From Organisational Structure to Organisational Behaviour Formalisation[*]

Catholijn M. Jonker and Jan Treur

[1]Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, NL-1081 HV Amsterdam, The Netherlands
`Email: {jonker, treur}@cs.vu.nl`
`URL: http://www.cs.vu.nl/~{jonker, treur}`

Abstract To understand how an organisational structure relates to organisational behaviour is an interesting fundamental challenge in the area of organisation modelling. Specifications of organisational structure usually have a diagrammatic form that abstracts from more detailed dynamics. Dynamic properties of agent systems, on the other hand, are often specified in the form of a set of logical formulae in some temporal language. This paper addresses the question how these two perspectives can be combined in one framework. It is shown how for different aggregation levels and other elements within an organisation structure, sets of dynamic properties can be specified. Organisational structure provides a structure of (interlevel) relationships between these multiple sets of dynamic properties. Thus organisational structure is reflected in formalisation of the dynamics of organisational behaviour. As an illustration, for the AGR organisation modelling approach it is shown how a formal foundation can be obtained for integrated specification of both structure and behaviour of an organisation.

1 Introduction

Societies are characterised by complex dynamics involving interaction between large numbers of actors and groups of actors. If such complex dynamics takes place in an completely unstructured, incoherent manner, any actor involved has not much to rely on to do prediction, and is not able to function in a knowledgeable manner. This has serious disadvantages, which is a reason why in history within human societies organisational structure has been developed as a means to manage complex dynamics. Organisational structure provides co-ordination of the processes in such a manner that a process or agent involved can function in a more adequate manner. The dynamics shown by a given organisational structure are much more dependable than in an entirely unstructured situation. It is assumed that the organisational structure itself is relatively stable, i.e., the structure may change, but the frequency and scale of change are assumed low compared to the more standard dynamics through the structure. Within the field of Organisation Theory such organisational structures regulating societal dynamics are studied; e.g., [29], [37]. In summary, organisational *structure* is used to obtain *dynamics* (or organisational *behaviour*) of a desired type. For further analysis a crucial issue here is how exactly a formalisation of organisational structure is able to affect formalisation of organisational dynamics.

This implies in particular that as a foundation for an appropriate approach to organisation modelling, both the *structural* aspects and the *dynamic* aspects and their *relation* have to be

---

[*] A preliminary, shorter version of this work was presented at the Third International Workshop on Multi-Agent Based Simulation, MABS'02.

formalised in an appropriate manner. Multi-agent or organisation modelling approaches have been developed in three manners.

Firstly, informal or semi-formal graphical representations of the organisational structure have been developed; i.e., pictures with boxes and arrows; e.g., [37]. Such organisation models, although they provide a detailed account of the organisation structure, remain on a rather abstract level. In particular they do not give indications how the more detailed dynamics takes place; it does not specify how these structures relate to dynamics.

Secondly, within the area of Computational Organisation Theory and Artificial Intelligence, a number of organisation modelling approaches have been developed to simulate and analyse dynamics within organisations in society; e.g., [39], [32], [38], [12], [16], [23], [26]. Some of these approaches explicitly focus on modelling organisational structure, abstracting from the detailed dynamics. Other approaches put less emphasis on organisational structure but focus on the dynamics in the sense of implementing and experimenting with simulation models. Often these simulation models are based on some implementation environment and not specified in an implementation-independent manner using a formally defined conceptual language. The Strictly Declarative Modelling Language SDML [38] (see also [9]) , and the use of the agent-oriented modelling approach DESIRE in social simulation as presented in [8] are some of the few exceptions. Both modelling approaches focus on specification and simulation; however, they do not offer dedicated support for a specific type of organisational structure. Moreover, simulation of dynamics is the main purpose; not much formally defined support is offered for analysis of dynamics, such as checking whether a given simulation or empirical trace satisfies a given dynamic property.

Thirdly, temporal modelling is one of the dominant approaches for specification and analysis of dynamic properties in agent systems in general; e.g., [1], [9], [15], [16], [17], [28], [33], [35], [38]; see also [18], [19] for an overview. One of the strong points in this area of research is the declarative modelling of simulation models, for example based on the paradigm of Executable Temporal Logic [1]. Also formal analysis (verification) can be supported using this type of formalisation; e.g., [9], [16], [33], [35]. However, the temporal languages or logics usually adopted do not have means to explicitly reflect organisational structure in an organisational behaviour specification.

The Agent/Group/Role (AGR) approach (previously called Aalaadin) introduced in [12] is an example of an approach initially focussing on organisational structure, abstracting from the details of the dynamics. However, [13] and [14] are some first steps to relate specifications of dynamic properties to the organisational structure provided by AGR. In [1] the MOCA system presented combines the AGR model with Madkit into a platform, based on a theoretical foundation. This paper shows how dynamics of the organisational structure itself can be modelled: agents that can dynamically create, join, or quit groups.

This paper presents a formal foundation that can be used to develop an organisation modelling approach that takes into account organisation structure within organisation behaviour. First it is explored in more detail (Section 2) how *organisational structure* can be specified based on a formally defined foundation. Section 3 addresses how, given a formalisation of organisational structure, *sets of dynamic properties* can be associated to each aggregation level and element within this structure. These dynamic properties can be used for simulation (especially when expressed in executable format) and formal analysis of empirical or simulated traces.

As different parts or aggregation levels are structurally related within an organisational structure, a next question in the context of the relation between organisation structure and dynamics is how the associated sets of dynamic properties can be related accordingly. To this end, in Section 4, as part of an analysis of how organisational structure relates to organisational dynamics, *logical interlevel relationships between sets of dynamic properties* of different elements or aggregation levels within an organisational structure are described. Finally, for realisation of an organisation, requirements can be specified on *agents allocated to roles* within an organisation model (Section 5). The paper provides a generic foundation for integrated specification languages covering both structure and dynamics. However, the specific choice of such a language (for example, also using semi-formal and graphical elements) is left open. As an illustration two example languages are discussed: TTL (Temporal Trace Language) and an extension of LTL (Linear Time Logic). The foundational approach is illustrated for the AGR organisation modelling approach, but has a wider applicability.

2 Specification of Organisation Structure

This section presents an approach to a foundation for the specification of organisation *structure*. Organisation structure is often depicted in diagrammatic form (for example, as kind of labelled graph; e.g., see Figures 1 and 2) consisting of different aggregation levels and different types of elements within the organisation (such as roles, groups, interactions), and relationships between these elements. A suitable formalisation approach for such structure descriptions is the notion of semantic structures (or models, in terms of logic) for many-sorted predicate logic; e.g., [36]. These structures will be denoted by tuples

$$< S_1, \dots S_n; R_1, \dots R_p; F_1, \dots, F_q >$$

where S_i are sorts, R_j relations over sorts, and F_k functions on sorts. This formalisation approach is adopted as a foundation for specification of organisational structure, and illustrated for AGR organisation structures.

Within the Agent/Group/Role or AGR organisation modelling approach [12], an organisation structure is described at three aggregation levels: the *organisation* consists of a set of *groups*, and each group consists of the *roles* in that group. Furthermore, *connections* between roles and between groups are possible; see Figure 1. Here the smaller ovals indicate roles and bigger ovals groups. Connections are indicated by the two types of arrows (dashed indicates an intergroup interaction, not dashed indicates a transfer). To indicate which role belongs to which group is depicted by drawing the smaller role oval within the bigger group oval. Moreover the organisation is *realized* by *agents* fulfilling roles;

As an example a factory is considered that is organised at the highest aggregation level according to two divisions: *division A* that produces certain components and *division B* that assembles these components to (composite) products. At one aggregation level lower the division A is organised according to two departments: *department A1* (the work planning department for division A) and *department A2* (component production department). Similarly, division B is organised according to two department roles: *department B1* (for assembly work planning) and *department B2* (product production department).

The two divisions are modeled as *groups* (depicted by the larger ovals), with the departments as their *roles* (depicted by smaller ovals within larger ones). A third group, the Connection Group C, models the communication between the two divisions. This group consists of the two *roles* ‘division A representative’ and ‘division B representative’. *Intergroup role interactions* (depicted by pairs of dotted lines) are modeled between the role ‘department A1’ in the division A group and the role ‘division A representative’ within the connection group, and between the role ‘department B1’ in the division B group and the role ‘division B representative’ within the connection group. *Intragroup role transfers* model communication between the two roles within each of the groups (depicted by the arrows).

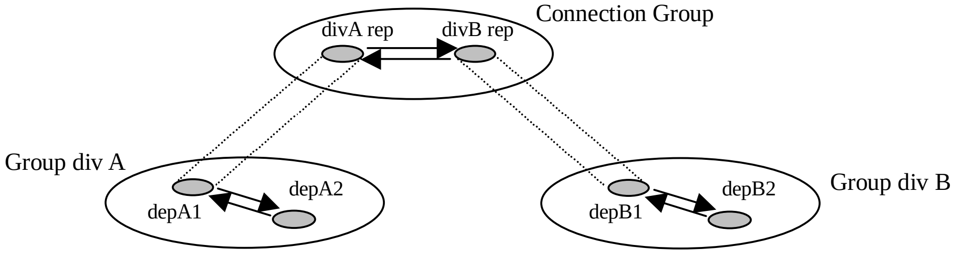

Fig. 1. An example AGR organisation structure

Connections have destination roles (indicated by the arrow points) and source roles (no pointing). Based on the semantic structures of many-sorted predicate logic a more precise formal definition is the following.

Definition 1 (AGR Organisation Structure)
Let IDENT be the set of all identifiers, and let Groups, Roles, Transfers, and Intergroup_interactions be subsets of IDENT which subsequently contain the names of groups, roles, transfers, and intergroup interactions in the organisation. An AGR *organisation structure* is defined by a tuple of sets and relations in the following manner:

AGROrg = < Groups, Roles, Intergroup_interactions, Transfers,
role_in, source_of_interaction, destination_of_interaction,
source_of_transfer, destination_of_transfer >

Each group involves a set of roles. The relationship

role_in: Roles × Groups

defines which role is in which group. The source and destination roles that an intergroup interaction connects are specified by

source_of_interaction, destination_of_interaction: Roles × Intergroup_interactions

Furthermore, transfers relate source roles to destination roles. The relationships

source_of_transfer, destination_of_transfer: Roles × Transfers

specify the roles a transfer connects. It is demanded that source and destination of a transfer belong to the same group:

$\forall$sr, dr $\in$ Roles, $\forall$ t $\in$ Transfers:

$$(\text{source\_of\_transfer}(sr, t) \wedge \text{destination\_of\_transfer}(dr, t)) \Rightarrow \exists g \in \text{Groups}: (\text{role\_in}(sr, g) \wedge \text{role\_in}(dr, g))$$

The example organisation of Figure 1 has

Groups = {divA, divB, C},
Roles = {depA1, depA2, depB1, depB2, divArep, divBrep},
Intergroup_interactions = {iAC, iCA, iBC, iCB}
Transfers = {tA12, tA21, tB12, tB21},

An overview of the relationships depicted in Figure 1 is:

within divA , divB and C	*organisation level*
role_in(depA1, divA) role_in(depA2, divA)	source_of_interaction(depA1, iAC) destination_of_interaction(divArep, iAC)
role_in(depB1, divB) role_in(depB2, divB)	source_of_interaction(divArep, iCA) destination_of_interaction(depA1, iCA)
role_in(divArep, C) role_in(divBrep, C)	source_of_interaction(depB1, iBC) destination_of_interaction(divBrep, iBC)
source_of_transfer(depA1, tA12) destination_of_transfer(depA2, tA12)	source_of_interaction(divBrep, iCB) destination_of_interaction(depB1, iCB)
source_of_transfer(depA2, tA21) destination_of_transfer(depA1, tA21)	
source_of_transfer(depB1, tB12) destination_of_transfer(depB2, tB12)	
source_of_transfer(depB2, tB21) destination_of_transfer(depB1, tB21)	
source_of_transfer(divArep, tC12) destination_of_transfer(divBrep, tC12)	
source_of_transfer(divBrep, tC21) destination_of_transfer(divArep, tC21)	

Note that, for simplicity, no difference is made between a role and role instances that inherit properties of the role. When desired, within a specific language developed on the basis of the formalisation presented here such a difference can be made. Furthermore, intergroup interactions are defined between two roles; this can easily be generalised for intergroup interactions involving more than two roles.

For all i ∈ Intergroup_interactions (resp. t ∈ Transfers or g ∈ Groups), let involved_roles(i) (resp. involved_roles(t), and involved_roles(g)) denote the set of all roles that are involved in interaction i (resp. transfer t or group g):

involved_roles(i) = { r ∈ Roles | destination_of_interaction(r, i)∨ source_of_interaction(r, i) }
involved_roles(t) = { r ∈ Roles | destination_of_transfer(r, t)∨ source_of_transfer(r, t) }
involved_roles(g) = { r ∈ Roles | role_in(r, g) }

3 Specification of Organisation Behaviour Based on Organisation Structure

After a foundation of an organisation structure has been defined, foundations for specification of dynamic properties in an organisation are addressed. The aim is not only to cover simple types of dynamics, such as simple reactive behaviour, but also more complex dynamics. For specification of more dynamic properties, often temporal logical languages are used; such language have no internal structuring (other than the manner in which formulae can be formed by logical connectives).

3.1 From Organisation Structure to Sets of Dynamic Properties for Organisation Behaviour

The challenge here is to incorporate somehow the organisational structure within the logical description of the organisation's internal dynamics. Just specifying one logical theory as a set of dynamic (temporal) properties describing the behaviour of the organisation would not be fully satisfactory, as in that case the organisational structure is lost in the dynamics specification or at least remains implicit. To capture the organisation structure within the organisation behaviour description, at least the different aggregation levels and more in general the different elements of the organisation structure are to be addressed in an explicit manner. To this end the following approach is introduced:

- for each *element* within the organisational structure characterise its dynamics by a *specific set of dynamic properties*; this is addressed in Section 3
- based on *structural relations between elements* in an organisational structure, identify *relationships between the sets of dynamic properties* corresponding with these elements; this is addressed in Section 4

In general, the dynamics of an element within an organisation structure can be characterised by specification of dynamic properties expressing relationships of states of that element over time. For a role the concept 'state' needs to be defined both for the input and the output of the role. Since transfers and intergroup interactions are assumed to operate only on input and output states of roles, without having their own internal state, no further state is assumed for transfers and intergroup interactions. To define states the notion of state property is useful, which is expressed in terms of a state ontology. Moreover, the notion of trace as a sequence of states over a time frame is used to formalise dynamics.

Definition 2 (Ontology, State, Trace)
(a) A *state ontology* is a specification (in order-sorted logic) of a vocabulary, i.e., a signature. A state for ontology Ont is an assignment of truth-values {true, false} to the set At(Ont) of ground atoms expressed in terms of Ont. The *set of all possible states* for state ontology Ont is denoted by STATES(Ont).
(b) A fixed *time frame* T is assumed which is linearly ordered. A *trace* $\mathcal{T}$ over a state ontology Ont and time frame T is a mapping $\mathcal{T}$: T → STATES(Ont), i.e., a sequence of states $\mathcal{T}_t$ ($t \in T$) in STATES(Ont). The set of all traces over state ontology Ont is denoted by TRACES(Ont).

Depending on the application, it may be dense (e.g., the real numbers), or discrete (e.g., the set of integers or natural numbers or a finite initial segment of the natural numbers), or any other form, as long as it has a linear ordering.

Definition 3 (State Properties and Dynamic Properties)
Let Σ be a given set of state ontologies.
(a) The set of *state properties* STATPROP(Σ) is the set of all propositions over ground atoms expressed in the ontologies from Σ.
(b) Let L be a language for dynamic properties. The set of *dynamic properties* $\mathrm{DYNPROP_L}(\Sigma)$ is the set of formulae that can be formulated in language L with respect to traces based on the set of state ontologies Σ.

The subscript L is dropped, when no confusion is expected and when the usage of the set is irrespective of a choice of language. For the paper such a language L is assumed with semantic consequence relation $\models$. The approach is independent of the choice of this language. In the next subsection two examples of languages that can be used are discussed.

3.2 Two Example Languages for Dynamic Properties

From a philosophical perspective [18] considers two main streams in temporal logic: modal logic approaches to temporal logic (developed mainly within Computer Science; e.g., [9], [15], [16], [17]), and predicate logic approaches to temporal logic (developed mainly within AI). In [19] different approaches in the latter stream are addressed in more detail. Two substreams distinguished are the use of additional temporal arguments within domain predicates, and the reification approach, where state properties are represented not by statements but by terms in the language, and predicates are used to express temporal structure over these term expressions; examples are event calculus [30], situation calculus [40] and the language TTL discussed below. In this approach part of the model theory is incorporated in the language. This reification approach to predicate logical temporal modelling is the approach adopted here. Two examples of languages that can be used to express dynamic properties that are considered in some more detail are:

- LTL (Linear Time Logic)
  This language is one from the stream based on modal logic approaches (cf. [15], [17], [9]); to make it suitable it is extended with operators that are specific for elements of the organisational structure or multi-agent system
- TTL (Temporal Trace Language)
  This language is one from the stream based on predicate logic approaches, and in particular the substream of reification approaches; e.g., [28], [16]

With in TTL, explicit reference is made to the part of the organisation within the state-function. Therefore, for TTL the set is defined relative to a set of identifiers to parts of the organisation structure and a set of ontologies. The set of identifiers can, for example, be the set of role names that can occur in the dynamic properties:

Definition 4 (DYNPROP$_{TTL}$)

The set DYNPROP$_{TTL}$(R, Σ) is defined inductively as follows:

- For all $r \in R$, for all parts P within O containing r, all traces $\gamma \in$ Traces(ONT(P)), for all $t \in T$, and for all state properties $s \in$ STATPROP(Σ), the formulae (here $\models$ is an infix predicate symbol):
  state(γ, t, r) $\models$ s
  state(γ, t, input(r)) $\models$ s
  state(γ, t, output(r)) $\models$ s
  are elements of DYNPROP$_{TTL}$(R, Σ).
- If $\varphi \in$ DYNPROP$_{TTL}$(R, Σ) and $\psi \in$ DYNPROP$_{TTL}$(R, Σ), then also
  $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$, $\varphi \Rightarrow \psi \in$ DYNPROP$_{TTL}$(R, Σ).
- Quantification is allowed in the normal way (e.g., over time, traces, parts and state properties).
- Mathematical operations and equations can be used in the normal way.

Typically, in terms of TTL, each role dynamic property φ of role r contains at least one reference to both the terms input(r) and output(r). For a linear temporal logic, this entails that the role name is used as index for some modal operator within the property. Furthermore, the properties only refer to r and to concepts from the ontology used in the interfaces of r.

Next, consider the linear time temporal logic LTL with the usual modal operators H (always in the past), P (at some time in the past), C (currently), X (next), F (at some time in the future), G (always in the future). The modal operators can be extended with a temporal parameters constraining the operator; e.g., $F^{<10}(\mathrm{prop})$ means that property prop will hold somewhere in the future before 10 time units have passed. The expression $F^{10}(\mathrm{prop})$ means that property prop will hold exactly after 10 time units. The modal operators can be indexed with either a group or a role name to indicate which part of the information state is meant. For example, $C_r(\mathrm{finished})$ means that currently in the information state of the role r the proposition answer holds. For a formalization of compositional multi-agent systems using temporal multi-epistemic logic, see (Engelfriet, Jonker & Treur, 2002).

Definition 5 (DYNPROP$_{LTL}$)

The set $\mathrm{DYNPROP}_{LTL}(R, \Sigma)$ is defined inductively as follows:

- For all $r \in R$, for all time constraints tc, for all propositions $s \in \Sigma$, for all modal operators M, the formula $M_r^{tc}(s)$ is an element of $\mathrm{DYNPROP}_{LTL}(R, \Sigma)$.
- If $\varphi \in \mathrm{DYNPROP}_{LTL}(R, \Sigma)$ and $\psi \in \mathrm{DYNPROP}_{LTL}(R, \Sigma)$, then also $\varphi \wedge \psi,\ \varphi \vee \psi,\ \neg\varphi.\ \varphi \Rightarrow \psi \in \mathrm{DYNPROP}_{LTL}(R, \Sigma)$.

In Table 1 some simple examples of dynamic properties in the two languages are shown.

Informal Dynamic Properties	**Formal Dynamic Properties in DYNPROP$_{TTL}$ and DYNPROP$_{LTL}$**
Role property of r1: Requests received by r1 are answered by r1 within 10 seconds.	In TTL: $\forall t\ [\ \mathrm{state}(\gamma, t, \mathrm{output}(r1)) \models \mathrm{request}$ $\Rightarrow\ \exists t': t \leq t' \leq t + 10 \wedge \mathrm{state}(\gamma, t', \mathrm{output}(r1)) \models \mathrm{answer}\ ]$ In LTL: $C_{r1}(\mathrm{request}) \Rightarrow F_{cl2}^{\leq 10}(\mathrm{answer})$
Transfer property for transfer between r1 *and* r2: Information available on the output of r1 arrives at the input of r2 within 1 second.	In TTL: $\forall t\ [\ \mathrm{state}(\gamma, t, \mathrm{output}(r1)) \models \mathrm{answer}$ $\Rightarrow \exists t': t \leq t' \leq t+1 \wedge \mathrm{state}(\gamma, t', \mathrm{input}(r2)) \models \mathrm{answer}\ ]$ In LTL: $C_{r1}(\mathrm{answer}) \Rightarrow X_{r2}(\mathrm{answer})$
Intergroup role interaction I1: Requests received at its input by r1 in group g1 will become available within group g2 on the output of role r2 within 1 second.	In TTL: $\forall t\ [\ \mathrm{state}(\gamma, t, \mathrm{input}(r1)) \models \mathrm{request}$ $\Rightarrow \exists t': t \leq t' \leq t+1 \wedge \mathrm{state}(\gamma, t', \mathrm{output}(r2)) \models \mathrm{request}\ ]$ In LTL: $C_{r1}(\mathrm{request}) \Rightarrow X_{r2}(\mathrm{request})$
Relative Adaptivity: 'Experience leads to quality': The more intensively requests have been handled by r, the higher the quality.	In TTL: $[\forall w1, w2\ \mathrm{state}(\gamma 1, 0, \mathrm{output}(r)) \models \mathrm{has\_level}(\mathrm{quality}, w1)\ \&$ $\mathrm{state}(\gamma 2, 0, \mathrm{output}(r)) \models \mathrm{has\_level}(\mathrm{quality}, w2) \Rightarrow w1 \leq w2\ ]\ \&$ $[\ \forall t\ [\ \mathrm{state}(\gamma 1, t, \mathrm{input}(r)) \models \mathrm{has\_intensity}(\mathrm{requests}, v1)\ \&$ $\mathrm{state}(\gamma 2, t, \mathrm{input}(r)) \models \mathrm{has\_intensity}(\mathrm{requests}, v2) \Rightarrow v1 \leq v2\ ]$ $\Rightarrow\ \forall t\ [\forall w1, w2\ \mathrm{state}(\gamma 1, t, \mathrm{output}(r)) \models \mathrm{has\_level}(\mathrm{quality}, w1)\ \&$ $\mathrm{state}(\gamma 2, t, \mathrm{output}(r)) \vert= \mathrm{has\_level}(\mathrm{quality}, w2) \Rightarrow w1 \leq w2\ ]\ ]$ It is not possible to express this in LTL.

Table 2. Example Dynamic Properties in TTL and LTL

3.3 Organisation Dynamics for an AGR Organisation Structure

In order to characterise the dynamics within an organisation, dynamic properties have to be specified for each of the aggregation levels and, more specifically, for the elements of an organisation structure: for the AGR modelling approach for each role, each transfer, each group, each intergroup interaction, and for the organisation as a whole. A specification of the dynamics requires a specification of the state ontologies used (for expressing state properties) for input states and output states of roles. The specifications of the dynamic properties are based on the given state ontologies.

Definition 6 (AGR Organisation Dynamics)

Let ONT be a set of (state) ontologies and O an AGR organisation structure defined by

O = < Groups, Roles, Transfers, Interactions,
role_in, source_of_transfer, destination_of_transfer,
source_of_interaction, destination_of_interaction >.

The dynamics of the AGR organisation O over ONT is formalised by a tuple as follows:

AGRDyn = < O, role_input_ontologies, role_output_ontologies,
role_dynproperties, transfer_dynproperties, interaction_dynproperties >

where

role_input_ontologies, role_output_ontologies: Roles $\rightarrow \wp$(ONT)

ontologies for input resp. output of each role r

role_dynproperties: Roles $\rightarrow \wp$(DYNPROP(ONT))

dynamic properties for each role r

transfer_dynproperties: Transfers $\rightarrow \wp$(DYNPROP(ONT))

dynamic properties for each transfer t

group_dynproperties(g) Groups $\rightarrow \wp$(DYNPROP(ONT))

dynamic properties for each group g

intergroup_interaction_dynproperties: Intergroup_Interactions $\rightarrow \wp$(DYNPROP(ONT))

dynamic properties for each intergroup interaction i.

organisation_dynproperties: {O} $\rightarrow \wp$(DYNPROP(ONT))

dynamic properties for the organisation O.

Thus the dynamics of the AGR organisation O is specified by: ontology sets for each role r, sets of dynamic properties for each transfer t, group g and intergroup interaction I, and the organisation as a whole. For these mappings the constraints C1 … C5 listed below are assumed to be fulfilled.

C1 Role dynamic properties

Role dynamic properties relate input to output of that role:

$\forall r \in$ Roles: role_dynproperties(r) $\subseteq$ DYNPROP(r, ONT(r))

For example, the gossip role behaviour: ‘whenever somebody tells you something, you will tell it to everybody else’ is expressed in terms of input of the role leading to output of the role in a reactive manner. An example relating to Figure 1:

DP(depA1) Progress Information Generates Planning in depA1

If within division A department A1 receives progress information on component production, then an updated planning will be generated by department A1 taking this most recent information into account.

C2 Transfer dynamic properties

Transfer properties relate output of the source roles to input of the destination roles:

$$\forall t \in \text{Transfers: transfer\_dynproperties}(t) \subseteq$$

$$\text{DYNPROP}(\text{ involved\_roles}(t) ,\ \bigcup\{ \text{role\_output\_ontologies}(r) \mid \text{source\_of\_transfer}(r, t) \} \cup \bigcup\{ \text{role\_input\_ontologies}(r) \mid \text{destination\_of\_transfer}(r, t) \})$$

Typically, such sets contain properties like, information is indeed transferred from source to destination, transfer is brought about within x time, arrival comes later than departure, and information departs before other information also arrives before that other information.

C3 Group dynamic properties

Group dynamic properties relate input and/or output of roles within a group.

$$\text{group\_dynproperties}(G) \subseteq \text{DYNPROP}(G,\ \text{ONT}(G))$$

An example of a group property is: "if the manager asks anyone within the group to provide the secretary with information, then the secretary will receive this information". A special case of a group property is an *intragroup interaction* relating the outputs of two roles within a group. A typical (informal) example of such an intragroup interaction property is: "if the manager says 'good afternoon', then the secretary will reply with 'good afternoon' as well". Other examples may involve statistical information, such as "3 out of the 4 employees within the organisation never miss a committed deadline". Example relating to Figure 1 for division A:

DP(A) A Progress Information Generation

This property is the conjunction of the following properties.

DP1(A) Initial A Progress Information Generation

Department A1 receives initial progress information on component production processes, involving already available components.

DP2(A) Subsequent A Progress Information Generation

Within the division A group, for any component production planning generated by department A1, incorporating a specific required set of components, progress information on the production of these components will be received by department A1.

C4 Intergroup interaction dynamic properties

Intergroup interaction properties relate the input of the source role to the output of the destination role:

$$\forall i \in \text{Intergroup\_interactions:}$$

$$\text{intergroup\_interaction\_dynproperties}(i) \subseteq \text{DYNPROP}(\text{ involved\_roles}(i) ,\ \bigcup\{ \text{role\_input\_ontologies}(r) \mid \text{source\_of\_interaction}(r, i) \} \cup \bigcup\{ \text{role\_output\_ontologies}(r) \mid \text{destination\_of\_interaction}(r, i) \})$$

For example, a project leader is asked by one of the project team members (input of role 'project leader' within the project group) to put forward a proposal in the meeting of project leaders (output of role 'member' within the project leaders group). An example relating to Figure 1:

Intergroup Role Interaction between A and C: IrRI(A, C)
For the connectivity between the groups A and C, the following intergroup role interaction properties are considered, one from A to C, and one from C to A.

IrRI(depA1, divArep) Progress Information Provision A to B
If within division A progress information on component production is received by department A1, then within the connection group this will be communicated by the division A representative to the division B representative.

IrRI(divArep, depA1) B Progress Information Incorporation by A
If within the connection group the division A representative receives information from the division B representative on which components are needed, then within division A a component production planning will be generated by department A1 taking these into account.

C5 Organisation dynamic properties

Organisation dynamic properties relate to input and/or output of roles within the organisation.

$$\text{organisation\_dynproperties}(G) \subseteq \text{DYNPROP}(O,\ \text{ONT}(O))$$

A typical (informal) example of such a property is: “if within the organisation, role A promises to deliver a product, then role B will deliver this product”. An example relating to Figure 1:

DP(F) Overall Progress Notification
If a request for a product is made (by a client),
then progress information will be provided (for the client).

The different types of dynamic properties all relate to different combinations of input and output. Table 1 provides an overview of these combinations. Note that with respect to simulation, the above dynamics definition can contain elements that are redundant: a smaller subset of dynamical properties can form an executable specification of the dynamics of an AGR type organisation. For example, on the basis of the role and transfer dynamic properties and intergroup interactions the organisation can be simulated. The group dynamic properties, including the intragroup role interaction properties, and the organisation properties should emerge in the execution, and testing for them can validate the model.

Table 1. Types of Dynamic Properties for an AGR Organisation Model

Dynamic Property Type	**Relating**	
Role r	Role r Input	Role r Output
Transfer from r1 to r2	Role r1 Output	Role r2 Input
Group G	Input or Output of roles in G	
Intragroup interaction	Role r1 Output	Role r2 Output
Intergroup interaction	Role r1 Input	Role r2 Output
Organisation	Input or Output of roles in O	

In order to make an executable organisation model the dynamical properties need to be chosen from the set of executable dynamical properties $\text{EXEDYNPROP} \subseteq \text{DYNPROP}$, for example Executable Temporal Logic [1], or the ‘leads to’ format presented in [29], [2].

4 Interlevel Relations between Dynamics at Different Aggregation Levels

An organisational structure defines relations between different elements in an organisation. In Section 3 the dynamics of these different elements were characterised by sets of dynamic properties. An organisational structure has the aim to keep the overall dynamics of the organisation manageable; therefore the structural relations between the different elements within the organisational structure have to impose somehow relationships or dependencies between their dynamics. In the introduction to their book Lomi and Larsen [32] emphasize the importance of such relationships. Organisations can be seen as adaptive complex information processing systems of (boundedly) rational agents, and as tools for control; central challenges are [32]:

- from the first view: 'given a set of assumptions about (different forms of) individual behaviour, how can the aggregate properties of a system be determined (or predicted) that are generated by the repeated interaction among those individual units?'
- from the second view: 'given observable regularities in the behaviour of a composite system, which rules and procedures - if adopted by the individual units - induce and sustain these regularities?'.

Both views and problems require means to express relationships between dynamics of different elements and different levels of aggregation within an organisation. In [32] two levels are mentioned: the level of the organisation as a whole versus the level of the units. Also in the development of MOISE (cf. [16], [23], [26]) an emphasis is put on relating dynamics to structure. Within MOISE dynamics is described at the level of units by the goals, actions, plans and resources allocated to roles to obtain the organisation's task as a whole. Specification of the task as a whole may involve achieving a final (goal) state, or an ongoing process (maintenance goals) and an associated plan specification.

As in our formalisation introduced in Section 3, dynamics are characterised by sets of dynamic properties for the respective elements of the organisation, the next step to be made is to identify how organisational structure determines (mathematically defined) relationships between these sets of dynamic properties for the different elements and aggregation levels within an organisation. Preferrably such relations between sets of dynamic properties would be of a logical nature; this would allow the use of logical methods to analyse, verify and validate organisation dynamics in relation to organisation structure. Indeed, in our approach presented below, logical relationships between sets of dynamic properties of elements in an organisation turn out an adequate manner to (mathematically) express such dynamic cross-element or cross-level relationships. This will be illustrated for the AGR approach. Within AGR organisation models three aggregation levels are involved:

- the organisation as a whole; the highest aggregation level.
- the level of a group
- the level of a role within a group

A general pattern for the dynamics in the organisation as a whole in relation to the dynamics in groups is as follows:

dynamic properties for the groups &
dynamic properties for intergroup role interaction
$\Rightarrow$ dynamic properties for the organisation

Moreover, dynamic properties of groups can be related to dynamic properties of roles as follows:

dynamic properties for roles &
dynamic properties for transfer between roles
$\Rightarrow$ dynamic properties for a group

The idea is that these are properties dynamically relating a number of roles within one group. To get the idea, consider the special case of an intragroup role interaction from role r1 to role r2, characterised by dynamic properties, that relate output of one role r1 to output of another role r2. Assuming that transfer from output of r1 to input of r2 is adequate and simply copies the information, this property mainly depends on the dynamics of the role r2.

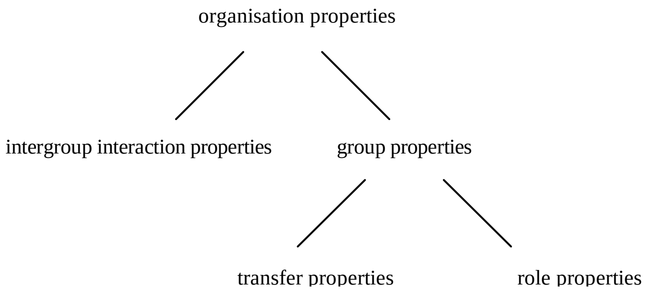

Fig. 2. Overview of interlevel relations between dynamic properties

Therefore in this case the relationship has the form:

dynamic properties for role r2 &
dynamic properties for transfer from role r1 to role r2
$\Rightarrow$ dynamic properties of intragroup interaction from r1 to r2

An overview of the logical relationships between dynamic properties at different aggregation levels is depicted as an AND-tree in Figure 2. The logical relationships put forward above can be formalised in the following manner. In Definition 5 below the following notations are used, where G is a group:

con(F) is the conjunction of all dynamic properties in a finite set F

role_dynproperties(G) = $\cup$ { role_dynproperties(t) | role_in(r, G) }

transfer_dynproperties(G) =

$\cup$ { transfer_dynproperties(t) | t: Transfer, involved_roles(t) $\subseteq$ involved_roles(G) }

transfer_dynproperties(O) = $\cup$ { transfer_dynproperties(t) | t: Transfer }

intergroup_interaction_dynproperties(O) =

$\cup$ { intergroup_interaction_dynproperties(i) | i: Intergroup_interaction }

Definition 7 (Role-Group and Group-Organisation Interlevel Relations)

Let AGRDyn be a model for the dynamics of an AGR organisation structure O.

(a) A *logical role-group interlevel relation* for group G is a logical statement of the form

$$DP_1 \ \& \ \dots \ \& \ DP_n \ \& \ TR \quad \Rightarrow \quad DP$$

where

DP	a group dynamics property for G, from	group_dynproperties(G)
DP_i	a role dynamics property or conjunction thereof from	role_dynproperties(r_i)
TR	a transfer dynamics property or conjunction thereof from	transfer_dynproperties(G)

The set of all role-group interlevel relations for G is denoted by RGIR(G); the union of all of them over all groups is denoted by RGIR.

(b) A *logical group-organisation interlevel relation* for O is a logical statement in of the form

$$DP_1 \ \& \ \dots \ \& \ DP_n \ \& \ TR \ \& \ IID \quad \Rightarrow \quad DP$$

where

DP	an organisation dynamics property from	org_dynproperties(O)

DP_i a group dynamics property or conjunction thereof from group_dynproperties(G_i)
TR a transfer dynamics property or conjunction thereof from transfer_dynproperties(O),
IID an intergroup interaction property or conjunction thereof from intergroup_interaction_dynproperties(O)

The set of all group-organisation interlevel relations is denoted by GOIR.

(c) A *logical interlevel relation assignment* for an AGR organisation model AGRDyn is a mapping

$$\text{interlevel\_relations:} \quad \{O\} \cup \text{Groups} \to \wp(\text{RGIR} \cup \text{GOIR})$$

such that the set interlevel_relations(O) consists of logical group-organisation interlevel relations for O (i.e., from GOIR), and for each group G the set interlevel_relations(G) consists of logical role-group interlevel relations for G (i.e., from RGIR(G)).

(d) The *standard logical interlevel relation assignment*

$$\text{standard\_interlevel\_relations:} \quad \{O\} \cup \text{Groups} \to \wp(\text{RGIR} \cup \text{GOIR})$$

for an AGR organisation model AGRDyn is the mapping defined by:

standard_interlevel_relations(G) is the set of logical relations

$$\{\, \text{con}(\text{role\_dynproperties}(G) \cup \text{transfer\_dynproperties}(G)) \Rightarrow DP \mid DP \in \text{group\_dynproperties}(G) \,\}$$

standard_interlevel_relations(O) is the set of logical relations

$$\{\, \text{con}(\text{role\_dynproperties}(O) \cup \text{transfer\_dynproperties}(O) \cup \text{intergroup\_interaction\_dynproperties}(O)) \Rightarrow DP \mid DP \in \text{organisation\_dynproperties}(O) \,\}.$$

Notice that this definition provides a formalisation for the type of relations that Lomi and Larsen [32] put forward as one of the challenges in organisation modelling (see above). Following the general implication pattern given above, the following specific relationships between dynamic properties at different aggregation levels can be established:

From group properties and intergroup role interaction properties to organisation properties

DP(A) & DP(B) & DP(C) & IrRI(F) ⇒ DP(F)

From intragroup role interaction properties to group properties

IaRI(A) & TRD(A) ⇒ DP(A)
IaRI(B) & TRD(B) ⇒ DP(B)
TRD(C) ⇒ DP(C)

From dynamic role properties to intragroup interaction properties

DP(depA1) & DP(depA2) & TRD(A) ⇒ IaRI(A)
DP(depB1) & DP(depB2) & TRD(B) ⇒ IaRI(B)

Such relationships between dynamic properties can be visualised in the form of an AND-tree; see Figure 3 (the names have been kept short to keep the picture concise).

Given these definitions notions such as 'valid' and 'complete' can be defined and a proposition can be formulated.

Definition 8 (Valid and Complete)
Let AGRDyn be a model for the dynamics of an AGR organisation structure O

(a) A logical interlevel relations assignment interlevel_relations is called *valid* if all involved logical interlevel relations are valid statements in the logic used, i.e., for any trace, if the antecedent holds, also the consequent holds.

(b) The model AGRDyn is called *grounded* if its standard logical interlevel relation assignment is valid.

(c) A logical interlevel relations assignment interlevel_relations is called *connected* if for any group G each group property occurring (in an antecedent) in interlevel_relations(O) also occurs (as a consequent) in interlevel_relations(G).
(d) A logical interlevel relations assignment interlevel_relations is called *complete* if for any group G each group property from group_dynproperties(G) occurs (as a consequent) in interlevel_relations(G) and each organisation property from organisation_dynproperties(O) occurs (as a consequent) in interlevel_relations(O).

Notice that the standard logical interlevel relation assignment is connected and complete. However, the antecedents used are not minimal in the sense that many irrelevant conjuncts may occur in them.

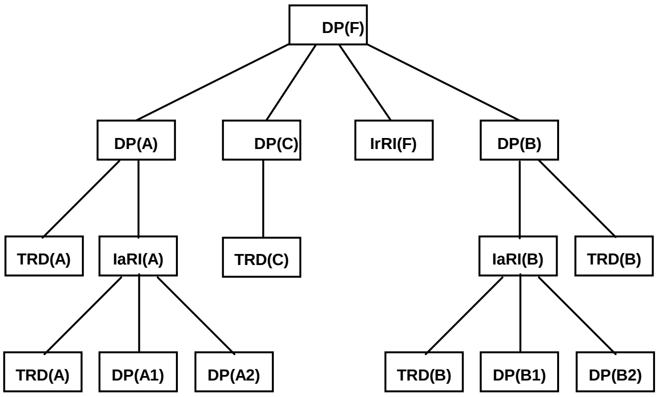

Fig. 3. Example interlevel relations between dynamic properties in the form of an AND-tree

Proposition

Let AGRDyn be a model for the dynamics of an AGR organisation structure O, and interlevel_relations an interlevel relation assignment. Moreover, let $\mathcal{T}$ be a trace that satisfies all dynamic properties in role_dynproperties(r) for all roles r, all transfer properties in transfer_dynproperties(t) for all transfers t, and all intergroup interaction properties in intergroup_interaction_dynproperties(i) for all intergroup interactions i.
(a) If the interlevel relation assignment interlevel_relations is valid, then trace $\mathcal{T}$ satisfies all dynamic group properties occurring in interlevel_relations(G) for any group G.
(b) If the interlevel relation assignment interlevel_relations is valid and complete, then the trace $\mathcal{T}$ satisfies all dynamic group and organisation properties in organisation_dynproperties(O) and group_dynproperties(G) for any group G.
(c) If a valid and complete logical interlevel relations assignment for AGRDyn exists, then AGRDyn is grounded.

This Proposition implies the following for an organisation model with a valid and complete interlevel relation assignment. If the properties for roles, transfers and intergroup interactions are in executable format, and used for simulation (e.g., based on the paradigm of Executable Temporal Logic), then a generated trace will satisfy these properties, and, hence by the Proposition satisfy all group and organisation properties as well. Among others, this gives means to validate (in the sense of falsification) an organisation model.

5 Organisation Realisation

In this section criteria are discussed when allocation of a set of agents to roles is appropriate to realize the organisation dynamics, illustrated for the AGR approach. One of the advantages of an organisation model is that it abstracts from the specific agents fulfilling the roles. This means that all dynamic properties of the organisation remain the same, independent of the particular allocated agents. However, the behaviours of these agents have to fulfil the dynamic properties of the roles and their interactions. The organisation model can be (re)used for any allocation of agents to roles for which:

- for each role, the allocated agent's behavior satisfies the dynamic role properties,
- for each intergroup role interaction, one agent is allocated to both roles and its behavior satisfies the intergroup role interaction properties, and
- the communication between agents satisfies the respective transfer properties.

Expressed differently, for a given allocation of agents to roles the following logical relationships between dynamic properties hold:

agent – role
from dynamic agent properties to dynamic role properties:

agent A is allocated to role r &
dynamic properties of agent A ⇒
dynamic properties of role r

agent – intergroup role interaction
from dynamic agent properties to dynamic intergroup role interaction properties:

agent A is allocated to roles r1 and r2 in different groups &
dynamic properties of agent A ⇒
dynamic properties of intergroup role interaction between r1 and r2

agent communication – role transfer
from dynamic agent communication properties to dynamic transfer properties:

agent A is allocated to role r1 and agent B to role r2 in one group &
dynamic properties of communication from A to B ⇒
dynamic properties of transfer from r1 to r2

Notice that in these relationships, if an agent is allocated to a role, it might be assumed that the input and output ontologies of the agent are subsets of the role's input and output ontologies, but this assumption is not necessary. However, to satisfy the logical relationships specified above, at least a relevant overlap between the agent's ontologies and the role ontologies will be needed; for more discussion on this issue, see [41]. Moreover, note that if in the last relationship, A = B (an agent fulfilling two roles in one group), then dynamic properties of communication from A to A are required, i.e., that A will receive (at its input state) what it communicates (at its output state): 'A hears itself talking'. The logical relationships can be depicted as in the extension of Figure 2 shown as Figure 4. The following formalised criteria are required for an organisation realisation.

Definition 9 (AGR Organisation Realisation)
To realise an organisation, agents fulfil roles in the groups in the AGR organisation modelling approach. A realisation of an AGROrg organisation structure is then:

AGRReal = <AGROrg, Agents, fulfils>

A set of agent names is given, so Agents ⊆ IDENT. The relationship

fulfils: Agents × Roles

specifies which roles an agent fulfils. The realisation dynamics specifies dynamic properties and ontologies for the agent input and output.

$\text{AGRRealDyn} = < \text{AGRReal}, \text{AGRDyn}, \text{agent\_input\_ontologies}, \text{agent\_output\_ontologies}, \text{agent\_dynproperties} >$

where

$\text{agent\_input\_ontologies}, \text{agent\_output\_ontologies}: \quad \text{Agents} \rightarrow \wp(\text{ONT})$

$\text{agent\_dynproperties}: \quad \text{Agents} \rightarrow \wp(\text{DYNPROP})$

The agent must have the required ontologies on input and output, as well as the required properties must be implemented for the roles it fulfils:

$\forall a \in \text{Agents}, \forall r \in \text{Roles}: \text{fulfils}(a, r) \Rightarrow$

$\text{role\_input\_ontologies}(r) \subseteq \text{agent\_input\_ontologies}(a) \wedge$
$\text{role\_output\_ontologies}(r) \subseteq \text{agent\_output\_ontologies}(a) \wedge \quad \text{agent\_dynproperties}(a) \models$
$\text{role\_dynproperties}(r)$

For all $i \in \text{Intergroup\_interactions}$ and for all $a \in \text{Agents}$, let $\text{involved\_in}(a, i)$ denote:

$\exists\, r \in \text{Roles}: \text{fullfils}(a, r) \wedge (\text{destination\_of\_interaction}(r, i) \vee \text{source\_of\_interaction}(r, i))$

The dynamical properties of intergroup interactions are assumed to be realised by having the same agent fulfil all roles in the intergroup interaction and having this agent process from its input to its output according to the intergroup interaction dynamical property; under this assumption, the intergroup interaction formally has to satisfy:

$\forall a \in \text{Agents}, i \in \text{Intergroup\_interactions}:$

$(\forall r \in \text{involved\_roles}(i): \text{fulfils}(a, r)) \Rightarrow \text{agent\_dynproperties}(a) \models \text{intergroup\_interaction\_dynproperties}(i)$

The transfers also need to be realised by successfulness of sending messages by the involved agents:

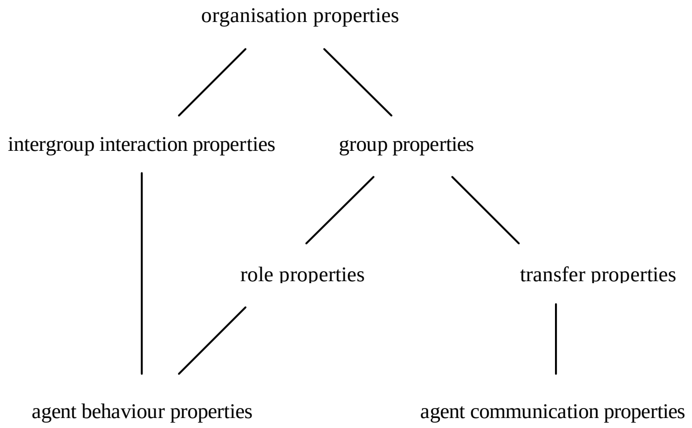

Fig. 4. Interlevel relations between dynamic properties for a realised organisation model

$\forall t \in \text{Transfers}, \; r1, r2 \in \text{Roles}:$
$\text{source\_of\_transfer}(r1, t) \wedge \text{fullfils}(a,r1) \wedge$
$\text{destination\_of\_transfer}(r2, t) \wedge \text{fullfils}(b,r2) \Rightarrow$
$\text{agent\_commproperties}(a, b) \models \text{transfer\_dynproperties}(t)$

For the example, the following allocation of agents agentA1, agentA2, agentB1, agentB2 to roles is possible:

agentA1 - depA1
agentB1 - depB1
agentA1 - divArep
agentA2 - depA2
agentB2 - depB2
agentB1 - divBrep

To realise the organisation model, for example agentA1 has to satisfy the following dynamic properties:

DP(agentA1)
If agent A1 receives progress information on component production,
then an updated planning will be generated by agent A1 taking this most recent information into account.

IrRI(agentA1)
If progress information on component production is received by agent A1,
then this will be communicated by agent A1 to agent B1
If agent A1 receives information on which components are needed,
then a component production planning will be generated by agent A1 taking these components into account.

Alternatively, if the roles in an intergroup interaction would not be fulfilled by one agent, but by several, this would create a mystery, since input to one agent creates output for another agent, even though the agents are not connected by any transfer since the roles they fulfil are from separate groups. This would suggest that the organisation structure is not complete. Therefore, in an AGR organisation model it is assumed that the roles in an intergroup interaction are fulfilled by one agent.

6 Discussion

Both in human society and for software agents, organisational structure is a means to make complex multi-agent dynamics manageable. To understand and formalize how exactly organisational structure constrains complex dynamics is a fundamental challenge in the area of organisational modelling. The framework combining structure and dynamics introduced in this paper provides support in addressing this challenge. Specification of organisation structure usually takes the form of pictorial descriptions, in a graph-like framework. These descriptions usually abstract from dynamics within an organisation. Specification of the dynamic properties of agent systems, on the other hand, usually takes place in a completely different conceptual framework; these dynamic properties are often specified in the form of a set of logical formulae in some temporal language.

This paper shows how these two perspectives can be combined in one framework. It is shown how for different types of elements within an organisation structure different sets of dynamic properties can be specified. Illustrated for [12]'s AGR organisation modelling approach, it has been shown how a foundation can be obtained for integrated specification of both structure and dynamic properties of an organisation. The organisational structure provides structural relations between different elements of the organisation; these relations induce logical relationships between the sets of dynamic properties for the different elements of the organisation. From the perspective of meta-level reasoning and representation (e.g., [6], [33], [42]), these relationships between sets of properties, based on organisational structure, can be considered a metatheory, reasoning about a collection of object level theories (or viewpoints or local contexts) that represent the behavioural theories of the different elements of the organisation; cf. [2], [3], [5],

[6], [20], [21], [42]. Such logical relationships make explicit dependencies between dynamic properties of parts of an organisation. The logical relationships express the kind of relations between dynamics of parts of an organisation, their interaction and dynamic properties of the organisation as a whole, which are indicated as crucial by (Lomi and Larsen, 2001) in their introduction.

The framework presented here contributes directly to the foundations of some of the criteria proposed by [9] (e.g., the criteria regarding the identification of constraints and flexible analysis) for an ideal social simulation language (ISSL), and it illustrates for some of the other criteria how such an ISSL can come about (e.g., the criteria regarding modeller' s intentions, compositionality, and emergence) in the context of the structure and dynamics of organizations. Furthermore, a shared basis of the work of [9] and the work presented here lies in the view that the different simulation processes need to be specified separately from the significant outcomes of the simulation, and that the processes that emerge from the simulation are typically more interesting than the resulting states of the simulation.

Also in MOISE and MOISE+ (cf. [16], [23], [26]), as for AGR, organisational structures are based on roles and groups and on links between roles. A difference with AGR is that tasks are taken into account explicitly and specified by goals, plans, resources and actions. These specifications can be considered one of the specific instantiations possible for the notion of dynamic property that is central in our foundational perspective as developed. In this case interlevel relations take the form of relations between tasks for roles and tasks at a higher aggregation level, for example for groups or (parts of) the organisation.

Having one framework that integrates the two perspectives, as well as the logical relationships between the two perspectives enables formal economic diagnostic analysis. Any simulation or empirical trace can be checked against a given dynamic property. Assuming the logical relationships, diagnosis of dysfunctioning within an organisation can be performed see, for example, [27].

For further work, this framework provides a basis for development of more specific organisation specification languages, covering both structure and dynamics. Such languages could make use of semiformal and graphical elements, and specification would be supported by software tools. The specification tools can mediate between a user and further software environments for simulation and analysis. Moreover, verification techniques and tools (e.g., [9], [16], [33]) can be considered for verification of dynamic properties and their interlevel relations.

Acknowledgements

The authors are grateful to Wouter Wijngaards who has contributed to some elements of this paper, in particular in Section 2.

References

1. Amiguet, M., Müller, J.-P., Baez-Barranco, J.-A., and Nagy, A., (2003), The MOCA Platform Simulated the Dynamics of Social Networks. In: J.S. Sichman, F. Bousquet, and P. Davidson (eds.), *Multi-Agent-Based Simulation II, Proc. of the Third International Workshop on Multi-Agent Based Simulation, MABS'02*. Lecture Notes in AI, vol. 2581, Springer Verlag, 2003.
2. Attardi, G., and Simi, M. (1984). Metalanguage and reasoning across viewpoints. In Tim O' Shea, (ed.), Proc. 6th European Conference on AI, ECAI'84. North-Holland, 1984, pp. 413-422
3. Attardi, G., and Simi, M. (1995). A formalization of viewpoints. Fundamenta Informaticae 23(3), pp. 149-173.

4. Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J., (2005). LEADSTO: a Language and Environment for Analysis of Dynamics by SimulaTiOn. In: Eymann, T., Kluegl, F., Lamersdorf, W., Klusch, M., and Huhns, M.N. (eds.), *Proc. of the Third German Conference on Multi-Agent System Technologies, MATES'05*. Lecture Notes in Artificial Intelligence, vol. 3550. Springer Verlag, 2005, pp. 165-178.
5. Bowen K.A. (1985). Meta-Level Programming and Knowledge Representation, New Generation Computing, 3 (1985), 359-383.
6. Bowen, K. and Kowalski, R., (1982). Amalgamating language and meta-language in logic programming. In: K. Clark, S. Tarnlund (eds.), *Logic programming*. Academic Press, 1982.
7. Barringer, H., Fisher, M., Gabbay, D., Owens, R., and Reynolds, M., (1996). The Imperative Future: Principles of Executable Temporal Logic, Research Studies Press Ltd. and John Wiley & Sons.
8. Brazier, F.M.T., Eck, P.A.T. van, and Treur, J., (2001). Modelling a Society of Simple Agents: From Conceptual Specification to Experimentation. Journal of Applied Intelligence, vol. 14, pp. 161-178.
9. Clarke, E.M. , O. Grumberg, D.A. Peled, (1999). *Model Checking*, MIT Press, Cambridge Massachusetts, London England, 1999.
10. Edmonds, B., (2003). Towards an Ideal Social Simulation Language. In: J.S. Sichman, F. Bousquet, and P. Davidson (eds.), *Multi-Agent-Based Simulation II, Proc. of the Third International Workshop on Multi-Agent Based Simulation, MABS'02*. Lecture Notes in AI, vol. 2581, Springer Verlag, 2003.
11. Engelfriet, J., Jonker, C.M., and Treur, J. (2002). Compositional Verification of Multi-Agent Systems in Temporal Multi-Epistemic Logic. Journal of Logic, Language and Information, **11**: 195-22.
12. Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis and design of organisations in multi-agent systems. In: Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98), IEEE Computer Society Press, pp. 128 -135.
13. Ferber, J., and Gutknecht, O. (1999). Operational Semantics of a role-based agent architecture. Proceedings of the 6th Int. Workshop on Agent Theories, Architectures and Languages (ATAL'1999). In: Jennings, N.R. & Lesperance, Y. (eds.) Intelligent Agents VI, Lecture Notes in AI, vol. 1757, Springer Verlag, 2000, pp. 205-217.
14. Ferber, J., Gutknecht, O., Jonker, C.M., Müller, J.P., and Treur, J., (2001). Organization Models and Behavioural Requirements Specification for Multi-Agent Systems. In: Y. Demazeau, F. Garijo (eds.), Multi-Agent System Organisations. Proceedings of the 10th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW' 01, 2001.
15. Fisher, M. (1994). A survey of Concurrent MetateM — the language and its applications. In: D.M. Gabbay & H.J. Ohlbach (eds.), Temporal Logic - Proceedings of the First International Conference, Lecture Notes in AI, vol. 827, pp. 480–505.
16. Fisher, M., and M. Wooldridge (1997). On the formal specification and verification of multi-agent systems. International Journal of Co-operative Information Systems, vol. 6, pp. 37–65.
17. Fisher, M., (2005). Temporal Development Methods for Agent-Based Systems, *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 10, 2005, pp. 41-66.
18. Galton, A. (2003). Temporal Logic. *Stanford Encyclopedia of Philosophy*, URL: http://plato.stanford.edu/entries/logic-temporal/#2.
19. Galton, A. (2006). Operators vs Arguments: The Ins and Outs of Reification. *Synthese*, vol. 150, 2006, pp. 415-441
20. Giunchiglia, E., Traverso, P., and Giunchiglia, F. (1993). Multi-context Systems as a Specification Framework for Complex Reasoning Systems, In: J. Treur and Th. Wetter (eds.), Formal Specification of Complex Reasoning Systems, Ellis Horwood, 1993, pp. 45-72.
21. Giunchiglia, F., Serafini, L. (1994). Multilanguage Hierarchical Logics (or: How we can do without modal logics), Artificial Intelligence, 65:29-70, 1994
22. Hannoun, M., Sichman, J.S., Boissier, O., and Sayettat, C. (1998). Dependence Relations between Roles in a Multi-Agent System: Towards the Detection of Inconsistencies in Organization. In: J.S. Sichman, R. Conte, and N. Gilbert (eds.), Multi -Agent Systems and Agent-Based Simulation (Proc. of the 1st. International Workshop on Multi-Agent Based Simulation, MABS' 98), Lecture Notes in Artificial Intelligence, vol. 1534, Springer -Verlag Berlin-Alemanha, 1998, pp. 169-182.
23. Hannoun, M., Boissier, O., Sichman, J.S., and Sayettat, C. (2000). MOISE: An organizational model for multi-agent systems. In: M. C. Monard and J. S. Sichman (eds.), Advances in Artificial Intelligence, (Proc. International Joint Conference 7th. Ibero-American Conference on Artificial

Intelligence (IBERAMIA' 00) and 15th. Brazilian Symposium on Artificial Intelligence (SBIA' 00), Atibaia, Brasil). Lecture Notes in Artificial Intelligence, vol. 1952, Springer-Verlag, Berlin, 2000, pp. 152-161.

24. Herlea Damian, D.E., Jonker, C.M., Treur, J., and Wijngaards, N.J.E. (2005). Integration of Behavioural Requirements Specification within Compositional Knowledge Engineering. *Knowledge-Based Systems Journal*, vol. 18, 2005, pp. 353-365.
25. Herlea, D.E., Jonker, C.M., Treur, J., and Wijngaards, N.J.E., (1999). Specification of Behavioral Requirements within Compositional Multi-Agent System Design. In: F.J. Garijo, M. Boman (eds.), Multi-Agent System Engineering, Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW' 99. Lecture Notes in AI, vol. 1647, Springer Verlag, Berlin, 1999, pp. 8-27.
26. Hubner, J.F., Sichman, J.S., and Boissier, O. (2002). A Model for the Structural, Functional and Deontic Specification of Organizations in Multiagent Systems. In: Proc. 16th Brazilian Symposium on Artificial Intelligence (SBIA' 02), Porto de Galinhas, Brasil, 2002. Extended abstract: MOISE+: Towards a Structural, Functional and Deontic model for MAS Organizations. In: C. Castelfranchi and W.L. Johnson (eds.), Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS' 02. ACM Press, 2002, pp. 501-502.
27. Jonker, C.M., Letia, I.A., and Treur, J., (2002). Diagnosis of the Dynamics within an Organisation by Trace Checking of Behavioural Requirements. In: Wooldridge, M., Weiss, G., and Ciancarini, P. (eds.), Proceedings of the 2nd International Workshop on Agent-Oriented Software Engineering, AOSE' 01. Lecture Notes in Computer Science, vol. 2222. Springer Verlag, 2002, pp. 17-32.
28. Jonker, C.M. and Treur, J. (2002). Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. *International Journal of Cooperative Information Systems*, vol. 11, 2002, pp. 51-92. Earlier, shorter version in: W.P. de Roever, H. Langmaack, A. Pnueli (eds.), Proceedings of the International Workshop on Compositionality, COMPOS' 97. Lecture Notes in Computer Science, vol. 1536, Springer Verlag, 1998, pp. 350-380.
29. Jonker, C.M., Treur, J., and Wijngaards, W.C.A., (2001). Temporal Languages for Simulation and Analysis of the Dynamics Within an Organisation. In: B. Dunin-Keplicz and E. Nawarecki (eds.), From Theory to Practice in Multi-Agent Systems, Proceedings of the Second International Workshop of Central and Eastern Europe on Multi-Agent Systems, CEEMAS' 01, 2001. Lecture Notes in AI, vol. 2296, Springer Verlag, 2002, pp. 151-160.
30. Kowalski, R., and M.A. Sergot (1986). A logic-based calculus of events, New Generation Computing, 4, 1986, 67-95.
31. Kreitner, R., and Kunicki, A. (2001). Organisational Behavior, McGraw – Hill.
32. Lomi, A., and Larsen, E.R. (2001). Dynamics of Organizations: Computational Modeling and Organization Theories, AAAI Press, Menlo Park.
33. Maes, P, Nardi, D. (eds), *Meta-level architectures and reflection*, Elsevier Science Publishers, 1988.
34. Manna, Z., and Pnueli, A., Temporal Verification of Reactive Systems: Safety. Springer Verlag, 1995.
35. McMillan, K., *Symbolic Model Checking*, Kluwer Academic Publishers, 1993.
36. Meinke, K., and Tucker J.V. (eds.) (1993). Many-sorted Logic and its Applications, John Wiley & Sons, Inc. Publishers - Chichester; New York.
37. Mintzberg, H. (1979). The Structuring of Organisations, Prentice Hall, Englewood Cliffs, N.J.
38. Moss, S., Gaylard, H., Wallis, S., and Edmonds, B. (1998). SDML: A Multi-Agent Language for Organizational Modelling, Computational and Mathematical Organization Theory 4, (1), 43-70.
39. Prietula, M., Gasser, L., Carley, K. (1997). Simulating Organizations. MIT Press.
40. Reiter, R. (2001). *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, Cambridge MA: MIT Press, 2001
41. Sichman, J.S. and Conte, R., On Personal and Role Mental Attitudes: A Preliminary Dependence-Based Analysis. In: F. Oliveira (ed.), Advances in AI (Proc. 14th Brazilian Symposium on Artificial Intelligence, SBIA' 98). Lecture Notes in Artificial Intelligence, vol. 1515, Springer Verlag, Berlin-Alemanha, 1998, pp 1-10.
42. Weyhrauch, R.W. (1980). Prolegomena to a theory of mechanized formal reasoning, Artificial Intelligence 13 (1980), pp. 133-170.