Organization-Oriented Chemical Programming

Naoki Matsumaru,Peter Kreyssig, Peter Dittrich (2011)

Bio Systems Analysis Group, Department of Mathematics and Computer Science,
Friedrich Schiller University Jena, 07743 Jena, Germany,
{peter.kreyssig|peter.dittrich}@uni-jena.de

Published as:

P. Dittrich, P. Kreyssig. Organization-Oriented Chemical Programming. In: C. Muller-Schloer, H. Schmeck, T. Ungerer (Eds.), Organic Computing A Paradigm Shift for Complex Systems, Autonomic Systems, Volume 1, Part 1, 67-78, Springer, Basel , 2011

doi: 10.1007/978-3-0348-0130-0_13

Organisation-Oriented Chemical Programming

Naoki Matsumaru[1], Peter Kreyssig[2] and Peter Dittrich[2]

[1] Gifu University Hospital, Gifu University, Gifu City 501-1193, Japan,
`n_matsu@gifu-u.ac.jp`
[2] Bio Systems Analysis Group, Department of Mathematics and Computer Science, Friedrich Schiller University Jena, 07743 Jena, Germany,
`{peter.kreyssig|peter.dittrich}@uni-jena.de`

Summary. *Chemical information processing* possesses a variety of valuable properties, such as robustness, concurrency, fault-tolerance and evolvability. However, it is difficult to predict and program a chemical system because the computation emerges as a global phenomenon from microscopic reactions. For programming chemical systems a theoretical method to cope with that *emergent behaviour* is desirable. Here we will review *design principles* for chemical programs. We focus on programs that should compute a qualitative and not a quantitative result. The design principles are based on chemical organisation theory, which defines a chemical organisation as a closed and self-maintaining set of molecular species. The fundamental assumption of so called *organisation-oriented programming* is that computation should be understood as a movement between chemical organisations. In this case we expect that the resulting system is more robust and fine-tuning of the kinetic laws will be less important. As examples for the usage of this design method we show a logic gate and a solution to the maximal independent set problem implemented as artificial chemistries.

1 Introduction

By employing a large number of simple components interacting with each other in an orchestrated way, biological systems invented a variety of information processing mechanisms, which are robust, self-organising, adaptable, decentralised, asynchronous, fault-tolerant and evolvable. This principle of biological information processing has been exploited to cope with the fast-growing complexity of technical information processing systems [28, 35, 36]. Since all known life forms process information using chemical processes [20], the chemical reaction metaphor has been proposed as a source of inspiration [2, 10]. Using chemical reactions for formal computations has initially been suggested by Banâtre and Métayer [2]. In their GAMMA system [3], a chemical reaction is defined as a rewriting operation on a multiset, mimicking a well-stirred reaction vessel. In order to capture the spatial context of chemical systems, chemical rewriting systems have been extended to the chemical abstract machine (CHAM) [6]; P-Systems [31] stressing the importance of membranes; and MGS [14] allowing arbitrary topologies [4]. However there is still a lack of knowledge of how to actually design chemical computing algorithms.

The difficulty of engaging the chemical reaction process for computing is that the solution appears as an emergent global behaviour based on the manifold local interactions [5]. An emergent behaviour of biological systems is simulated by combining simple biochemical signalling pathways [7]. As recently demonstrated by Tsuda et al. [34] with a slime mould in the genus of Physarum, the discrepancy between local and global behaviour may also be problematic in a practical case. A satisfying theory of emergence is lacking [27].

This paper reviews the contributions towards the establishment of a theoretical analysis of the emergent behaviour in chemical computing. We suggest chemical organisation theory [33, 11, 29] as a tool helping to construct (program) and analyse (describe and understand) chemical computing systems.

Inspired by Fontana and Buss [13], we define a (chemical) organisation as a set of molecular species that is (algebraically) closed and (stoichiometrically) self-maintaining [11]. It is important to note that when we talk about organisations, we abstract details like concentration levels or the spatial distribution of a chemical species. On this relatively high level of abstraction, a system state is characterised by the molecular species present only and we can describe the dynamics of a system more qualitatively, namely, as a movement between sets of species instead of a movement in a more complex state space [33].

In Sec. 2 we describe the concepts from chemical organisation theory needed here. As an example of chemical computing, an XOR logic gate is implemented using an (artificial) chemical reaction network with the help of chemical organisation theory in Sec. 3.1. A second example is given by a chemical algorithm for the solution of the maximal independent set problem in Sec. 3.2. In Sec. 4 we review three different ways of designing chemical programs guided by chemical organisation theory. Finally, in Sec. 5 we discuss the potential of the theory as a theoretical base for emergence analysis.

2 Chemical Reaction Networks, Chemical Organisation Theory and Movement between Organisations

By a *chemical reaction network* we mean a pair $\langle \mathcal{M}, \mathcal{R} \rangle$ where $\mathcal{M}$ is a set and $\mathcal{R}$ is a subset of $\mathcal{P}_{mult}(\mathcal{M}) \times \mathcal{P}_{mult}(\mathcal{M})$. Here $\mathcal{P}_{mult}(\mathcal{M})$ is the set of multisets over $\mathcal{M}$. We call the elements of $\mathcal{M}$ *molecular species* and the elements of $\mathcal{R}$ *reactions* resembling the notions of chemistry. Reaction networks can also be described by Petri nets or multiset rewriting. We summarise the needed concepts of *chemical organisation theory* [11].

We fix a reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$. By *applying* a reaction $(l, r) \in \mathcal{R}$ to a multiset over $\mathcal{M}$ we mean replacing the subset l by the subset r. We assume that the multiset is always large enough.

For $(l, r) \in \mathcal{R}$ we also write $l \longrightarrow r$ or

$$\sum_{m \in \mathcal{M}} l_m m \longrightarrow \sum_{m \in \mathcal{M}} r_m m$$

where we denote by $l_m, r_m \in \mathbb{N}_0$ the multiplicity of m in l, r respectively. This resembles notation from chemistry. Furthermore the *support* and the *product* of (l, r) are

$$\operatorname{supp}(l, r) := \{m \in \mathcal{M} \mid l_m > 0\}, \;\; \operatorname{prod}(l, r) := \{m \in \mathcal{M} \mid r_m > 0\}.$$

Let $\mathcal{A}$ be a subset of $\mathcal{M}$. We define $\mathcal{R}_A$ by setting

$$\mathcal{R}_{\mathcal{A}} := \{(l, r) \in R \mid \operatorname{supp}(l, r) \subseteq \mathcal{A}\}.$$

The *stoichiometric matrix* $\mathbf{M}_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{R}_{\mathcal{A}}|}$ for $\mathcal{A}$ is given by

$$(\mathbf{M}_{\mathcal{A}})_{a,(l,r)} = r_a - l_a, \quad a \in \mathcal{A}, (l, r) \in \mathcal{R}_{\mathcal{A}}.$$

Definition 1. *A subset $\mathcal{A}$ of $\mathcal{M}$ is* closed *if for all reactions $(l, r) \in \mathcal{R}_A$ we have* $\operatorname{prod}(l, r) \subseteq \mathcal{A}$*, i.e. if $(\mathcal{A}, \mathcal{R}_{\mathcal{A}})$ is a reaction network.*

$\mathcal{A}$ being closed means that by applying reactions from $\mathcal{R}_{\mathcal{A}}$ to multisets over $\mathcal{A}$ we do not get molecules outside $\mathcal{A}$.

Definition 2. *A subset $\mathcal{A}$ of $\mathcal{M}$ is* semi-self-maintaining *if for every $a \in \mathcal{A}$ and $(l, r) \in \mathcal{R}_{\mathcal{A}}$ with $l_a - r_a > 0$ there is an $(l', r') \in \mathcal{R}_{\mathcal{A}}$ with $r'_a - l'_a > 0$.*

$\mathcal{A}$ being semi-self-maintaining means that if a reaction application destroys a species, there is also a reaction producing this species.

Definition 3. *A subset $\mathcal{A}$ of $\mathcal{M}$ is* self-maintaining *if there is a vector $v \in \mathbb{R}^{|\mathcal{R}_{\mathcal{A}}|}$ with strictly positive entries such that $\mathbf{M}_{\mathcal{A}} v \in \mathbb{R}^{|\mathcal{A}|}$ has only non-negative entries.*

$\mathcal{A}$ being self-maintaining means that applying reactions from $\mathcal{R}_{\mathcal{A}}$ at certain rates to a multiset over $\mathcal{M}$ does not reduce the number of molecules of any species of $\mathcal{A}$.

Definition 4. *A subset of $\mathcal{M}$ is a* chemical (semi-)organisation *if it is closed and (semi-)self-maintaining.*

The organisational analysis decomposes the reaction network given into overlapping sub-networks or organisations. We visualise the set of all organisations by a Hasse diagram in which the ordering is given by the subset relation, e.g., Fig. 1(A).

What we mean by *movement between chemical organisations* is described in the following. For the dynamics of the reaction networks, we assume the standard *law of mass action*, remarking that other kinetics can be used equally well. After assigning constant reaction rates to each reaction, we can derive a system of ordinary differential equations to describe the change in concentration of the species. The solution of the system of equations yields a trajectory in the space of species concentrations. For every point on this trajectory we can find a set of present species, i.e., the species with non-zero concentration, called the *abstraction* of this point. Following the trajectory we get a sequence of abstractions, i.e., subsets of the set of species. This also yields a "movement" between chemical organisations. The choice of the reaction rate constants only influences the quantitative and not the qualitative aspects of the movement.

The fundamental assumption of the so called *organisation-oriented programming* is that computation should be understood as such a movement between chemical organisations. The underlying hypothesis is that when a computation can be explained like this, it is more robust and fine-tuning of the kinetics is less important, since the organisations only depend on stoichiometric information. For example, classical DNA Computing [1] can also be understood in terms of chemical organisation theory. For each solution there is at least one organisation and the experimental steps assure that the system will end up in such an organisation.

3 Examples

3.1 A Chemical XOR – Reaction Network, Organisations and Dynamics

To demonstrate how chemical organisation theory can be used for chemical computing, an (artificial) chemical reaction network is designed to implement an XOR logic gate. We follow the general recipe described by Matsumaru and Dittrich [23].

The XOR logic gate is defined as a set of three boolean variables $\{\mathsf{c}, \mathsf{a}, \mathsf{b}\}$ and a set of one boolean function $\{F_{\mathsf{c}}\}$ where the function is $\mathsf{c} = F_{\mathsf{c}}(\mathsf{a}, \mathsf{b}) := \mathsf{a} \oplus \mathsf{b}$. An algebraic chemistry $\langle \mathcal{M}_{\text{XOR}}, \mathcal{R}_{\text{XOR}} \rangle$ is generated to implement the logic gate. Since there are $N = 3$ boolean variables, the set of molecular species consists of six molecular species:

$$\mathcal{M}_{\text{XOR}} = \{a, A, b, B, c, C\} \tag{1}$$

where the lower and upper case version of the variable name are assigned to the boolean variable of that name. For example, molecular species a represents boolean variable $\mathsf{a} = 0$ and A stands for $\mathsf{a} = 1$. The set of reaction rules $\mathcal{R}_{\text{XOR}}$ is decomposed into two parts:

$$\mathcal{R}_{\text{XOR}} = \mathcal{L}_{\text{XOR}} \cup \mathcal{D}_{\text{XOR}} \tag{2}$$

where $\mathcal{L}_{\text{XOR}}$ is a set of reactions for the logical operation and $\mathcal{D}_{\text{XOR}}$ is a set of destructive reactions:

$$\mathcal{L}_{\text{XOR}} = \mathcal{L}^{\mathsf{c}}_{\text{XOR}} = \{a + b \to c,\ a + B \to C,\ A + b \to C,\ A + B \to c\}, \tag{3}$$

$$\mathcal{D}_{\text{XOR}} = \{a + A \to \emptyset, b + B \to \emptyset, c + C \to \emptyset\}. \tag{4}$$

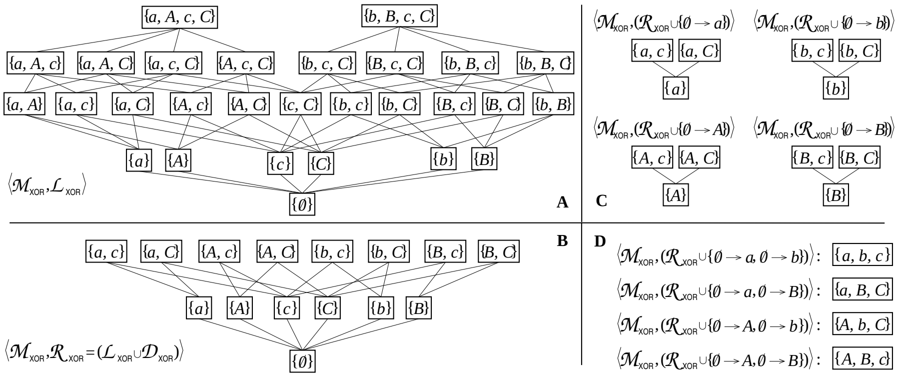

Fig. 1. Hierarchy of organisations for the chemical reaction network implementing an XOR logic gate. (A) The network consists only of the logical reactions $\mathcal{L}_{\text{XOR}}$. (B) Destructive reactions $\mathcal{D}_{\text{XOR}}$ are added to exclude contradictions. The resulting reaction network $\langle \mathcal{M}_{\text{XOR}}, \mathcal{R}_{\text{XOR}} \rangle$ implements the XOR logic gate without any input specified. (C) One input is defined by adding one influx reaction. (D) Adding the second input. The hierarchy of organisations collapses from (A) to (D), with the desired output as the only organisation left in (D). (Appeared in [25])

Now we set the input variables of the boolean network a and b to initiate the computational process. For the initialisation, an inflow reaction is added to the reaction network. We start with providing one input only, leaving the other input variable undefined. Fig. 1(C) shows the results for the four resulting algebraic chemistries. We can see that providing one input signal has further reduced the behavioural freedom of the reaction system. Only three combinations of molecular species are left, which may be encountered in the reaction vessel as a stationary state. Furthermore we can see that – in this special case – the output is not determined from a stoichiometric point of view since, in all four Hasse diagrams, sets containing c and C are found to be closed and self-maintaining.

When we finally provide both inputs, the Hasse diagram of organisations collapses so that only one organisation remains for every input condition, see Fig. 1(D). This implies that, no matter how we chose the dynamics, no other molecular species than those of the organisation can be sustained in the

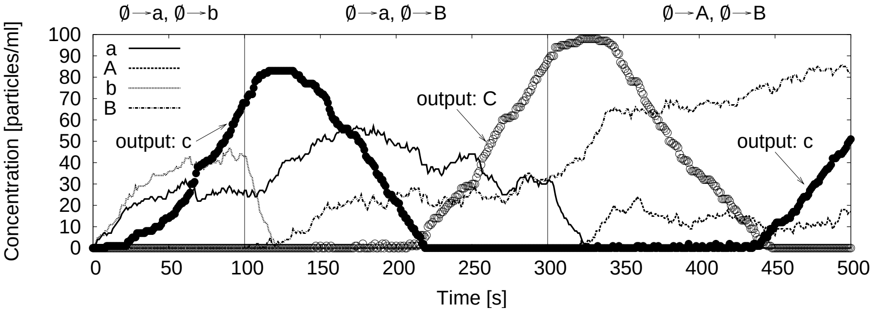

Fig. 2. Dynamic behaviour of the chemical reaction network implementing an XOR logic gate. The time course of all six molecular species is shown. Irreversible mass action kinetics are assumed for all reactions. Reaction rates are set to $k = 0.001$ for logical reactions. Reaction rates of destruction reactions are set to $k = 0.1$. For all irreversible constant influxes (e.g. $\emptyset \to A$), the rates are set to $k = 1$. The reaction system is stochastically simulated with the biochemical network simulator *Copasi* using a compartment size of 1 ml. See text for details. (Appeared in [25])

reaction vessel regardless of the initial state. We can see that the remaining organisation contains the desired output molecular species c or C, respectively.

To validate the results from applying organisation theory to the XOR reaction network, stochastic simulations are performed using the simulator packages *MGS* [14] and *Copasi* [32].

Fig. 2 shows a typical simulation run. The influx is defined as an irreversible constant flux with kinetic parameter set to 1. For all other reactions we chose irreversible mass action kinetics. The parameters for the destructive reactions $\mathcal{D}_{\text{XOR}}$ are set to $k = 0.1$ and those for the logical reactions $\mathcal{L}_{\text{XOR}}$ are set to $k = 0.001$. At several simulation times the input is changed in order to observe the switching of the XOR gate. Initially, there exist no molecular particles in the reactor and two influxes of a and b are present. This corresponds to the case in which both the input variables a and b are set to 0. Since molecular species c is generated, the output is computed to $\mathsf{c} = 0$.

At simulation time $100s$ the content of input variable b is switched to 1 by replacing the influx of molecular species b with the influx $\emptyset \to B$. The molecular particles b and c, whose concentrations are still high from the previous computation, deteriorate and finally vanish. The desired output C does not appear until the time point of approximately $200s$. Then, instead of a, the molecular species A is applied as an input starting from simulation time $300s$. The remaining molecules of species a and C from the previous computation decay first and the desired answer c appears in the end.

3.2 Maximal Independent Set Problem – A Chemical Algorithm and a Small Example

General Algorithm

The *maximal independent set (MIS) problem* is formally defined as follows. Let an *undirected graph* $G = (V, E)$ be defined by a set of N vertices $V = \{v_1, \ldots, v_N\}$ and a set of edges E. When two vertices v_p and v_q are connected, the pair of the vertices is in the set of edges, i.e. $(v_p, v_q) \in E$. Note that the order of the pair is insignificant, that is $(v_p, v_q) = (v_q, v_p)$. A set of vertices $I \subset V$ is *independent* if no two vertices in the set are adjacent, i.e. $\forall v_p, v_q \in I : (v_p, v_q) \notin E$. An independent set is *maximal* if no vertex can be added to the set while keeping the property of independence. Including another vertex in a maximal independent set would violate the independence property.

A reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ is designed as follows. For each vertex v_j, we assign two molecular species s_j^0 and s_j^1 representing the membership of the vertex in the MIS. High concentration of species s_j^1, higher than a threshold chosen to be smaller than any positive coordinate of any fixed point, means that the vertex v_j is included in the MIS. High concentration of species s_j^0 expresses that the vertex v_j is not included in the MIS. Thus the set of molecular species $\mathcal{M}$ contains $2N$ molecular species:

$$\mathcal{M} = \{s_j^0, s_j^1 \mid j = 1, \ldots, N\}. \tag{5}$$

The set of reaction rules $\mathcal{R}$ is constructed by assembling reactions for each vertex:

$$\mathcal{R} = \bigcup_{i=1}^{N} \mathcal{R}^i = \bigcup_{i=1}^{N} (\mathcal{V}^i \cup \mathcal{N}^i \cup \mathcal{D}^i). \tag{6}$$

A reaction rule to *produce species* s_i^1 is the first:

$$\mathcal{V}^i = (\overbrace{s_j^0 + s_k^0 + \cdots + s_l^0}^{n_i} \rightarrow n_i s_i^1) \tag{7}$$

where n_i is the number of vertices connected to vertex v_i and $v_j, v_k, \ldots, v_l$ are its neighbouring vertices, that is $(v_i, v_j), (v_i, v_k), \ldots, (v_i, v_l) \in E$. This reaction is interpreted as follows. When no neighbouring vertex is included in the MIS, the target vertex v_i should be included in the set.

The negation of this predicate is considered by a set of n_i reactions:

$$\mathcal{N}^i = \{s_j^1 \rightarrow s_i^0 | (v_i, v_j) \in E\}. \tag{8}$$

This is the second type of reactions, which produce species s_i^0 from any species corresponding to the neighbouring vertices with superscript 1. This rule can be interpreted as follows. If there exists at least one neighbouring vertex included in the MIS, then the target vertex v_i should be excluded from the maximal independent set (otherwise the definition of the MIS would be violated).

The last component of set $\mathcal{R}^i$ is a *destructive reaction*. Since the membership of the MIS is a binary state, the state becomes undefined when neither or both of the species are present. In order to avoid the latter case, the two opposite molecular species are defined to vanish upon collision:

$$\mathcal{D}^i = s_i^0 + s_i^1 \to \emptyset. \tag{9}$$

Note that the reaction network is defined such that molecules react only if they are located on the same vertex or are neighbours. Thus the resulting (artificial) chemical system can be interpreted as a spatially distributed compartmentalised reaction system, where a compartment j holds only the two chemical species representing a vertex v_j, namely s_j^0 and s_j^1 and where the topological structure of the compartments is equivalent to the undirected graph.

Small Example

Provided that an undirected graph $G = (V, E)$ consists of three vertices and those vertices are connected linearly as shown in Fig. 3(A), i.e.

$$G = (\{v_1, v_2, v_3\}, \{(v_1, v_2), (v_2, v_3)\}). \tag{10}$$

Following the recipe, a reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ is constructed. The set of molecular species $\mathcal{M}$ consists of six species because the graph contains $N = 3$ vertices:

$$\mathcal{M} = \{s_1^0, s_1^1, s_2^0, s_2^1, s_3^0, s_3^1\}. \tag{11}$$

According two the previous section the set of reactions $\mathcal{R}$ results in:

$$\begin{aligned}\mathcal{R} = \{&s_2^0 \to s_1^1, s_2^1 \to s_1^0, s_1^0 + s_3^0 \to 2s_2^1,\\ &s_1^1 \to s_2^0, s_3^1 \to s_2^0, s_2^0 \to s_3^1, s_2^1 \to s_3^0,\\ &s_1^0 + s_1^1 \to \emptyset, s_2^0 + s_2^1 \to \emptyset, s_3^0 + s_3^1 \to \emptyset\}.\end{aligned}$$

The reaction network is analysed for its hierarchical chemical organisational structure within the reaction network. In our example, the reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ possesses five organisations:

$$O = \{\emptyset, \{s_1^0\}, \{s_3^0\}, \{s_1^0, s_2^1, s_3^0\}, \{s_1^1, s_2^0, s_3^1\}\}. \tag{12}$$

Fig. 3(B) visualises these organisations as a Hasse diagram. We note that the organisations do not form a lattice, because there is not a unique largest organisations. The two largest organisations represent the two desired solutions to the MIS problem, namely "010" and "101". This explains that in a dynamical reaction system implementing the designed reaction network, the species combinations representing desired solutions are more likely to stay in the dynamical system and the other solutions consisting of species that are not an organisation cannot stably exist [11].

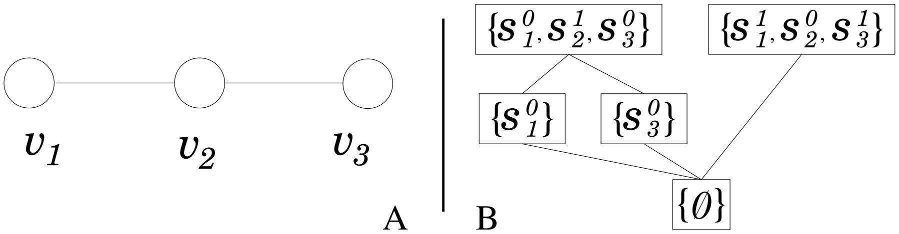

Fig. 3. Analysis of a chemical program with organisation theory. (A) Graph structure and (B) hierarchy of organisations within the chemical reaction network for the maximal independent set problem for the linear 3-vertex graph. (Appeared in [26])

4 Design Principles

4.1 Design Principles Derived from Heuristics

When following an organisation-oriented approach, we first concentrate on the reaction network neglecting kinetic laws. The reaction network is designed with respect to its organisational structure, considering the following principles **P1**–**P6**. Then, in the second step, the kinetics including kinetic parameters is specified for fine tuning of the computation as stated in **P7**. The kinetic laws determine the dynamics between and inside organisations.

P1: There should be one organisation for each output behaviour class.
Assume that computation appears as a movement between organisations and the output behaviour can be categorised in different discrete behaviour classes. That is, species combinations in the computational reaction system uniquely identify the output behaviour. The reaction network should be designed so that there should exist at least one organisation corresponding to each output behaviour class categorised. For instance there are two organisations in the MIS problem example corresponding to the results, cf. the two biggest organisations in Fig. 3(B).

P2: The set of molecular species (and the organisation) representing a result should be in the closure of the species representing the initial input.
This principle assures that there is a reaction path from the initial input configuration to the desired output species. Otherwise, the desired output will not appear as a result of the computation. The chemical XOR for example are designed following this principle. Furthermore, it is expected that the desired output set is contained in a self-maintaining set within that closure. The self-maintenance property of the set of molecular species indicates theoretical possibilities to sustain all the species in the dynamical reaction systems, so the desired output species may be sustained in the reaction system until the outcomes of the computation is observed. The ideal case is that the desired output is represented by a largest self-maintaining set within that closure. In

case that there exists a larger self-maintaining set than the desired output set, the dynamics may settle above the desired one. This argument leads to the next principle.

P3: The set of molecular species representing an input should generate the organisation representing the desired output.
This principle will be fulfilled on the following two conditions. The desired output is contained within the closure of the input (**P2** is fulfilled) and the largest self-maintaining set contained in the closure corresponds to the desired output. For the chemical XOR examples this was also the case. When inflow reactions are added, the closure of the input species turns out to be also self-maintaining. The largest self-maintaining set within a closure is not always unique in general. But this is not required and it can even be beneficial.

P4: Eliminate organisations not representing a desired output.
Since each organisation potentially includes fixed points, the reaction system's dynamics may converge to one of the organisations. Hence, it makes sense to eliminate organisations not representing an output in order to avoid false computational outputs. This can be achieved by destroying either its closure property or its self-maintenance.

P5: An output organisation should have no organisation below.
The dynamics of the reaction system that moves from one organisation O_1 to another O_2 below (i.e. $O_2 \subseteq O_1$) is called a downward movement. This dynamical move can be theoretically prevented by the self-maintenance property with the right kinetics. Practically speaking, this move may occur spontaneously due to, e.g. stochastic effects, because the self-maintenance property only ensures the possibilities to sustain all species. Following this principle, a downward movement can be restricted.

P6: Assure, if possible, stoichiometrically the stability of an output organisation.
Instead of eliminating organisations below the desired output as in the previous principle **P5**, the downward movement can be ruled out by a purely stoichiometric argument. It may be possible to design the reaction network such that the organisation representing the desired output is stable for any kinetic law. As a simple example consider the system $\mathcal{R} = \{a \to b, b \to a\}$, which has the two organisations $\{\emptyset\}$ and $\{a, b\}$. Due to mass-conservation, the system can never move spontaneously from the organisation with two species to the empty one.

P7: Use kinetic laws for fine tuning.
The kinetic laws determine the system's behaviour within an organisation and the transition dynamics between organisations. One of rationales for the right kinetics is to assure that the dynamical reaction systems are stable in the output organisations, restricting mainly the downward movement. The existence of such laws is ensured by chemical organisation theory to a certain extend and with classical dynamical systems theory it is even possible to derive at least in some cases rigorously dynamical stability from network structure [9, 12]. Another point of consideration is a trade-off between stability and the

speed of computation since chemical reaction systems may compute by moving amongst organisations.

4.2 Design by Evolution

In the previous section we described heuristics for a constructive programming strategy. This is similar to conventional programming, for which the target problem should be divided into sub-problems or at least the programmer should be aware of the structure of the problem intuitively or logically. While that constructive approach for designing the chemical reaction network in vivo has also been pursued [16], our focus here is to design chemical systems in an *autonomous* manner by *evolution*. The main difference is that the programmer is now free of the structural analysis of the problems. In other words, it is not necessarily clear for the programmer how the target problem is solved.

Comparing these two approaches of constructing and evolving, the engineered system tends to be simpler and more effective because irrational components are usually omitted. These simplicity and effectiveness lead to a stiffness so that the system is intolerant to modifications. Tiny modifications cause unpredictable and often negative effects on the system such as function failure. It may also be the case that major system changes cause no effects at all. On the other hand, evolved systems embrace more components than necessary. This extra complexity may lead to distinctive characteristics such as *robustness* and *adaptability*. The evolved systems may be robust against a functional failure of components, for example, by assigning an identical sub-function to different components in order to sustain the functionality as a whole. As an example we mention the comparison between the two approaches for a reaction network realising a flip-flop [22].

Making use of the advantages of evolvability is believed to distinguish the chemical computing systems from conventional computers because conventional computing systems exhibit severe difficulties on adaptability etc. due to the intolerance to modifications.

4.3 Design by Exploration

We discuss an approach for *autonomous design*, cooperating with principles of *exploration* [24]. This explorative approach is fundamentally different because programming is not associated with modification of chemical computing systems. Instead, systems are explored and searched for interesting behaviours. The basic idea is that an autonomous system is used, as a preliminary step, to explore the behaviour of the chemical reaction system. Then a specific aspect of the system's behaviour will be utilised for a particular computational purpose. When dealing with real chemical systems, this approach has an apparent advantage because modifying the reaction network is very much restricted.

The scouting algorithm developed by Pfaffmann and Zauner [30] is an evolutionary experimentation method for autonomous experimentation. Experiments are dynamically scheduled to explore systems' behaviour such that maximal information gain at each step is achieved. In accordance with communication theory, information is quantified as the surprise value of arriving data [8]. Motivations of this algorithm are to obtain experimental data sufficient to build quantitative system-level models as intended in Systems Biology [18]. For this purpose computational techniques have not only to discover regularities in existing data, but rather the experimental procedure itself has to be embedded in a closed-loop discovery process [21, 17].

Kulkarni and Simon [19] developed a program that attempts to generate experiments, in which unexplained phenomena are enhanced. Notably, the program does not start out with a pre-set goal as is common in optimisation experiments but decides on its objectives dynamically. This work demonstrated that an algorithm can successfully navigate an immense search space by emulating the interplay of adjusting hypotheses and modifying experiments, which is characteristic of human experimenters [15].

5 Conclusion

In this paper we reviewed a theoretical analysis method that helps to discover and implement computing capabilities in (artificial) chemical reaction networks. Given a list of molecular species and a list of reaction rules, the reaction network is decomposed into a hierarchy of closed and self-maintaining sub-networks called organisations. We have shown that this structure helps to assess the emergent dynamical behaviour of the chemical reaction network under study. In particular, we argue that the computation can be understood as a movement between organisations.

When the approach is applied to a reaction network implementing an XOR logic gate, the hierarchy of organisations helps to predict its emergent dynamical behaviour. Defining different inputs leads to different organisations corresponding to the various states of the gate. Even though a few of the logic gates are connected, the hierarchy of organisations is helpful for analysing the emergent dynamical behaviour. The second example showed how to implement a distributed and robust chemical algorithm to solve the MIS problem. The analysis via organisations gives an insight to the possible dynamical behaviour.

When designing a system with numerous small, extensively interacting components, its global behaviour cannot easily be predicted from the known local interactions. A general theory of emergence is desirable not only for analytical purposes but also for engineering such systems. We mentioned three different approaches to the design of artificial chemical reaction networks. Firstly, the design principles **P1**–**P7** are heuristic guidelines for the manual construction of chemical algorithms. Secondly and thirdly the two approaches, evolution and exploration, share the common assumption that the effects of changing a reaction system are hard to predict in advance. The evolutionary design approach modifies reaction networks and evaluates the changes afterwards. A prediction process is not involved in this way of programming. For the explorative approach, the intention to alter reaction systems are even disregarded. When dealing with natural systems, however, this assumption is believed to be appropriate since interactions established within the systems are complex and tangled. That way, there is the possibility to utilise the abundant complexity embraced within natural systems.

The results reviewed in this paper suggest that the theory of chemical organisations is a promising candidate to contribute to a general theoretical framework to master self-organisation in complex chemical-like information systems.

References

[1] L. M. Adleman Molecular computation of solutions to combinatorical problems. *Science*, 266:1021, 1994.

[2] J.-P. Banâtre and D. L. Métayer. A new computational model and its discipline of programming. Tech. Rep. RR-0566, INRIA, 1986.

[3] J.-P. Banâtre and D. L. Métayer. The GAMMA model and its discipline of programming. *Sci. Comput. Program.*, 15(1):55–77, 1990.

[4] J.-P. Banâtre, P. Fradet, and Y. Radenac. Principles of chemical programming. In S. Abdennadher and C. Ringeissen, editors, *RULE'04 Fifth International Workshop on Rule-Based Programming*, pages 98–108. Tech. Rep. AIB-2004-04, Dept. of Comp. Sci., RWTH Aachen, Germany, 2004.

[5] W. Banzhaf, P. Dittrich, and H. Rauhe. Emergent computation by catalytic reactions. *Nanotechnology*, 7(1):307–314, 1996.

[6] G. Berry and G. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96(1):217–248, 1992.

[7] U. S. Bhalla and R. Iyengar. Emergent properties of networks of biological signaling pathways. *Science*, 283:381–387, 1999.

[8] C. Cherry. *On Human Communication: A Review, a Survey, and a Criticism*, chapter Chapter 5. MIT Press, Cambridge, MA, 2nd edition, 1966.

[9] B. L. Clarke. Stability of complex reaction networks. *Advances in Chemical Physics*, 42:1–213, 1980.

[10] P. Dittrich. The bio-chemical information processing metaphor as a programming paradigm for organic computing. In U. Brinkschulte, J. Becker, C. Hochberger, T. Martinetz, C. Müller-Schloer, H. Schmeck, T. Ungerer, and R. Würtz, editors, *ARCS '05 - 18th International Conference on Architecture of Computing Systems 2005*, pages 95–99. VDE Verlag, Berlin, 2005.

[11] P. Dittrich and P. Speroni di Fenizio. Chemical organisation theory. *Bulletin of Mathematical Biology*, 69(4):1199–1231, 2007.

[12] M. Feinberg and F. J. M. Horn. Dynamics of open chemical systems and the algebraic structure of the underlying reaction network. *Chem. Eng. Sci.*, 29(3):775–787, 1974.

[13] W. Fontana and L. W. Buss. "The arrival of the fittest": Toward a theory of biological organization. *B. Math. Biol.*, 56:1–64, 1994.

[14] J.-L. Giavitto and O. Michel. MGS: a rule-based programming language for complex objects and collections. In M. van den Brand and R. Verma, editors, *Electr. Notes in Theor. Comput. Sci.*, volume 59. Elsevier Science Publishers, 2001.

[15] D. Gooding. *Experiment and the Making of Meaning.* Kluwer Academic Publishers, Dordrecht, 1990.

[16] N. J. Guido, X. Wang, D. Adalsteinsson, D. McMillen, J. Hasty, C. R. Cantor, T. C. Elston, and J. J. Collins. A bottom-up approach to gene regulation. *Nature*, 439(7078):856–860, Feb 2006.

[17] R. D. King, K. E. Whelan, F. M. Jones, P. G. K. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, 2004.

[18] H. Kitano. Systems biology: A brief overview. *Science*, 295:1662–1664, 2002.

[19] D. Kulkarni and H. A. Simon. Experimentation in machine discovery. In J. Shrager and P. Langley, editors, *Computational Models of Scientific Discovery and Theory Formation*, pages 255–273. Morgan Kaufmann Pubishers, San Mateo, CA, 1990.

[20] B.-O. Küppers. *Information and the Origin of Life.* MIT Press, Cambridge, MA, 1990.

[21] P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Zytkow. *Scientific discovery: Computational exploration of the creative processes.* MIT Press, Cambridge, MA, 1987.

[22] T. Lenser, N. Matsumaru, T. Hinze, and P. Dittrich. Tracking the evolution of chemical computing networks. In S. Bullock, J. Noble, R. A. Watson, and M. A. Bedau, editors, *Proceedings of the Eleventh International Conference on Artificial Life*, pages 343–350. MIT Press, Cambridge, MA, 2008.

[23] N. Matsumaru and P. Dittrich. Organization-oriented chemical programming for the organic design of distributed computing systems. In *1st international conference on bio inspired models of network, information and computing systems (BIONETICS)*, volume 275 of *ACM International Conference Proceeding*, Cavalese, Italy, December 11-13 2006. IEEE.

[24] N. Matsumaru, F. Centler, K.-P. Zauner, and P. Dittrich. Self-adaptive scouting - autonomous experimentation for systems biology. In G. R. Raidl, S. Cagnoni, J. Branke, D. W. Corne, R. Drechsler, Y. Jin, C. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. D. Smith, and

G. Squillero, editors, *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, volume 3005 of *LNCS*, pages 52–61, Coimbra, Portugal, 5-7 Apr. 2004. Springer Verlag.

[25] N. Matsumaru, F. Centler, P. Speroni di Fenizio, and P. Dittrich. Chemical organization theory as a theoretical base for chemical computing. *International Journal of Unconventional Computing*, 3(4):285–309, 2007.

[26] N. Matsumaru, T. Lenser, T. Hinze, and P. Dittrich. Toward organization-oriented chemical programming: A case study with the maximal independent set problem. In F. Dressler and I. Carreras, editors, *Advances in Biologically Inspired Information Systems*, volume 69 of *Studies in Computational Intelligence*, pages 147–163. Springer, Berlin, 2007.

[27] C. Müller-Schloer. Organic computing: On the feasibility of controlled emergence. In *Proceedings of the 2nd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS2004*, pages 2–5. ACM Press, New York, 2004. ISBN 1-58113- 937-3. doi: http://doi.acm.org/10.1145/1016720.1016724.

[28] C. Müller-Schloer, C. von der Malsburg, and R. P. Würtz. Aktuelles Schlagwort: Organic Computing. *Informatik Spektrum*, 27(4):332–336, 2004.

[29] S. Peter and P. Dittrich. On the Relation between Organizations and Limit Sets in Chemical Reaction Systems (in print). *Advances in Complex Systems (ACS)*, 13(5):?, 2010.

[30] J. O. Pfaffmann and K.-P. Zauner. Scouting context-sensitive components. In D. Keymeulen, A. Stoica, J. Lohn, and R. S. Zebulum, editors, *The Third NASA/DoD workshop on Evolvable Hardware*, pages 14–20, Long Beach, California, 12-14 July 2001. Jet Propulsion Laboratory, California Institute of Technology, IEEE Computer Society.

[31] G. Păun. Computing with membranes. *J. Comput. Syst. Sci.*, 61(1): 108–143, 2000.

[32] S. Sahle, R. Gauges, J. Pahle, N. Simus, U. Kummer, S. Hoops, C. Lee, M. Singhal, L. Xu, and P. Mendes. Simulation of biochemical networks using copasi - a complex pathway simulator. In *Winter Simulation Conference, 2006. WSC 06. Proceedings of the*, pages 1698–1706, 3-6 Dec. 2006.

[33] P. Speroni di Fenizio and P. Dittrich. Artificial chemistry's global dynamics. movement in the lattice of organisation. *The Journal of Three Dimensional Images*, 16(4):160–163, 2002.

[34] S. Tsuda, M. Aono, and Y.-P. Gunji. Robust and emergent *physarum* logical-computing. *Biosystems*, 73(1):45–55, January 2004.

[35] C. von der Malsburg. The challenge of organic computing, 1999. Memorandum, Comp. Sci. Dept.

[36] R. P. Würtz. Organic computing for face and object recognition. In P. Dadam and M. Reichert, editors, *Informatik 2004*, volume 2, pages 636–640. Gesellschaft für Informatik, 2004.