

Analysing Adaption Processes of Hornets

Michael Köhler-Bussmeier¹, Heiko Rölke²

¹University of Applied Science Hamburg, Berliner Tor 7, D-20099 Hamburg

²FH Graubünden, Pulvermühlestrasse 57, CH-7000 Chur

Abstract

In this paper we study the dynamics of self-adapting systems. Our main objective is to *compare* adaption dynamics of similar systems, i.e. systems that differ only in their organisational networks. We specify adaption in the formalisms of HORNETS – a formalism that uses nets as tokens, i.e. they follow the *nets-within-nets* approach. We identify different abstractions of the reachability graph that focus on the adaption aspects of the processes. We develop key measures for adaption processes on these abstracted graphs. The key measures are used e.g. to compare two variations of the same adapting system. The approach is illustrated by a case study: We analyse a HORNET-model of Axelrod’s well-known tournament where the playing agents adapt their strategies during the game.

Keywords

Adaption, Hornets, nets-within-nets, key measures, organisational networks

1. Introduction: The Dynamics of Adaption Processes

In general the evolution (i.e. adaption) of a system is determined by the interaction aspect of adaptivity, i.e. how adaption is performed (e.g. via a cross-over of genes) and how the gene pool looks like (the “how”). But, on the other hand it is relevant which objects have the possibility to interact: Is there a opportunity for each pair of genes to meet and perform a cross-over in the system – or are they too separated? This aspect relates to the *network* aspect of adaptivity (the “who”). A company’s workflow management system constitutes an example: The set of workflows is the gene pool, while the evolution of the workflows during the company’s lifetime strongly relates to the organisational network.

In this paper we like to study adaption processes where the network part plays a significant role which cannot be ignored. We like to study these adaption processes, i.e. the trajectories of the gene pool, in the setting of discrete event system, namely in the formal framework of HORNETS [1] as this formalism has been developed to study adaptive systems. In the context of HORNETS these two aspects – how and whom – are naturally addressed: HORNETS are Petri nets where the tokens are Petri nets again. The surrounding net is called system net and the net-tokens are called object nets. The object nets model the adapting objects – as they define what can happen; and the system net models the network part together with the adaption logic

PNSE’22, International Workshop on Petri Nets and Software Engineering, Bergen, Norway, 2022

✉ michael.koehler-bussmeier@haw-hamburg.de (M. Köhler-Bussmeier); heiko.roelke@fhgr.ch (H. Rölke)

🌐 <https://www.haw-hamburg.de/michael-koehler-bussmeier> (M. Köhler-Bussmeier);

<https://www.fhgr.ch/personen/person/roelke/> (H. Rölke)

🆔 0000-0002-3074-4145 (M. Köhler-Bussmeier)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

itself – as the system net defines which object nets are involved and how adaption looks like. In the example of a workflow management system the object nets describe the set of workflows and the net-tokens are the workflow instances while the system net models the whole adaption process of workflows.

A state μ of a HORNET is a nested multiset, where a place \hat{p}_i of the system net is marked with net-tokens of the form $[N_i, M_i]$; the topology of a net-token is defined by the object net N_i and each net-token has its own marking M_i :

$$\mu = \sum_{i=1}^n \overbrace{\hat{p}_i}^{\text{system net}} \left[\underbrace{N_i}_{\text{object net}}, \underbrace{M_i}_{\text{ON marking}} \right]_{\text{net-token}}$$

For HORNETS the gene pool of this state μ is given as the set of object nets, i.e. the set of possible net-token types: $\{N_1, \dots, N_n\}$. We are interested in the dynamics of this set. More specifically, our main objective is to *compare* adaption dynamics of similar systems. For the example of a workflow management system we like to compare two companies with the same initial set of workflows but with different organisational networks, i.e. we like to study the impact of the network on the adaption dynamics. We like to develop key values for these dynamics to answer questions like: “Which company adapts faster?”, “Which company develops a gene pool of greater diversity?” etc.

The paper has the following structure: Section 2 recalls the definition of HORNETS and EHORNETS. The main purpose is to recall the definition of the reachability graph of HORNETS. It can be skipped on first reading. The main contribution is given in Section 3, which describes how adaption process can be studied in the formal framework of HORNETS. We will derive a structure describing the core adaption processes as an abstraction of the reachability graph of a given HORNET. The dynamics of the gene pool is somehow in between two other dynamics: on the one hand, it is much coarser than the dynamics of the whole HORNET as it abstracts from the internal actions of the net-tokens; on the other hand is much finer than the dynamics of the system net alone when considered as a normal p/t net. Different key values will be considered to capture essential aspect of the adaption dynamics. In Section 4 we exemplify the theoretical framework with a generalisation of Axelrod’s well-known tournament. The tournament is a well-known example for adaption, especially in the context of genetic algorithms [2]. Here, we study the impact of differences in the agents’ neighbourhood for the adaption dynamics: We compare the dynamics on a *Erdős-Rényi graph* [30] to that on a *Watts-Strogatz graph* [31]. The work ends with a conclusion and an outlook.

2. Hornets and Object Nets

We have defined HORNETS in [1] as a generalisation of our object nets [3, 4], which follow the *nets-within-nets* paradigm as proposed by Valk [5]. Approaches adapting the nets-within-nets approach are nested nets [6], mobile predicate/transition nets [7], Reference nets [8], PN² [9], hypernets [10], Mobile Systems [11], and adaptive workflow nets [12]. Object Nets can be seen as the Petri net perspective on contextual change, in contrast to the Ambient Calculus [13] or the π -calculus [14].

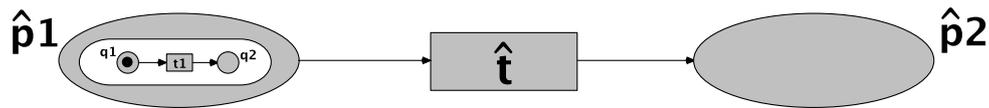


Figure 1: An Elementary Object Net System (Eos)

With object nets we study Petri nets where the tokens are nets again, i.e. we have a nested marking. Events are also nested. We have three different kinds of events – as illustrated by the example given in Figure 1:

1. System-autonomous: The system net transition \hat{t} fires autonomously, which moves the net-token from \hat{p}_1 to \hat{p}_2 without changing its marking.
2. Object autonomous: The object net fires transition t_1 “moving” the black token from q_1 to q_2 . The object net remains at its location \hat{p}_1 .
3. Synchronisation: Whenever we add matching synchronisation inscriptions at the system net transition \hat{t} and the object net transition t_1 , then both must fire synchronously: The object net is moved to \hat{p}_2 and the black token moves from q_1 to q_2 inside. Whenever synchronisation is specified, autonomous actions are forbidden.

For HORNETS we extend object nets with algebraic concepts that allow to modify the structure of the net-tokens as a result of a firing transition. This is a generalisation of the approach of algebraic nets [15], where algebraic data types replace the anonymous black tokens.

Example We consider a HORNET with two workflow nets N_1 and N_2 as tokens – cf. Figure 2. To model a run-time adaption, we combine N_1 and N_2 resulting in the net $N_3 = (N_1 \parallel N_2)$. This modification is modelled by transition t of the HORNETS in Fig. 2. In a binding α with $x \mapsto N_1$ and $y \mapsto N_2$ the transition t is enabled. Assume that $(x \parallel y)$ evaluates to N_3 for α . If t fires it removes the two net-tokens from p and q and generates one new net-token on place r . The net-token on r has the structure of N_3 and its marking is obtained as a transfer from the token on v in N_1 and the token on s in N_2 into N_3 . This transfer is possible since all the places of N_1 and N_2 are also places in N_3 and tokens can be transferred in the obvious way. c

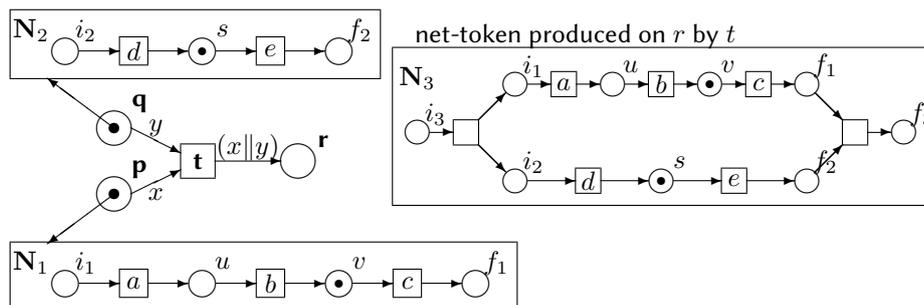


Figure 2: Modification of the net-token’s structure

It is not hard to prove that the general HORNET formalism is Turing-complete. In [1] we have proven that there are several possibilities to simulate counter programs: One could use the

nesting to encode counters. Another possibility is to encode counters in the algebraic structure of the net operators.

The use of algebraic operations in HORNETS relates them to *algebraic higher-order (AHO) systems* [16], which are restricted to two-levelled systems but have a greater flexibility for the operations on net-tokens, since each net transformation is allowed. There is also a relationship to Nested Nets [17], which are used for adaptive systems, and to distributed graph transformations [18].

2.1. Adaption: Complexity Issues for eHORNETS

Let us recall our results on *complexity* issues of HORNETS. We introduced different restrictions to guarantee that the system has a finite state space: First, we allow at most one token on each place, which results in the class of *safe HORNETS*. However, this restriction does not guarantee finite state spaces, since we have the nesting depth as a second source of undecidability [4]. Second, we restrict the universe of object nets to finite sets. Finally, we restrict the nesting depth and introduce the class of *elementary HORNETS*, which have a two-levelled nesting structure. This is done in analogy to the class of *elementary object net systems (Eos)* [3], which are the two-level specialisation of general object nets [3, 4]. If we rule out these sources of complexity the remaining origin of complexity is the use of algebraic transformations, which are still allowed for safe, elementary HORNETS – a class defined in analogy to *safe Eos* [19].

Safe, elementary HORNETS have greater complexity when compared to their Eos counterpart: On the one hand, we have shown in [19, 20, 21] that most problems for *safe Eos* are PSPACE-complete. More precisely: All problems that are expressible in LTL or CTL, which includes reachability and liveness, are PSPACE-complete. This means that with respect to these problems *safe Eos* are no more complex than P/T nets. On the other hand, we have shown in [22] that *safe, elementary HORNETS* are beyond PSPACE: We have shown a lower bound, i.e. that “the reachability problem requires exponential space” for safe, elementary HORNETS – similarly to well known result of for *bounded P/T nets* [23]. In [24] we give an algorithm that needs at most exponential space, which shows that lower and upper bound coincide.

In [25] we studied further restrictions of Elementary HORNETS – among them state machines. The main result is that the reachability problem is PSPACE-complete for this class.

2.2. Formal Definition of Elementary Hornets (eHORNETS)

In the following we recall the definition of eHORNETS from [22] which is specialised from the general formalism of HORNETS [1].

Multisets and P/T Nets A multiset \mathbf{m} on the set D is a mapping $\mathbf{m} : D \rightarrow \mathbb{N}$. Multisets can also be represented as a formal sum in the form $\mathbf{m} = \sum_{i=1}^n x_i$, where $x_i \in D$.

Multiset addition is defined component-wise: $(\mathbf{m}_1 + \mathbf{m}_2)(d) := \mathbf{m}_1(d) + \mathbf{m}_2(d)$. The empty multiset $\mathbf{0}$ is defined as $\mathbf{0}(d) = 0$ for all $d \in D$. Multiset-difference $\mathbf{m}_1 - \mathbf{m}_2$ is defined by $(\mathbf{m}_1 - \mathbf{m}_2)(d) := \max(\mathbf{m}_1(d) - \mathbf{m}_2(d), 0)$.

The cardinality of a multiset is $|\mathbf{m}| := \sum_{d \in D} \mathbf{m}(d)$. A multiset \mathbf{m} is finite if $|\mathbf{m}| < \infty$. The set of all finite multisets over the set D is denoted $MS(D)$. The *domain* of a multiset is

$\text{dom}(\mathbf{m}) := \{d \in D \mid \mathbf{m}(d) > 0\}$.

Multiset notations are used for sets as well. The meaning will be apparent from its use.

Any mapping $f : D \rightarrow D'$ extends to a multiset-homomorphism $f^\# : MS(D) \rightarrow MS(D')$ by $f^\#(\sum_{i=1}^n x_i) = \sum_{i=1}^n f(x_i)$.

A *p/t net* N is a tuple $N = (P, T, \mathbf{pre}, \mathbf{post})$, such that P is a set of places, T is a set of transitions, with $P \cap T = \emptyset$, and $\mathbf{pre}, \mathbf{post} : T \rightarrow MS(P)$ are the pre- and post-condition functions. A marking of N is a multiset of places: $\mathbf{m} \in MS(P)$. We denote the enabling of t in marking \mathbf{m} by $\mathbf{m} \xrightarrow{t}$. Firing of t is denoted by $\mathbf{m} \xrightarrow{t} \mathbf{m}'$.

Net-Algebras We define the algebraic structure of object nets. For a general introduction of algebraic specifications cf. [26].

Let K be a set of net-types (kinds). A (many-sorted) *specification* (Σ, X, E) consists of a signature Σ , a family of variables $X = (X_k)_{k \in K}$, and a family of axioms $E = (E_k)_{k \in K}$.

A signature is a disjoint family $\Sigma = (\Sigma_{k_1 \dots k_n, k})_{k_1, \dots, k_n, k \in K}$ of operators. The set of terms of type k over a signature Σ and variables X is denoted $\mathbb{T}_\Sigma^k(X)$.

We use (many-sorted) predicate logic, where the terms are generated by a signature Σ and formulae are defined by a family of predicates $\Psi = (\Psi_n)_{n \in \mathbb{N}}$. The set of formulae is denoted PL_Γ , where $\Gamma = (\Sigma, X, E, \Psi)$ is the *logic structure*.

Object Nets and Net-Algebras Let Σ be a signature over K . A *net-algebra* assigns to each type $k \in K$ a set \mathcal{U}_k of object nets – the net universe. Each object $N \in \mathcal{U}_k, k \in K$ net is a p/t net $N = (P_N, T_N, \mathbf{pre}_N, \mathbf{post}_N)$. We identify \mathcal{U} with $\bigcup_{k \in K} \mathcal{U}_k$ in the following. We assume the family $\mathcal{U} = (\mathcal{U}_k)_{k \in K}$ to be disjoint.

The nodes of the object nets in \mathcal{U}_k are not disjoint, since the firing rule allows to transfer tokens between net tokens within the same set \mathcal{U}_k . Such a transfer is possible, if we assume that all nets $N \in \mathcal{U}_k$ have the same set of places P_k . P_k is the place universe for all object nets of kind k . In the example of Fig. 2 the object nets N_1, N_2 , and N_3 must belong to the same type since otherwise it would be impossible to transfer the markings in N_1 and N_2 to the generated N_3 .

In general, P_k is not finite. Since we like each object net to be finite in some sense, we require that the transitions T_N of each $N \in \mathcal{U}_k$ use only a finite subset of P_k , i.e. $\forall N \in \mathcal{U} : |\bullet T_N \cup T_N \bullet| < \infty$.

The family of object nets \mathcal{U} is the universe of the algebra. A *net-algebra* $(\mathcal{U}, \mathcal{I})$ assigns to each constant $\sigma \in \Sigma_{\lambda, k}$ an object net $\sigma^\mathcal{I} \in \mathcal{U}_k$ and to each operator $\sigma \in \Sigma_{k_1 \dots k_n, k}$ with $n > 0$ a mapping $\sigma^\mathcal{I} : (\mathcal{U}_{k_1} \times \dots \times \mathcal{U}_{k_n}) \rightarrow \mathcal{U}_k$.

A net-algebra is called *finite* if P_k is a finite set for each $k \in K$.

Since all nets $N \in \mathcal{U}_k$ have the same set of places P_k , which is required to be finite for EHNETS, there is an upper bound for the cardinality of \mathcal{U}_k .

Theorem 1 (Lemma 2.1 in [22]). *For each $k \in K$ the cardinality of each net universe \mathcal{U}_k is bound as follows: $|\mathcal{U}_k| \leq 2^{(2^{|P_k|})}$.*

A variable assignment $\alpha = (\alpha_k : X_k \rightarrow \mathcal{U}_k)_{k \in K}$ maps each variable onto an element of the algebra. For a variable assignment α the evaluation of a term $t \in \mathbb{T}_\Sigma^k(X)$ is uniquely defined and will be denoted as $\alpha(t)$.

A net-algebra, such that all axioms of (Σ, X, E) are valid, is called *net-theory*.

Nested Markings A marking of an eHORNET assigns to each system net place one or many net-tokens. The places of the system net are typed by the function $k : \hat{P} \rightarrow K$, meaning that a place \hat{p} contains net-tokens of kind $k(\hat{p})$. Since the net-tokens are instances of object nets, a *marking* is a *nested* multiset of the form:

$$\mu = \sum_{i=1}^n \hat{p}_i[N_i, M_i] \quad \text{where} \quad \hat{p}_i \in \hat{P}, N_i \in \mathcal{U}_k(\hat{p}_i), M_i \in MS(P_{N_i}), n \in \mathbb{N}$$

Each addend $\hat{p}_i[N_i, M_i]$ denotes a net-token on the place \hat{p}_i that has the structure of the object net N_i and the marking $M_i \in MS(P_{N_i})$. The set of all nested multisets is denoted as \mathcal{M}_H . We define the partial order \sqsubseteq on nested multisets by setting $\mu_1 \sqsubseteq \mu_2$ iff $\exists \mu : \mu_2 = \mu_1 + \mu$.

The projection $\Pi_N^1(\mu)$ is the multiset of all system-net places that contain the object net N :

$$\Pi_N^1 \left(\sum_{i=1}^n \hat{p}_i[N_i, M_i] \right) := \sum_{i=1}^n \mathbf{1}_N(N_i) \cdot \hat{p}_i \quad (1)$$

where the indicator function $\mathbf{1}_N$ is defined as: $\mathbf{1}_N(N_i) = 1$ iff $N_i = N$.

Analogously, the projection $\Pi_N^2(\mu)$ is the multiset of all net-tokens' markings (that belong to the object net N):

$$\Pi_N^2 \left(\sum_{i=1}^n \hat{p}_i[N_i, M_i] \right) := \sum_{i=1}^n \mathbf{1}_k(N_i) \cdot M_i \quad (2)$$

The projection $\Pi_k^2(\mu)$ is the sum of all net-tokens' markings belonging to the same type $k \in K$:

$$\Pi_k^2(\mu) := \sum_{N \in \mathcal{U}_k} \Pi_N^2(\mu) \quad (3)$$

Synchronisation The transitions in an HORNET are labelled with synchronisation inscriptions. We assume a fixed set of channels $C = (C_k)_{k \in K}$.

- The function family $\hat{l}_\alpha = (\hat{l}_\alpha^k)_{k \in K}$ defines the synchronisation constraints. Each transition of the system net is labelled with a multiset $\hat{l}^k(\hat{t}) = (e_1, c_1) + \dots + (e_n, c_n)$, where the expression $e_i \in \mathbb{T}_\Sigma^k(X)$ describes the called object net and $c_i \in C_k$ is a channel. The intention is that \hat{t} fires synchronously with a multiset of object net transitions with the same multiset of labels. Each variable assignment α generates the function $\hat{l}_\alpha^k(\hat{t})$ defined as:

$$\hat{l}_\alpha^k(\hat{t})(N) := \sum_{\substack{1 \leq i \leq n \\ \alpha(e_i) = N}} c_i \quad \text{for} \quad \hat{l}^k(\hat{t}) = \sum_{1 \leq i \leq n} (e_i, c_i) \quad (4)$$

Each function $\hat{l}_\alpha^k(\hat{t})$ assigns to each object net N a multiset of channels.

- For each $N \in \mathcal{U}_k$ the function l_N assigns to each transition $t \in T_N$ either a channel $c \in C_k$ or \perp_k , whenever t fires without synchronisation, i.e. autonomously.

System Net Assume we have a fixed logic $\Gamma = (\Sigma, X, E, \Psi)$ and a net-theory $(\mathcal{U}, \mathcal{I})$. An *elementary higher-order object net* (eHORNET) is composed of a system net \widehat{N} and the set of object nets \mathcal{U} . W.l.o.g. we assume $\widehat{N} \notin \mathcal{U}$. To guarantee finite algebras for eHORNETS, we require that the net-theory $(\mathcal{U}, \mathcal{I})$ is finite, i.e. each place universe P_k is finite.

The system net is a net $\widehat{N} = (\widehat{P}, \widehat{T}, \mathbf{pre}, \mathbf{post}, \widehat{G})$, where each arc is labelled with a multiset of terms: $\mathbf{pre}, \mathbf{post} : \widehat{T} \rightarrow (\widehat{P} \rightarrow MS(\mathbb{T}_\Sigma(X)))$. Each transition is labelled by a guard predicate $\widehat{G} : \widehat{T} \rightarrow PL_\Gamma$. The places of the system net are typed by the function $k : \widehat{P} \rightarrow K$. As a typing constraint we have that each arc inscription has to be a multiset of terms that are all of the kind that is assigned to the arc's place:

$$\mathbf{pre}(\widehat{t})(\widehat{p}), \quad \mathbf{post}(\widehat{t})(\widehat{p}) \in MS(\mathbb{T}_\Sigma^{k(\widehat{p})}(X)) \quad (5)$$

For each variable binding α we obtain the evaluated functions $\mathbf{pre}_\alpha, \mathbf{post}_\alpha : \widehat{T} \rightarrow (\widehat{P} \rightarrow MS(\mathcal{U}))$ in the obvious way.

Definition 1 (Elementary Hornet, eHORNET). Assume a fixed many-sorted predicate logic $\Gamma = (\Sigma, X, E, \Psi)$.

An *elementary HORNET* is a tuple $EH = (\widehat{N}, \mathcal{U}, \mathcal{I}, k, l, \mu_0)$ such that:

1. \widehat{N} is an algebraic net, called the system net.
2. $(\mathcal{U}, \mathcal{I})$ is a finite net-theory for the logic Γ .
3. $k : \widehat{P} \rightarrow K$ is the typing of the system net places.
4. $l = (\widehat{l}, l_N)_{N \in \mathcal{U}}$ is the labelling.
5. $\mu_0 \in \mathcal{M}_H$ is the initial marking.

Events The synchronisation labelling generates the set of system events Θ . We have three kinds of events:

1. **Synchronised firing:** There is at least one object net that has to be synchronised, i.e. there is a N such that $\widehat{l}(\widehat{t})(N)$ is not empty.
Such an event is a pair $\theta = \widehat{t}^\alpha[\vartheta]$, where \widehat{t} is a system net transition, α is a variable binding, and ϑ is a function that maps each object net to a multiset of its transitions, i.e. $\vartheta(N) \in MS(T_N)$. It is required that \widehat{t} and $\vartheta(N)$ have matching multisets of labels, i.e. $\widehat{l}(\widehat{t})(N) = l_N^\sharp(\vartheta(N))$ for all $N \in \mathcal{U}$. (Remember that l_N^\sharp denotes the multiset extension of l_N .)
The intended meaning is that \widehat{t} fires synchronously with all the object net transitions $\vartheta(N), N \in \mathcal{U}$.
2. **System-autonomous firing:** The transition \widehat{t} of the system net fires autonomously, whenever $\widehat{l}(\widehat{t})$ is the empty multiset $\mathbf{0}$.
We consider system-autonomous firing as a special case of synchronised firing generated by the function ϑ_{id} , defined as $\vartheta_{id}(N) = \mathbf{0}$ for all $N \in \mathcal{U}$.
3. **Object autonomous firing:** An object net transition t in N fires autonomously, whenever $l_N(t) = \perp_k$.
Object autonomous events are denoted as $id_{\widehat{p}, N}[\vartheta_t]$, where $\vartheta_t(N') = \{t\}$ if $N = N'$ and $\mathbf{0}$ otherwise. The meaning is that in object net N fires t autonomously within the place \widehat{p} .

For the sake of uniformity we define for an arbitrary binding α :

$$\mathbf{pre}_\alpha(id_{\widehat{p},N})(\widehat{p}')(N') = \mathbf{post}_\alpha(id_{\widehat{p},N})(\widehat{p}')(N') = \begin{cases} 1 & \text{if } \widehat{p}' = \widehat{p} \wedge N' = N \\ 0 & \text{otherwise.} \end{cases}$$

The set of all *events* generated by the labelling l is $\Theta_l := \Theta_1 \cup \Theta_2$, where Θ_1 contains synchronous events (including system-autonomous events as a special case) and Θ_2 contains the object autonomous events:

$$\begin{aligned} \Theta_1 &:= \left\{ \widehat{\tau}^\alpha[\vartheta] \mid \forall N \in \mathcal{U} : \widehat{l}_\alpha(\widehat{t})(N) = l_N^\#(\vartheta(N)) \right\} \\ \Theta_2 &:= \left\{ id_{\widehat{p},N}[\vartheta_t] \mid \widehat{p} \in \widehat{P}, N \in \mathcal{U}_{k(\widehat{p})}, t \in T_N \right\} \end{aligned} \quad (6)$$

Firing Rule A system event $\theta = \widehat{\tau}^\alpha[\vartheta]$ removes net-tokens together with their individual internal markings. Firing the event replaces a nested multiset $\lambda \in \mathcal{M}_H$ that is part of the current marking μ , i.e. $\lambda \sqsubseteq \mu$, by the nested multiset ρ . The enabling condition is expressed by the *enabling predicate* ϕ_{EH} (or just ϕ whenever EH is clear from the context):

$$\begin{aligned} \phi_{EH} \quad (\widehat{\tau}^\alpha[\vartheta], \lambda, \rho) &\iff \forall k \in K : \\ &\forall \widehat{p} \in k^{-1}(k) : \forall N \in \mathcal{U}_k : \Pi_N^1(\lambda)(\widehat{p}) = \mathbf{pre}_\alpha(\widehat{\tau})(\widehat{p})(N) \wedge \\ &\forall \widehat{p} \in k^{-1}(k) : \forall N \in \mathcal{U}_k : \Pi_N^1(\rho)(\widehat{p}) = \mathbf{post}_\alpha(\widehat{\tau})(\widehat{p})(N) \wedge \\ &\Pi_k^2(\lambda) \geq \sum_{N \in \mathcal{U}_k} \mathbf{pre}_N^\#(\vartheta(N)) \wedge \\ &\Pi_k^2(\rho) = \Pi_k^2(\lambda) + \sum_{N \in \mathcal{U}_k} \mathbf{post}_N^\#(\vartheta(N)) - \mathbf{pre}_N^\#(\vartheta(N)) \end{aligned} \quad (7)$$

The predicate ϕ_{EH} has the following meaning: Conjunct (1) states that the removed sub-marking λ contains on \widehat{p} the right number of net-tokens, that are removed by $\widehat{\tau}$. Conjunct (2) states that generated sub-marking ρ contains on \widehat{p} the right number of net-tokens, that are generated by $\widehat{\tau}$. Conjunct (3) states that the sub-marking λ enables all synchronised transitions $\vartheta(N)$ in the object N . Conjunct (4) states that the marking of each object net N is changed according to the firing of the synchronised transitions $\vartheta(N)$.

Note, that conjunct (1) and (2) assures that only net-tokens relevant for the firing are included in λ and ρ . Conditions (3) and (4) allow for additional tokens in the net-tokens.

For system-autonomous events $\widehat{t}^\alpha[\vartheta_{id}]$ the enabling predicate ϕ_{EH} can be simplified further: Conjunct (3) is always true since $\mathbf{pre}_N(\vartheta_{id}(N)) = \mathbf{0}$. Conjunct (4) simplifies to $\Pi_k^2(\rho) = \Pi_k^2(\lambda)$, which means that no token of the object nets get lost when a system-autonomous events fires.

Analogously, for an object autonomous event $\widehat{\tau}^\alpha[\vartheta_t]$ we have an idle-transition $\widehat{\tau} = id_{\widehat{p},N}$ and $\vartheta = \vartheta_t$ for some t . Conjunct (1) and (2) simplify to $\Pi_{N'}^1(\lambda) = \widehat{p} = \Pi_{N'}^1(\rho)$ for $N' = N$ and to $\Pi_{N'}^1(\lambda) = \mathbf{0} = \Pi_{N'}^1(\rho)$ otherwise. This means that $\lambda = \widehat{p}[M]$, M enables t , and $\rho = \widehat{p}[M - \mathbf{pre}_N(\widehat{t}) + \mathbf{post}_N(\widehat{t})]$.

Definition 2 (Firing Rule). Let EH be an $EHORNET$ and $\mu, \mu' \in \mathcal{M}_H$ markings.

- The event $\widehat{\tau}^\alpha[\vartheta]$ is enabled in μ for the mode $(\lambda, \rho) \in \mathcal{M}_H^2$ iff $\lambda \sqsubseteq \mu \wedge \phi_{EH}(\widehat{\tau}[\vartheta], \lambda, \rho)$ holds and the guard $\widehat{G}(\widehat{t})$ holds, i.e. $E \models_{\mathcal{I}}^{\alpha} \widehat{G}(\widehat{\tau})$.
- An event $\widehat{\tau}^\alpha[\vartheta]$ that is enabled in μ can fire – denoted $\mu \xrightarrow[EH]{\widehat{\tau}^\alpha[\vartheta](\lambda, \rho)} \mu'$.

- The resulting successor marking is defined as $\mu' = \mu - \lambda + \rho$.

Note, that the firing rule has no a-priori decision how to distribute the marking on the generated net-tokens. Therefore we need the mode (λ, ρ) to formulate the firing of $\widehat{\tau}^\alpha[\vartheta]$ in a functional way.

Reachability Graph Firing is extended to sequences $w \in \Theta_l^*$ in the usual way:

- The empty sequence $w = \epsilon$ is enabled iff $\mu' = \mu$.
- Whenever $\mu \xrightarrow{EH} \mu'$ and $\mu' \xrightarrow{\theta} \mu''$ then $\mu \xrightarrow{EH} \mu'$.

We denote $\mu \xrightarrow{EH}^* \mu'$ whenever there is some w such that $\mu \xrightarrow{EH} \mu'$ holds. We omit the subscript EH whenever it is clear from the context. The set of reachable markings is defined as:

$$RS(EH) := RS(\mu_0) := \left\{ \mu \mid \mu_0 \xrightarrow{EH}^* \mu \right\} \quad (8)$$

The *reachability graph* $RG(EH) = (V, E, \mu_0)$ contains all nested markings $V = RS(EH)$ as vertices (or: nodes), $E = \{(\mu, \mu') \mid \mu \xrightarrow{\theta} \mu'\}$ as edges and the initial marking μ_0 as a distinguished node.

3. Adaption Processes

For eHORNETS it is quite obvious how to define the gene pool of a system state: Obviously, we are interested in the (multi-)set of object nets that are contained in the marking μ . These are given by the projection $\Pi_{\mathcal{U}_k}(\mu)$:

$$\Pi_{\mathcal{U}_k} \left(\sum_{i=1}^n \widehat{p}_i[N_i, M_i] \right) := \sum_{i=1}^n \mathbf{1}_{\mathcal{U}_k}(N_i) \cdot N_i \quad (9)$$

where the indicator function $\mathbf{1}_{\mathcal{U}_k}$ is defined as: $\mathbf{1}_{\mathcal{U}_k}(N_i) = 1$ iff $N_i \in \mathcal{U}_k$.

For the complete universe we define:

$$\Pi_{\mathcal{U}}(\mu) := \sum_{k \in K} \Pi_{\mathcal{U}_k}(\mu) = \sum_{i=1}^n N_i \quad (10)$$

We define $\widetilde{\Pi}_{\mathcal{U}} := \text{dom} \circ \Pi_{\mathcal{U}}$ to obtain the set of object nets. We use the metaphor that $\widetilde{\Pi}_{\mathcal{U}}(\mu) = \{N_1, \dots, N_n\}$ is the *gene pool* of the marking μ .

Note, that for safe HORNETS we have no more object nets than system net places and therefore $|\widetilde{\Pi}_{\mathcal{U}}(\mu)| \leq |\Pi_{\mathcal{U}}(\mu)| \leq |\widehat{P}|$, i.e. the gene pool is bounded.

3.1. Adaption in the Reachability Graph

The set of reachable object nets of EH , defined as:

$$\mathcal{U}(EH) := \mathcal{U}(\mu_0) := \bigcup_{\mu \in RS(\mu_0)} \tilde{\Pi}_{\mathcal{U}}(\mu) \quad (11)$$

describes the universe of reachable genes, but it does not contain any information about the adaption *dynamics* at all. So, we look at the reachability graph. But, the RG of a HORNET is usually too large to analyse, since the reachability problem requires exponential space for safe EHORNETS [22]. Fortunately, the RG contains much information that is not relevant when studying the adaption process. We like to define an abstracted version of the RG that describes the evolution of the gene pool (as specified by the HORNET).

As we are interested in the adaption dynamics of object nets we start with the reachability graph and abstract from “irrelevant” events. Roughly speaking, an event is considered as irrelevant whenever the set of object nets $\tilde{\Pi}_{\mathcal{U}}(\mu)$ doesn’t change when firing the event (i.e. the HORNET is stuttering with respect to the set of object nets).

Definition 3. *The marking μ_{n+1} is \mathcal{U} -reachable in (V, E) from μ_1 whenever there is a sequence $\mu_1\mu_2 \cdots \mu_{n+1}$ such that for all $i \in \{1, \dots, n\}$ we have:*

$$\exists \theta_i : \mu_i \xrightarrow{\theta_i} \mu_{i+1} \wedge \tilde{\Pi}_{\mathcal{U}}(\mu_i) = \tilde{\Pi}_{\mathcal{U}}(\mu_{i+1})$$

Given $RG(EH) = (V, E)$ we define the following relations on reachable markings:

- $(\mu_1, \mu_2) \in R_1$ iff μ_2 is \mathcal{U} -reachable from μ_1 in (V, E) – and vice versa.
- $(\mu_1, \mu_2) \in R'_1$ iff μ_2 is \mathcal{U} -reachable from μ_1 in (V, E) .
- $(\mu_1, \mu_2) \in R_2$ iff μ_2 is \mathcal{U} -reachable from μ_1 in the undirected graph $(V, (E \cup E^{-1}))$.
- $(\mu_1, \mu_2) \in R_3$ iff $\tilde{\Pi}_{\mathcal{U}}(\mu_1) = \tilde{\Pi}_{\mathcal{U}}(\mu_2)$

In other words, R_1 relates markings that are strongly connected in the reachability graph via irrelevant events; for R'_1 the connection may hold in only one direction; and R_2 relates markings that are weakly connected via irrelevant events. Unlike the other relations, R_3 does not assume any reachability relation between the markings.

Obviously, the relations have the following properties:

- Lemma 1.**
- R_1 is reflexive, symmetric, and transitive.
 - R'_1 is reflexive and transitive, but in general not symmetric.
 - $R_1 = R'_1 \cap (R'_1)^{-1}$
 - R_2 is reflexive, symmetric, and transitive.
 - R_3 is reflexive, symmetric, and transitive.

Since R_1 , R_2 , and R_3 are equivalences we can use them to factor the RG, i.e. equivalence classes are condensed to a single node. All these graphs have $v_0 := [v_0]_{R_i}$ as a designated initial node.

Definition 4. Let $G = (V, E, v_0)$ be a graph with initial node and let $\approx \subseteq V^2$ be an equivalence on the vertices. The factorised graph is defined as $G_{/\approx} = (V', E', v'_0)$ with

$$V' = [V]_{\approx}, \quad E' = \{([\mu]_{\approx}, [\mu']_{\approx}) \mid (\mu, \mu') \in E\}, \quad \text{and} \quad v'_0 := [v_0]_{\approx}$$

Then, the abstracted graph $RG(EH)_{/R_i}$ where $i = 1, 2, 3$ describes the evolution of the gene pool that is specified by the HORNET.

The three relations leads to different kinds of abstraction due to the following inclusions:

Lemma 2 (Abstraction Hierarchy). $R_1 \subseteq R'_1 \subseteq R_2 \subseteq R_3$

Here, a finer equivalence leads to a larger factorised graph. With Lemma 2 we obtain that $RG(EH)_{/R_1}$ is closer to the original RG, while $RG(EH)_{/R_3}$ leads to a more abstract factorisation. More specifically:

- The graph $AG_1(EH) := RG(EH)_{/R_1}$ collapses areas in the reachability graph where the set of used object nets (i.e. the gene pool) doesn't change, i.e. it abstracts away from stuttering.
- The graph $AG_2(EH) := RG(EH)_{/R_2}$ collapses greater areas in the RG since it is sufficient that there is a weak connectivity of the markings in the RG.
- The graph $AG_3(EH) := RG(EH)_{/R_3}$ collapses all markings into one single vertice whenever the projection $\tilde{\Pi}_{\mathcal{M}}(\mu)$ is the same.

3.2. Key Values of Abstract Graphs

In the following, we identify suitable graph theoretic measures to give key values of the abstract graph (V, E, v_0) . Assume that $n = |V|$ and $m = |E|$. For more aspects of graph measures cf. [27, Chapter 17].

3.2.1. Number of Adaptivity Options

On a macroscopic scale the *average degree* $\frac{1}{n} \cdot \sum_{v \in V} \text{deg}(v)$ tells us about how many evolution options the system has. A higher value indicates that the system has many options to adapt.

Similarly, the *density* $\frac{m}{n(n-1)}$ tells us something about the specificity of the adaption process on a macroscopic scale, where a higher value indicates that the system has no specific direction to evolve.

Graph *transitivity* looks at density on a more microscopic scale. It is the probability that v 's neighbours are connected, too:

$$C_T := \frac{|\{(u, v, w) \mid (u, v), (v, w), (u, w) \in E\}|}{|\{(u, v, w) \mid (u, v), (v, w) \in E\}|} \quad (12)$$

Transitivity tells us how dense the adaption process is on a small scale, i.e. when considering triplets of nodes. A higher value indicates that there are many ways of achieving the same adaption state, while a lower value indicates that some adaption states are reachable via a more specific process, only

3.2.2. Measures for the Length of Adaption Processes

We are interested in measurements of the length of the adaption process. The path length tells us something about the convergence speed. Let $d(v \rightarrow v')$ denote the shortest path length from node v to v' . Then, the distances $d(v_0 \rightarrow v)$ describes how long the adaption dynamics is extended from initial node. A greater path length means a longer adaption dynamics.

The maximal distance from a given node is called its *eccentricity* $\epsilon(v)$:

$$\epsilon(v) := \max_{v' \in V} d(v \rightarrow v') \quad (13)$$

The eccentricity $\epsilon(v_0)$ tells us about the gene pool that is farthest away. We also consider the normalised variant $\epsilon(v_0)/n$.

3.2.3. Identifying central Gene Pools

We also identify the most “central” states in the dynamics, which are in some sense the discrete, graph-theoretical analogon of an attractor. There are several concepts of centrality in graph theory. Here, we use *betweenness centrality* $c_B(v)$ that is the probability for the shortest path between two nodes to go through that node v .

$$c_B(v) := \frac{1}{(n-1)(n-2)} \sum_{u, w \in V: u \neq v, w \neq v, u \neq w} \frac{N_{sp}(u \xrightarrow{v} w)}{N_{sp}(u \rightarrow w)} \quad (14)$$

Here, $N_{sp}(u \rightarrow w)$ is the number of shortest paths from node u to node w and $N_{sp}(u \xrightarrow{v} w)$ is the number of the shortest paths that go through node v .

We use the stochastic distribution of betweenness centrality $c_b(v)$ to identify which of the nodes/genes are very central in the dynamics. The variance indicates whether the distribution is tight around its average value: a higher variance indicates that some nodes are more central than others.

3.3. Directedness of Adaption

Assumed that adaption always leads to an improvement, the abstract graph would contain no cycles. In general, a system will allow for some regression for exploration issues – a try-and-error strategy. But it is reasonable that on larger time scales adaption will lead to an improvement. So, longer cycles are typically less common than short ones. When “ignoring” cycles, the abstract graph describes a partial order. Therefore, we have to find a maximal sub-graph that describes a partial order, i.e. a maximal DAG (directed acyclic graph). This sub-graph is interpreted as an underlying dynamics of *improvement* – when ignoring small phases of regression. We require that all the nodes remain reachable from the initial node v_0 in the DAG. So, we generate a maximal spanning DAG. The set of all such sub-graphs of $G = (V, E, v_0)$ being a DAG is:

$$DAGs(V, E, v_0) := \{D \mid D = (V, E_D, v_0) \text{ is a DAG} \wedge E_D \subseteq E \wedge \forall v \in V : (v_0, v) \in E_D^*\} \quad (15)$$

We are interested in a sub-DAG of maximal size:

$$m_0(G) := \max\{|E_D| \mid (V, E_D, v_0) \in \text{DAGs}(G)\} \quad (16)$$

In other words, $|E| - m_0(G)$ is the minimal number of edges to be deleted to obtain an acyclic graph from the abstracted reachability graph G .

Definition 5. *The degree of adaption directedness is $\text{dad}(G) := \frac{m_0(G)}{|E|}$.*

The degree of adaption directedness indicates, how much the dynamics is going into a certain direction (which can be interpreted as an improvement). When $\text{dad}(G)$ is close to 1, then the RG is acyclic, and the adaption dynamics goes (almost) strictly into one direction; while a $\text{dad}(G)$ close to 0 indicates an adaption process heavily running in cycles. Of course, we hope for a dynamics with $\text{dad}(G) \approx 1$, since they are easier to interpret.

Any subgraph $D = (V, E_D, v_0)$ of $G = (V, E, v_0)$ being a DAG with maximally many edges, i.e. $|E_D| = m_0(G)$, is a candidate for the underlying *improvement dynamics*.

$$\text{DAG}_{m_0}(G) := \{(V, E_D, v_0) \in \text{DAGs}(G) \mid m_0(G) = |E_D|\} \quad (17)$$

We use depth and width of all the DAGs $D \in \text{DAG}_{m_0}(V, E, v_0)$ as our central key measures:

- The *DAG-depth* is the largest *line*, i.e. a maximal clique w.r.t. **li**, which is the *dependence* relation of a DAG $D = (V, E, v_0)$ defined as $\mathbf{li} := E^* \cup (E^*)^{-1} \cup id_V$:

$$\text{depth}_D(D) := \max\{|L| \mid L \text{ is maximal li-clique}\} \quad (18)$$

A greater depth tell us that the system is very productive with respect to adaption, since it is going very straight into one direction.

- The *DAG-width* is the largest *cut*, i.e. a maximal independent set, or, alternatively, a maximal clique w.r.t. the *independence* relation $\mathbf{co} := \bar{\mathbf{li}} \cup id_V$:

$$\text{width}_D(D) := \max\{|C| \mid C \text{ is maximal co-clique}\} \quad (19)$$

A greater width tells us that for a given gene pool there is a large amount of options to adapt – depending on structural aspects we have abstracted from in the adaption graph.

- The depth-width ratio $\text{depth}(G)/\text{width}(G)$ tells us something about the number of adaption options given per productive evolution step.

We average the depth and width over all the DAGs in $\text{DAG}_{m_0}(V, E, v_0)$:

$$\text{depth}(G) := \frac{1}{|\text{DAG}_{m_0}(G)|} \sum_{D \in \text{DAG}_{m_0}(G)} \text{depth}_D(D) \quad (20)$$

Analogously for $\text{width}(G)$.

3.4. Relationship between exploration and exploitation

We can transform the DAG $D = (V, E, v_0)$ into a tree by using a different copy of a node for each path p from the initial node to this node. (The definition is unproblematic, since our DAGs have finitely many nodes, and therefore also only finitely many paths.)

Definition 6. Define the tree unfolding of a DAG $D = (V, E, v_0)$ as $tree(D) = (V_{tr}, E_{tr}, v_0)$ with

$$\begin{aligned} V_{tr} &= \{p \mid p = v_0 \dots v \text{ is a path in } D\} \\ E_{tr} &= \{(p, p') \mid p' = p \cdot v'\} \end{aligned}$$

The tree structure reveals information about the relationship between *exploration* of the adaption space, i.e. when the systems tests different options (and this accepts regression), and its *exploitation*, i.e. when the systems sticks to a pretty good option and tries to improve it with minor changes instead of evaluating some completely other option.

When the tree is growing the systems explores the adaption space. When the growth stagnates the system exploits the information gained. We try to capture this in the following measures: The adaption tree T has a different branching structure, i.e. less successors here and more over there; additionally, exploration and exploitation does not necessarily come in two successive phases, but in many, intertwined ones. So, we compare our tree with a very “tidy” reference tree, namely a tree that starts with the exploration phase followed by the exploitation phase. When the root is drawn at the top the tree has a \triangle -like shape, where the triangle describes the exploration (with a high branching degree) and the rectangle describes the exploitation (with no branching). A dynamics that has a maximal exploration leads to a tree that consists of the triangle part only.

Assume that the original tree T has height h_0 and l_0 leaves and n_0 nodes. Let h be the height of the exploration-triangle and h' the height of the exploitation-rectangle in the reference tree. Similarly, the triangle has n nodes and l leaves; the rectangle has n' nodes and l' leaves. As the rectangle has no branching, it has the same number of leaves as the triangle: $l = l'$. We want to have a \triangle -like tree that the same height and also the same number of nodes as the original tree: $h_0 = h + h'$ and $n_0 = n + n'$. We also require that the number of leaves does not change: $l_0 = l = l'$.

The new tree must have a (yet unknown) branching b such that $b^h = l = l_0$ in the \triangle -part of the \triangle -like shape. Thus the complete triangle then has $n = \frac{b^{h+1}-1}{b-1}$ nodes and $l = b^h$ leaves. The rectangle has as many leaves as the triangle, as we have no branching here, and $n' = h' \cdot b^h$ nodes. So, we obtain:

$$n_0 = n + n' = \frac{b^{h+1}-1}{b-1} + h' \cdot b^h \approx b^h + h' \cdot b^h = b^h(1 + h') = l_0(1 + h')$$

Therefore, the exploitation rectangle \square has the height $h' \approx \frac{n_0}{l_0} - 1$. From this we obtain the height h of the exploration triangle \triangle as $h = h_0 - h' \approx h_0 - \frac{n_0}{l_0} + 1$.

Let $D = (V, E, v_0)$ be a DAG and assume that its tree unfolding $T = tree(D)$ has l_0 leaves, n_0 nodes, and height h_0 . Then, the *exploitation rate* of $T = tree(V, E, v_0)$ is the ratio of the

rectangle height h' to the total height h_0 :

$$\chi(T) := \frac{h'}{h_0} \approx \frac{1}{h_0} \left(\frac{n_0}{l_0} - 1 \right) \quad (21)$$

Correspondingly, the *exploration rate* is given as $(1 - \chi(T))$.

We are interested in the corresponding exploration and exploitation rates in the adaption dynamics.

Definition 7 (exploration and exploitation rates). For the DAG D we define $\chi(D) := \chi(\text{tree}(D))$. For an abstract reachability graph G we define:

$$\chi(G) := \frac{1}{|\text{DAG}_{m_0}(G)|} \sum_{D \in \text{DAG}_{m_0}(G)} \chi(D) \quad (22)$$

These rates are used to characterise adaption dynamics: Systems with a high exploration rate could be considered as adventurous, while systems with a high exploitation rate could be considered as more conservative.

4. Case Study: Axelrod's Tournament

In Axelrod's Tournament [28] several agents are playing the well known prisoners' dilemma. If both agents cooperate (C) they both obtain a payoff, e.g. $r = 3$; If only one agent cooperate the defecting (D) gets a higher payoff of e.g. $t = 5$, while the cooperating one obtains nothing $s = 0$. If both agents defect they both obtain a small payoff, e.g. $p = 1$. In general, $t > r > p > s \geq 0$ is assumed.

	cooperate (C)	defect (D)
cooperate (C)	3,3	0,5
defect (D)	5,0	1,1

In the tournament version n agents are playing this game pairwise over several rounds. The agents are allowed to adapt their strategy over time. We assume that each agent remembers a finite history of games already played with a given opponent. Assuming a history of size 3 a history is $h = g_3 g_2 g_1 = (a_3, b_3)(a_2, b_2)(a_1, b_1)$ where $a_i, b_i \in \{C, D\}$. A strategy is therefore a function $s : \{C, D\}^6 \rightarrow \{C, D\}$. When describing strategies by automata each history $h \in \{C, D\}^6$ is a state and the action chosen for the current game is given by state changes.

Even for this very restricted history size we have $2^{2^6} = 2^{64}$ different strategies, i.e. the strategy space is far too large to be explored directly. The classical approach to adaption is genetic programming. The main idea is simple ([2, pg 80ff]): Each strategy evaluates its payoff so far. The higher the payoff the higher the number of offsprings. An offspring is generated via a cross-over mechanism, i.e. the two parents' histories are split at some fixed point and recombined in a cross-over fashion.

We generalise the setting: We assume that there is a *network* that connects the agents and only *connected* agents can play the game. We assume that a network also determines which

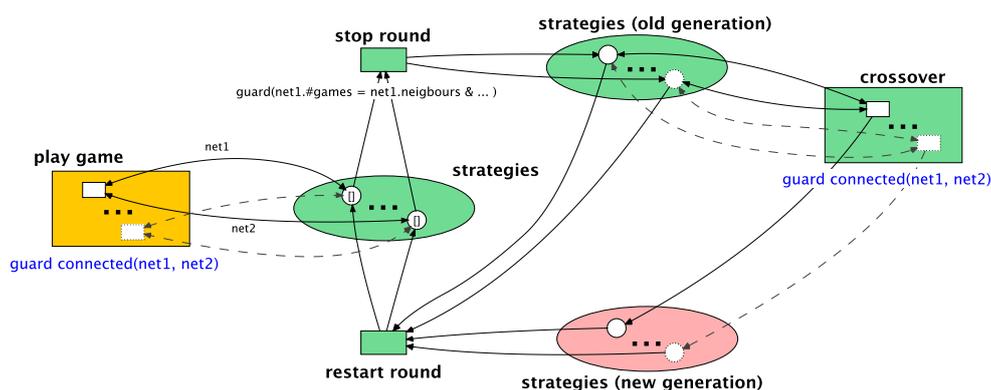


Figure 3: The eHORNET's System Net for the Tournament

agents may be chosen as parents when generating a cross-over. For simplicity we assume both networks to be equal.

We have specified this scenario as an eHORNET. Here, the object nets specify the agents, i.e. the strategy automata. The system net (shown in a simplified version in Fig. 3) specifies which agents are chosen for playing the game. It also specifies which agents are chosen in the adaption process for the cross-over. The system net is parametrised with the connection graph. The details of this case study can be found in [29].

adaption dynamics	Erdős-Rényi	Watts-Strogatz
average degree $\frac{1}{n} \cdot \sum_{v \in V} deg(v)$	3.42	3.14
density $\frac{m}{n(n-1)}$	0.13	0.26
transitivity C_T	0.07	0.00
eccentricity $\epsilon(v_0)$	4	3
variance of betweenness centrality $c_B(v)$	0.05	0.10
adaption directedness $dad(G)$	0.63	0.55
width(G)/depth(G)	7/4 = 1.75	4/3 = 1.33
rate of exploitation $\chi(G)$	0.20	0.25

Table 1

Summary of Key Measures of Adaption Processes for different Topologies

We study two different random graph topologies for the connection graph, namely a *Erdős-Rényi graph* [30] and a *Watts-Strogatz graph* [31] with a rewiring probability of $\beta = 0.01$ – both with the same number of nodes ($n = 28$) and the same number of edges ($m = 3n = 84$), modelled by the eHORNETS EH_1 and EH_2 . We have chosen these two graphs because they are both well known random graph structures. Moreover, the Erdős-Rényi graph is the special case of a Watts-Strogatz graph where $\beta = 1$. It is well known that for small values of β each pair of nodes is connected in a Watts-Strogatz graph via a quite short path of length $\propto \log(n)$, a fact that is also known as the *small world* property.

We generate the abstract adaption graphs $AG_3(EH_i)$, $i = 1, 2$ for the two topologies. The key measures of the two abstracted reachability graphs are given in Table 1. One can observe,

for example, that the Erdős-Rényi graph is more exploring than Watts-Strogatz graph as it has a higher average degree, a higher width/depth-ratio, and a lower exploitation rate $\chi(G)$. One may hypothesize that the reason for this lies in the *small world* property of the Watts-Strogatz, i.e. we have long-range connections and therefore every node is only some steps away from any other node, which is not the case for an Erdős-Rényi graph. Therefore, the Watts-Strogatz graph is faster in spreading good genes over the entire population.

5. Conclusion

In this contribution we developed key measures to compare adaption processes, usually adaption processes of variants of the same system. We used the nets-within-nets formalism of HORNETS as our formal model as they provide a natural description of what is adapted within the system: the set $\bar{\Pi}_{\mathcal{M}}(\mu)$ of net-types occurring in the current marking μ . Here, the system net describes the general system, while the marking's set of object net types describe the changing gene-pool. For the analysis we abstract from events in the HORNET that doesn't effect the gene pool and use network theoretical measures to describe a fingerprint of the adaption process.

Here, we abstracted from events that do not change the gene-pool. Currently, we are relaxing this even further by ignoring events that change the gene-pool only slightly, i.e. we use a distance ϵ between gene-pools as an additional parameter. In future work we also like to analyse the adaption dynamics via static properties of the Petri nets, like invariants etc. For most cases it may be sufficient to study an under- or an over-approximation of the state space.

References

- [1] M. Köhler-Bußmeier, Hornets: Nets within nets combined with net algebra, in: K. Wolf, G. Franceschinis (Eds.), International Conference on Application and Theory of Petri Nets (ICATPN'2009), volume 5606 of *Lecture Notes in Computer Science*, Springer-Verlag, 2009, pp. 243–262.
- [2] J. H. Holland, *Hidden Order: How Adaptation builds complexity*, Helix Books, 1995.
- [3] M. Köhler, H. Rölke, Properties of Object Petri Nets, in: J. Cortadella, W. Reisig (Eds.), International Conference on Application and Theory of Petri Nets 2004, volume 3099 of *Lecture Notes in Computer Science*, Springer-Verlag, 2004, pp. 278–297.
- [4] M. Köhler-Bußmeier, F. Heitmann, On the expressiveness of communication channels for object nets, *Fundamenta Informaticae* 93 (2009) 205–219.
- [5] R. Valk, Object Petri nets: Using the nets-within-nets paradigm, in: J. Desel, W. Reisig, G. Rozenberg (Eds.), Advanced Course on Petri Nets 2003, volume 3098 of *Lecture Notes in Computer Science*, Springer-Verlag, 2003, pp. 819–848.
- [6] I. A. Lomazova, Nested Petri nets – a formalism for specification of multi-agent distributed systems, *Fundamenta Informaticae* 43 (2000) 195–214.
- [7] D. Xu, Y. Deng, Modeling mobile agent systems with high level Petri nets, in: IEEE International Conference on Systems, Man, and Cybernetics'2000, 2000.
- [8] O. Kummer, *Referenznetze*, Logos Verlag, 2002.

- [9] K. Hiraishi, PN^2 : An elementary model for design and analysis of multi-agent systems, in: F. Arbab, C. L. Talcott (Eds.), *Coordination Models and Languages, COORDINATION 2002*, volume 2315 of *Lecture Notes in Computer Science*, Springer-Verlag, 2002, pp. 220–235.
- [10] M. A. Bednarczyk, L. Bernardinello, W. Pawlowski, L. Pomello, Modelling mobility with Petri hypernets, in: J. L. Fiadeiro, P. D. Mosses, F. Orejas (Eds.), *Recent Trends in Algebraic Development Techniques (WADT 2004)*, volume 3423 of *Lecture Notes in Computer Science*, Springer-Verlag, 2004, pp. 28–44.
- [11] C. Lakos, A Petri net view of mobility, in: *Formal Techniques for Networked and Distributed Systems (FORTE 2005)*, volume 3731 of *Lecture Notes in Computer Science*, Springer-Verlag, 2005, pp. 174–188.
- [12] I. A. Lomazova, K. M. van Hee, O. Oanea, A. Serebrenik, N. Sidorova, M. Voorhoeve, Nested nets for adaptive systems, in: *Application and Theory of Petri Nets and Other Models of Concurrency, Lecture Notes in Computer Science*, Springer-Verlag, 2006, pp. 241–260.
- [13] L. Cardelli, A. D. Gordon, G. Ghelli, Mobility types for mobile ambients, in: *Proceedings of the Conference on Automata, Languages, and Programming (ICALP'99)*, volume 1644 of *Lecture Notes in Computer Science*, Springer-Verlag, 1999, pp. 230–239.
- [14] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes, parts 1-2, *Information and computation* 100 (1992) 1–77.
- [15] W. Reisig, Petri nets and algebraic specifications, *Theoretical Computer Science* 80 (1991) 1–34.
- [16] K. Hoffmann, H. Ehrig, T. Mossakowski, High-level nets with nets and rules as tokens, in: *Application and Theory of Petri Nets and Other Models of Concurrency*, volume 3536 of *Lecture Notes in Computer Science*, Springer-Verlag, 2005, pp. 268 – 288.
- [17] I. Lomazova, Nested petri nets for adaptive process modeling, in: A. Avron, N. Dershowitz, A. Rabinovich (Eds.), *Pillars of Computer Science*, volume 4800 of *Lecture Notes in Computer Science*, Springer-Verlag, 2008, pp. 460–474.
- [18] G. Taentzer, Distributed graphs and graph transformation, *Applied Categorical Structures* 7 (1999) 431–462.
- [19] M. Köhler-Bußmeier, F. Heitmann, Safeness for object nets, *Fundamenta Informaticae* 101 (2010) 29–43.
- [20] M. Köhler-Bußmeier, F. Heitmann, Liveness of safe object nets, *Fundamenta Informaticae* 112 (2011) 73–87.
- [21] M. Köhler-Bußmeier, A survey on decidability results for elementary object systems, *Fundamenta Informaticae* 130 (2014) 99–123.
- [22] M. Köhler-Bußmeier, On the complexity of the reachability problem for safe, elementary Hornets, *Fundamenta Informaticae* 129 (2014) 101–116. Dedicated to the Memory of Professor Manfred Kudlek.
- [23] R. J. Lipton, The reachability problem requires exponential space, *Research Report 62*, Dept. of Computer science, 1976.
- [24] M. Köhler-Bußmeier, F. Heitmann, An upper bound for the reachability problem of safe, elementary hornets, *Fundamenta Informaticae* 143 (2016) 89–100.
- [25] M. Köhler-Bußmeier, Restricting Hornets to support adaptive systems, in: W. van der Aalst, E. Best (Eds.), *PETRI NETS 2017, Lecture Notes in Computer Science*, Springer-Verlag, 2017.

- [26] H. Ehrig, B. Mahr, *Fundamentals of algebraic Specification*, EATCS Monographs on TCS, Springer-Verlag, 1985.
- [27] H. Sayama, *Introduction to the Modeling and Analysis of Complex Systems*, Open SUNY Textbooks, 2015.
- [28] R. Axelrod, *The Evolution of Cooperation*, Basic Books, 1984.
- [29] M. Jankowski, P. B. Dang, J. Krukenberg, *Analyse evolutionärer Strategien*, Term Paper, HAW Hamburg, 2022.
- [30] P. Erdős, A. Rényi, *On random graphs*, *Publicationes Mathematicae* 6 (1959) 290–297.
- [31] D. J. Watts, S. H. Strogatz, *Collective dynamics of 'small-world' networks*, *Nature* 393 (1998) 440–442.