FRMs and FREFs.

ACKNOWLEDGMENTS

The author wishes to thank Professor Sergey Orlov, engineer Vladimir Tolchev, and Dmitry Lavrov for their assistance.

REFERENCES

[1] Abul K. M. Azad. Internet Accessible Remote Experimentation: Setting the Right Course of Action. // "*Intern. Journal of Online Engineering"* (iJOE), **6**(3), pp.4-12, 2010

[1] Naef O. Real laboratory, virtual laboratory or remote laboratory: what is the most efficient way? // *"Intern. Journal of Online Engineering"* (iJOE), **6**(3), pp.1-12., 2006.

[2] Aliane, N., Pastor, R., and Mariscal, G. Limitations of Remote Laboratories in Control Engineering Education. // *"Intern. Journal of Online Engineering"* (iJOE), **6**(1), pp. 31-33, 2010.

[3] Salaheddin Odeh, Shatha Abu Shanab, Mahasen Anabtawi, Rami Hodrob. A Remote Engineering LAB Based on Augmented Reality for Teaching Electronics..// iJOE – Volume 9, Special Issue 5: "EDUCON2013", June 2013, pp.61-67

[4] An XML Modular Approach in the Building of Remote Labs By Students: A Way to Improve Learning. / R. Pastor-Vargas, D. Sánchez, S. Ros, R. Hernández, A. Caminero, L. Tobarra, A. Robles-Gomez, M. Castro, E. San-Cristóbal, G. Diaz, M. Tawfik. // iJOE – Volume 9, Special Issue 5: "EDUCON2013", June 2013, pp.5-12

[5] Dag A. H. Samuelsen and Olaf H. Graven Low Cost Implementation of Remote Lab with Large Number of Configurations for a BJT Amplifier. // iJOE – Volume 7, Special Issue 1: "EDUCON2011", September 2011, pp.31-36

[6] Krylov, S.M.. Formal technology and universal systems. //"*Cybernetics and Systems Analysis*", Springer, New York, ISSN 1060-0396 (Print) 1573-8337 (Online), Part I, **22**(4), pp.512-518; Part II, **22**(5), pp.567-572, 1986.

[7] Krylov, S.M. Ontology of Meta-Science. Axioms, Technologies, Algorithms, Evolution (in Russ.). Saarbrücken: LAP LAMBERT Academic Publ., 2012.-408 pp.

[8] Krylov, S.M. Prospects of Metamathematical Structures in Science (in Russ.). // "*Vestn. Samar. Gos. Tekhn. Univ. Ser. Fiz.-Mat. Nauki*", №1 (26), pp. 233–242, 2012. Http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=vsgtu&paperid=1203&option_lang=eng

[9] Malcev, A.I. Algebraic Systems. Berlin: Springer-Verlag, 1973. xii+317 pp.

[2] Krylov, S.M. Universal Programmable Completely Automated Factories-on-a-Chip. Proceedings of the 9th International Conference on the Commercialization of Micro and Nano Systems COMS2004. Aug.29 - Sept.2, 2004, Edmonton, Alberta, Canada. - MANCEF, Washington, 2004, pp.269-273.

[3] Krylov, S.M. Metamathematical Foundations of Future Science (in Russ.). Samara: Samara State Technical University, 2014. 223 pp. Http://vt.samgtu.ru/images/stories/metamath.pdf. ISBN 978-5-7964-1686-0

[10] Krylov, S.M., Tolchev, V.N. Multifunctional Distance Labs for Real Experiments (in Russ.). // "*Vestn. Samar. Gos. Tekhn. Univ. Ser. Technich.. Nauki*", №1(29), pp. 85–91, 2011.

[11] Krylov, S.M. Formal Technology and Cognitive Processes. // "*Intern. Journal of General Systems",* **24**(3), pp.233-243.

[12] NI Educational Laboratory Virtual Instrumentation Suite (NI ELVIS), 31/10/2010, http://www.ni.com/academic/ni_elvis/.

[13] Amin A.M., Thottethodi M., Vijaykumar T.N., Wereley S., Stephen C. Jacobson S.J. Aquacore: a programmable architecture for microfluidics. // ACM SIGARCH Computer Architecture News archive, Vol. 35 , № 2 (May 2007), pp. 254-265.

[14] Lab on the Web: Running Real Electronics Experiments via the Internet. Edited by Tor A. Fjeldly and Michael S. Shur. October 2003, Wiley-IEEE Press. 256 pages. ISBN 0-471-41375-5 http://dx.doi.org/10.1002/0471727709

[15] 3D Printing. 10th November 2013, http://www.explainingthefuture.com/3dprinting.html

AUTHOR

Sergey M. Krylov is with the Computer Science Department of Samara State Technical University, Samara, Russia (e-mail: s_m_krylov@mail.ru).

Submitted, 30 March 2014. Published as resubmitted by the authors 13 September 2014.