Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1453–1460
DOI: 10.15439/2016F547
ACSIS, Vol. 8. ISSN 2300-5963

Modelling Group Constructions for Social Analysis

Daniela Xavier
GARP (Genomic and RNA Profiling Core),
Department of Molecular and Human Genetics,
Baylor College of Medicine
in Houston, United States
Email: xavier@bcm.edu

Rubén Fuentes-Fernández
GRASIA (Research Group on
Agent-based, Social & Interdisciplinary Applications),
Universidad Complutense de Madrid
in Madrid, Spain
Email: ruben@fdi.ucm.es

***Abstract*—Agent-based modelling is becoming widely used for studies in Social Sciences. However, its application faces limitations coming from its bias to software development, which precludes a more active involvement of social researchers. In order to deal with this problem, this work proposes using domain-specific modelling languages based on the socio-psychological Activity Theory. These languages apply agent research to crystallize that theoretical framework in a formal definition suitable for automated processing but close to Social Sciences. The paper focuses on the language for the specification of group constructions such as organizations, norms and shared knowledge. A case study about contradictory decisions in the space shuttle program illustrates the discussion.**

I. Introduction

AGENT-BASED Modelling (ABM) [1] has become a mainstream technique for research in Social Sciences. Traditionally, this research requires the gathering of data over potentially long periods of time and about large populations that are not fully controlled, which largely increases its costs. The formal description of social systems with models [2] allows applying analysis techniques based on simulation and verification, reducing the needs of eliciting data for the initial testing of hypotheses. ABM facilitates modelling by providing social and intentional computational abstractions that are closer to the concepts used in this research field than those of other approaches.

Despite of its advantages, ABM is limited by its inherent development complexity. The design, implementation and use of an archetypical agent-based model involve different subtasks and roles, which need diverse backgrounds and competences [3]. This situation may lead to misunderstandings between the different stakeholders, which make it hard to guarantee that the model really corresponds to the initial requirements of social science researchers [4]. Available methodologies to develop such ABM models [3], [5], [6] offer little help to address this problem. They focus on the researchers' conceptual models and describe very general tasks for the development. Agent-Oriented Software Engineering (AOSE) also seems not to be a suitable solution, since it mostly deals with the development of standard Multi-Agent Systems (MAS) [7]. On the contrary, ABM focuses more on the translation of the conceptual models of social science researchers to computational models using agent abstractions [5]. Besides, MAS are usually tied to certain architectural patterns, while ABM deals with an enormous heterogeneity of structures [1]. There are also differences about implementation. ABM applications usually require centralized monitoring components that can access the internals of agents in order to gather analysis data [8], and sacrifice the individual agent complexity in favour of huge populations [4]. These aspects are often not considered by AOSE methodologies [7].

Model-Driven Engineering (MDE) [9] with Domain-Specific Languages (DSL) [10] has been proposed as a way to overcome some of these limitations [11], [12]. A DSL for ABM uses a vocabulary grounded on conceptual frameworks from Social Sciences. The DSL has a formal definition which enables the use of automated MDE techniques to process its models, for instance to generate the code for simulations. Such approach reduces the impact of misunderstandings in ABM in two ways. First, social science researchers are able to perform the modelling themselves using a language they are familiar with. Second, transformations from conceptual models to computational ones can be generalized and reused in different projects. This offers improved opportunities for verification and validation. There are some preliminary examples of this trend, but they are still too biased to their foundations in software development [12] or only present partial solutions for some modelling aspects [11]. Especially issues like the management of large populations or centralized supervision are still open.

This paper introduces a DSL for modelling social constructs and organizational interactions. It is part of the ATCAS (Activity Theory for the Computational Analysis of Societies) framework. The Activity Theory (AT) [13] is a well-known paradigm for the analysis of human groups. It focuses on the study of *activities*, which are interactive acts between people and their physical and socio-cultural environment, where both of them act on and mould the other at the same time. ATCAS is intended to provide a general and extensible basis for ABM in social research supporting different applications. The actual representation of AT concepts in ATCAS depends on agent concepts, for instance about inconsistency management [14], code generation [12] and organizational interactions [15]. This last aspect is the focus of the current paper.

From the AT perspective [16], the social aspect of groups regards **communities** of subjects engaged in shared activities. The norms of the *division of labour* organize these activities and *rules* emerging from the socio-cultural environment influ-

978-83-60810-90-3/$25.00

ence and constraint both communities and activities. AT works at the level of abstraction of human societies but the automated analysis of models in MDE requires formalizing them as computational abstractions. ATCAS-IL adopts for this purpose the OperA [15] framework for the specification and analysis of agent organizations. OperA provides predefined modelling primitives for the high-level specification of organizations and the Logic for Contract Representation (LCR), which is an extension of deontic temporal logic, for the fine-grained details. The choice of OperA is motivated by the fact that AT explicitly considers social features that are not embedded in individual agents, but apply to the society as a whole. ATCAS-IL extends OperA primitives with AT concepts and introduces a macro mechanism intended to tailor the language for the specific needs of social science researchers. This gradually increases their autonomy in modelling, as they can define their own abstractions.

The formal description of ATCAS-IL uses metamodels. This is a common technique for language definition in MDE [9] that facilitates language extension and evolution. MDE manipulates models (i.e. instances of metamodels) mainly using standard transformation languages. In this way, engineers do not need to devote effort to the low-level details of model processing, but just to define the transformations for the different purposes. In the case of ATCAS-IL, transformations consider the semantics of the DSL according to AT and agent research. These transformations generate code, for instance, for checking contradictions as [14] and simulation as [12]. This kind of approach has already been successfully tested in other domains [17].

The remainder of the paper further explains the elements in this introduction. Section II presents AT and the case study that guides the discussion in this paper. Section III provides a brief introduction to OperA. ATCAS-IL is introduced in section IV, and section V applies it to the analysis of the case study. Section VI compares the presented approach with existing ABM works. Finally, section VII discusses some conclusions about ATCAS-IL.

II. Activity Theory

The Activity Theory (AT) [13] is a socio-psychological paradigm for the study of human behaviour. It focuses on the mutual dialectics between people and their physical and social environment: the environment shapes human actions and their execution, and is also changed by these same actions. Hence, human acts cannot be analyzed independently of their context. These contextualized acts constitute the minimal meaningful unit of analysis and are called *activities*.

An *activity* [18] is a transformation process driven by people's needs. These needs are satisfied with an *outcome* produced transforming an *object*. Any element used in this process is a *tool*. The active component that carries out the activity is the *subject*. Subjects with a set of common social meanings constitute a *community* [16], which represents the socio-historical context of the activity. Two bodies of social constructions mediate the relationships of communities in the activity: *rules* with the subject and the *division of labour* with the object. Both of them contain similar elements, such as knowledge, implicit assumptions or norms. The key difference is the focus. The division of labour regards task specialization in the community through aspects such as power relationships, goal decomposition or the assignment of responsibilities. On the contrary, rules are guides and constraints not targeted specifically to the activity but affecting it, such as group beliefs, country laws or accepted scientific theories. The different elements can be both physical and mental, so AT considers both types of activities with a unifying analysis. All these elements make up the context of an activity, which is named its *activity system*. Its traditional depiction [16] appears in Fig. 1.

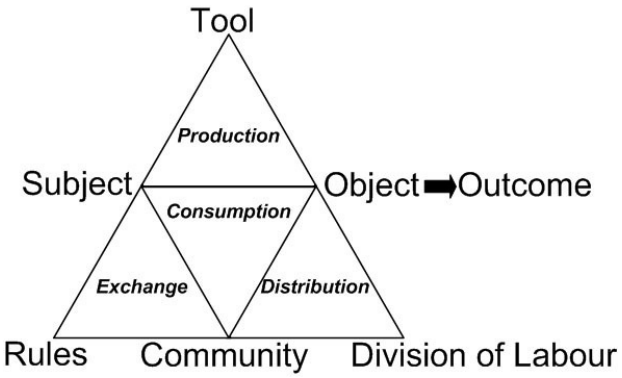

Fig. 1. AT depiction of an activity system.

Activity systems always exist in neighbourhoods of interconnected activity systems linked by shared elements. The execution of an activity produces outcomes that become the artefacts (e.g. subject, tool or rules) needed to execute other activities. Subjects carry out activities in these networks following their own rationality.

AT also considers the hierarchical decomposition of activities. *Activities* pursue high-level *objectives* that meet people needs. These activities are executed through sequences of *actions*, which try to achieve low-level *goals*. These goals do not satisfy by themselves any need, but they contribute or are part of higher-level goals. In their turn, actions are implemented through *operations* that depend on the specific state of the *environment*.

The evolution of activity systems over time depends on their inner *contradictions*. These contradictions are conflicts between the elements in the networks of activity systems, and they can appear both inter and intra systems. Subjects try to remove contradictions through the evolution of the involved activity systems, commonly generating new tensions that produce further evolution.

As an example of AT analysis, this paper considers the work in [19] about the Challenger crash. In 1986, the Challenger shuttle exploded shortly after launching. This accident has been frequently used as a case study about engineer ethics, communication and group thinking. According to Holt and

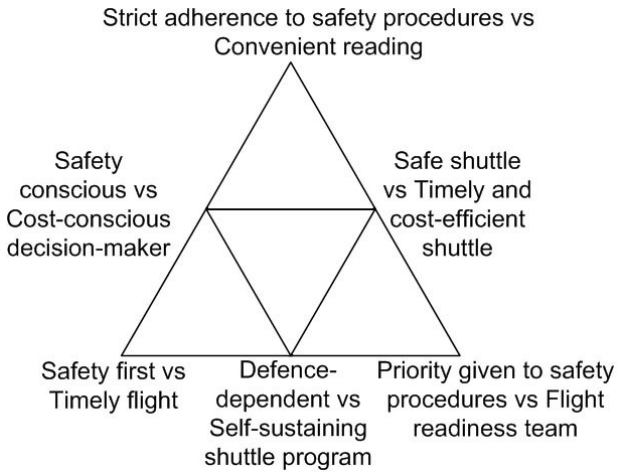

Fig. 2. Shuttle flight readiness activity system in the 1980s adapted from [19].

Morris [19], NASA began in the early 1960s as a tightly coupled set of subsystems aimed at space research and development. Over the 1980s, it became the home of several competing groups. At the same time, the lowering perceived value of the space research in public opinion put the agency under increasing pressure to cut down costs. NASA could not satisfy all its opposing goals, so a critical situation made it inevitable to violate some of them.

The analysis focuses on the readiness activity where the decision is taken to launch the shuttle or not. It discusses how the inner contradictions present in the NASA's organization culture at the end of the 1980s led to the catastrophe, how these contradictions generated the new and culturally more advanced activities in today's organization, and the still existing contradictions in the current forms of activities. Though the case is inevitably simplified, it synthesizes several complex settings and lengthy sources of information. For instance, the report of the Presidential Commission about the accident comprehends 5 volumes of documentation and analysis[1].

Fig. 2 depicts the readiness activity system. The nodes of the triangle show the opposition between the two perspectives about the NASA. The initial NASA from the time of the Cold War had priorities focused on research and safety, and almost unbounded resources. The new cost-aware agency competes with other agencies all over the world for the launching of satellites. The roots of this conflict appear in the object, which is the launching of a shuttle. It opposes the *safe shuttle* need to the *timely and cost-efficient shuttle* need. This opposition permeates all the other elements in the activity system. For the *subject*, it opposes the safety and the cost consciousness of the decision maker. The tools include information, devices and procedures supporting the activity. In this case, they are focused on the safety procedures, confronting a strict adherence to them with a convenient reading, which is only concerned about critical errors and relies on the low probability of certain failures. The social context of the activity shows this same duality. The community includes the NASA *defence-dependent* with high funding of the Cold War, and the one of decreasing funding in the 1980s aimed at getting a *self-sustaining shuttle program*. The division of labour regards work organization. In this case, it defines the acceptable tradeoffs between prioritizing security and the *flight readiness team*. These norms emerge from a wider context of rules where NASA was pushed to have successful and cost-effective *timely flights* more than to guarantee *safety first* beyond any reasonable risk.

These intra-node tensions scale up to contradictions between nodes. When the decision-maker needed to decide about launching, he/she was confronted with the dual nature of its community, tool and rules. Whatever decision this engineer made, it could lead to failure in some of the objectives. This situation is known as a double-bind contradiction [20]. In it, the system trying to satisfy opposed goals is under growing pressure that it can only overcome through its evolution.

This paper uses this case study to illustrate the application of ATCAS-IL to model complex organizations and subsequently analyze particular features of them. Specifically, it shows how its formal specification method supports the automated identification and analysis of the inherent contradictions of the NASA organization.

III. The OperA framework

AT considers that the integrated analysis of the individual and social contexts of activities is key for their understanding. Thus, its formalization as a DSL needs to include and define precisely the behavioural aspects common to the artefacts present in both scopes. For this reason, we have chosen OperA as the basis for ATCAS-IL.

OperA [15] is a framework for the specification of agent organizations. It considers the requirements regarding the structure, norms and external behaviour of these organizations and their members. Being focused on specification, OperA makes no assumption about the internals of the agents implementing the organization, and represents interactions through landmark patterns. These provide abstract representations of families of protocols. As such, OperA specifications enable different actual instantiations. Three models specify its organizations.

The Organizational Model (OM) describes the organization requirements using as main concepts, roles and scenes. The definition of a role includes its objectives and their decomposition, the norms applicable to that role, and the rights or capabilities it has. Scenes describe interactions between roles. They specify the participant roles, the ordering of actions through landmarks, the norms governing the scenes, and the results of their execution. This description makes use of information about the domain and the general normative structure applicable in the system.

The Social Model (SM) specifies the activity of agents in the society. Agents are executable entities able to enact the OM roles. OperA establishes constraints on the behaviour of agents using social contracts, but not their implementation.

[1]http://history.nasa.gov/Shuttlebib/ch7.html

These contracts indicate under which conditions an agent will play a role, and are used, for instance, to describe the benefits that the agent obtains, additional constraints or potential plans.

Finally, the Interaction Model (IM) describes how agents enacting roles participate in the scenes. It specifies the actual execution of the scene through a sequence of actions and adds additional norms for it.

OperA uses the Logic for Contract Representation (LCR) for the specification of contracts and norms in agent organizations. As stated in the introduction, it is an extension of deontic temporal logic that includes the *stit* operator (i.e. an agent *sees to it that*), the temporal operators of the branching temporal logics, and several deontic operators (i.e. obligation, permission and prohibition). Its expressions can be linked to deadlines corresponding to events or observed situations.

The OperA framework includes a graphical tool, OperettA [21], an IDE built as a plug-in of Eclipse. OperettA has facilities for the visual specification of organizations, syntax checking based on the OperA metamodel, static analysis, (e.g. dependent roles have to have a social link) and normative verification (e.g. inconsistencies between objectives and norms).

The formal logical semantics of OperA form a basis for the formal verification of the new specification language ATCAS-IL, as the paper shows in the next section.

IV. ATCAS-IL

ATCAS-IL extends the underlying OperA framework in several aspects determined by three main requirements. The first one is the foundation of ATCAS-IL in AT. It makes activities the focus of the analysis, which takes in turn the environment becoming a first class citizen of the specifications. The second one is the aim of ATCAS-IL for social analysis through automated tools. This requires specifying not only the constraints for the behaviour of the organization, but also some aspects of the actual behaviour of its members. The last requirement is the need of providing support to deal with the complexity of the description of social systems. Since these specifications account for a wide variety of issues and individuals, the language has to provide mechanisms to work with them at different levels of abstraction.

An activity is an act contextualized by its activity system [18]. The activity system includes subjects using tools to transform objects into outcomes, in the context of communities that specify the division of labour and the general rules applicable to that activity. While subjects, communities, and rules and division of labour can be roughly approximated by roles/agents, groups of these, and norms respectively, there are no suitable OperA concepts to describe activities, objects and tools.

Objects and tools are elements of the socio-physical environment. In ATCAS-IL, an *artefact* represents a general element of the environment. Following a widespread perspective in MAS [7], ATCAS-IL characterizes artefacts in terms of an internal state, the operations available to manipulate them, the events they can generate, and the knowledge and norms about their use. An *operation* is a basic act that can be executed when certain constraints are satisfied, and that generates some results. Both the artefact state, and the constraints and results of operations are defined with OperA formulae. *Environments* group artefacts adding specific norms.

Algorithm 1 Definition of the basic artefact. Bold words are ATCAS-IL (or OperA) keywords.

```
Artefact ( basic-id,
    Capabilities:,
    Knowledge:
        state(basic-id, non-available) or
        state(basic-id, available) or
        state(basic-id, blocked)
    Rules:
        FORBIDDEN (rule-0,
            Environmental-contract(
                Instance: _, Artefact: basic-id, Clauses: _ )
        )
        OBLIGED (rule-1,
            state(basic-id, available) BEFORE
            state(basic-id, blocked)
        )
        OBLIGED(rule-2,
            ( blockedBy(Ac1, basic-id) and
            blockedBy(Ac2, basic-id) ) ->
            ( Ac1 = Ac2)
        )
)
```

Algorithm 1 shows an example of the specification of an artefact. It introduces as knowledge the different states in which an artefact can be: *non-available* (i.e. non-created and therefore non-usable in activities), *available* (i.e. usable by activities) and *blocked* (i.e. used exclusively by an activity). Several specific rules limit the behaviour of this basic artefact: *rule-0* states that this is an abstract artefact and therefore no instance of it can be created; *rule-1* indicates that the artefact must be created (i.e. available) before it can be locked for exclusive use (i.e. blocked); *rule-2* states that an artefact cannot be blocked at the same time by different activities. All the artefacts in the case study satisfy this basic specification.

Communities control the access of subjects to *environments*. A community is a group of subjects that share social meanings and artefacts. ATCAS-IL defines them as an extension of the concept of role group in OperA. It comprehends several sets of elements: roles representing the subjects; the environments it has access to; accessibility rules mediating between subjects and artefacts; and knowledge and norms applicable in it. Roles are standard OperA roles whose capabilities are represented as operations. The accessibility rules indicate that when the states of the role and its environment meet some conditions, the role can use the operations of the artefact and be aware of its events.

The previous elements are put together in *activities*, whose definition following AT [16] includes the following elements:

- *Subjects* indicate who are in charge of executing the

activity. OperA roles describe them. *Objects* and *tools* represent the elements of the environment affected by the execution of the activity. *Artefacts* represent them.
- *Communities* determine how the subjects can access the artefacts related with the activity system.
- *Rules* and *division of labour* establish the norms that provide constraints for the execution of the activity.
- *Outcomes* of the activity are modelled as OperA formulae that become true after the activity execution.
- *Patterns* establish intermediate steps in the execution of activities.

As seen in Section II, activities are hierarchically decomposed in actions, which are further decomposed into operations. ATCAS-IL only considers operations in its refinement of activities. Patterns allow specifying the preferred order of generation of the results of operations.

The previous elements (i.e. communities and activities) define general patterns of behaviour, and appear at the level of OperA OM. The elements that actually implement them are the *agents* for the subjects and the *instances* for the artefacts. SM define these elements. An agent is characterized by its objectives, capabilities (i.e. operations), applicable norms and priorities about objectives. Objectives are states of the world that agents pursue defined as OperA formulae. Priorities determine the objectives that agents prefer trying to achieve when their states and that of the accessible environments meet certain conditions. These priorities provide the means to further describe the dynamic behaviour of systems. OM indicate existing objectives and their satisfaction conditions, and the priorities establish a simple way to choose the objective to attempt when several alternatives are available. The instances of artefacts are described in terms of their capabilities, and the knowledge and rules about them.

Finally, IM define for each activity the specific agents and instances that act in it, and sequences of operations to execute it. These sequences of operations must be able to satisfy the patterns included in the definition of the activity.

The division between the conceptual definition of the model (through communities and activity systems) and its actual realization (with instances, agents, and interactions) facilitates two goals in ATCAS-IL. First, it isolates the researchers' hypotheses, which appear at the conceptual level of OM, from the variability of individual elements, which SM and IM define. This allows managing the heterogeneity of individuals in populations, which is relevant to check that the obtained results are really a consequence of the hypotheses and not of specific configurations of the population. Second, it enables the independent evolution of the definition and implementation of subjects and artefacts. This evolution is required to adapt the realization of agent-based models to the requirements of different applications, as the already mentioned centralized supervision and different levels of complexity in the implementation of the individual agents depending on the population size.

ATCAS-IL also incorporates mechanisms to manage the models complexity. ABM [5] is increasingly being accepted as a tool for social research due to the inadequacy of analytical and traditional computational models to deal with large, heterogeneous and non-linear systems. Nevertheless, ABM models can also become quite complex, in cases that: involve several types of agents; where individuals evolve in different ways; or, large bodies of knowledge and rules affect the society behaviour. The management of such complexity requires that approaches incorporate abstraction mechanisms. Besides the decomposition of specifications in several models present in OperA, ATCAS-IL includes an extension mechanism for modelling primitives and the capability of defining macros.

The extension mechanism allows indicating that a given sub-concept extends a super-concept by including all its attributes. This mechanism is available for artefacts, environments, roles, communities, activities, activity systems, agents and instances. It reduces the size of the specifications and allows building conceptual hierarchies of concepts. Such hierarchies highlight the common features of concepts, allow their incremental description, and facilitate the definition of exceptions. For instance, they allow indicating that subjects usually comply with the rules of their communities, but a minority of them are going to break the law, that is, ignoring some of those constraints.

The second mechanism is the definition of macros. Deontic logic, used as formal basis of OperA, does not belong to the standard background of social science researchers, so they need the support of experts to use it. In order to improve the researchers' autonomy when using ATCAS-IL, their experiences are gradually crystallized in a set of tailored macros suitable for a domain. That is, researchers initially determine what they want to specify and experts in logics help them to describe these operators with basic logic primitives. After some studies, this early joint effort defines a researcher-friendly specialization of the DSL for that domain. That specialization includes macros for the most commonly used operators and concepts. Then, researchers can specify models on their own, and only need experts in logics to describe new and unusual properties.

V. Case study: NASA shuttle

This section uses ATCAS-IL to analyze the conflicts existing in the **flight readiness** activity system described in Section II. The AT description of this activity system has been presented in Section II and is depicted in Fig. 2. As stated there, the conflicts emerge from the NASA duality between an agency focused on research and safety, and the modern one trying to reduce costs. It affects the shuttle launching opposing a total guarantee of safety and a reduced one but with a timely launching.

The first step of this analysis is the formalization of the activity system using the primitives presented in Section IV. Algorithm 2 shows a simple mapping for it. The *flight-readiness-activity* extends the *basic-activity*, which includes basic information about activities. Its object has been simplified to focus on the contradictions coming from the community. The two communities reflect the duality of goals in NASA between the

Algorithm 2 ATCAS-IL activity system for Fig. 2.

Activity basic-activity (flight-readiness-activity,
Subjects : decision-maker,
Objectives : ,
Objects : Shuttle,
Tools : safety-procedure(Shuttle),
Communities : public-funded-agency,
autonomous-agency,
Rules : safety-first, timely-flight(Shuttle),
DivisionOfLabour : priority-to-safety-procedures,
flight-team-readiness,
Outcomes : shuttle-flight(Shuttle),
Patterns :
)

Algorithm 3 Communities of the flight readiness activity system.

Community (basic-community,
Roles : decision-maker,
Environments :,
Knowledge :,
Rules : **OBLIGED**(rule-3, cost(W) **and** safety(X) **and** funding(Y) **and** income(Z) **and** (W + X) < (Z + Y)),
Role-dependencies :,
Artefact-accessibility :
)
Community basic-community (public-funded-agency, ...)
Community basic-community (autonomous-agency,
Roles : cost-conscious-decision-maker : decision-maker,
Environments :,
Knowledge : funding(0),
Rules : ,
Role-dependencies : ,
Artefact-accessibility :
)

public-funded-agency and the *autonomous-agency*. The knowledge related with the artefacts in this activity is described as part of their definitions.

Algorithm 3 shows a partial definition of the communities involved in the previous activity system. The *basic-community* contains the role *decision-maker*, who is the subject of the activity, and a rule that states that the agency must run balancing its expenses (i.e. functioning and safety costs) and the incoming money (i.e. funding and incomes). As there are

Algorithm 4 Definition of rules.

DEF safety-first(Shuttle) = safety(Shuttle, X) **and** limit_safety(Y) **and** X < Y

DEF timely-flight(Shuttle) = expected_launch_time(Shuttle, T1) **and** launch_time(Shuttle, T2) **and** T2 < T1

Algorithm 5 Definition of roles.

Role (decision-maker,
Objectives : take-decision(Shuttle),
Sub-objectives :,
Rights : Check-readiness(Shuttle),
Decide-launch(Shuttle),
Knowledge :,
Rules :
)
Role (safety-conscious-decision-maker,
Objectives :,
Sub-objectives :,
Rights : Check-safety(Shuttle),
Knowledge :,
Rules : safety-first
)
Role (cost-conscious-decision-maker,
Objectives :,
Sub-objectives :,
Rights : Check-costs(Shuttle),
Knowledge :,
Rules : timely-flight
)

Algorithm 6 Implementation of the activity system.

Interaction-contract (
Activity : flight-readiness-activity,
Parties : (decision-maker : engineer),
Environment : (shuttle : challenger),
(safety-procedure : frr),
(public-funded-agency : nasa),
(autonomous-agency: nasa),
Clauses :,
Protocol : Check-readiness(challenger),
Check-safety(challenger), Check-costs(challenger),
Decide-launch(challenger)
)

no constraints about artefact accessibility, all the roles have granted full access to all the artefacts of the community. Depending on the type of agency, the specification adds more rules to its description. For instance, the *autonomous-agency* should run without public funding, which is asserted as *knowledge* of that community. Note that the specification of the *autonomous-agency* indicates that it constrains the general *decision-maker* to a *cost-conscious-decision-maker*, so it does not add a new subject but specializes the existing one.

The definition of the rules is illustrated with *safety-first* and *timely-flight* in Algorithm 4, which are also examples of the use of macros introduced by keyword *DEF*. The rule *safety-first* states that the probability of failure cannot be over a given limit, and *timely-flight* indicates that launchings must adhere to the expected planning.

The last type of elements to define is the roles involved

in the activity system. Algorithm 3 introduced the *decision-maker* as the subject of the flight readiness activity system. As also seen in Fig. 2, it is specialized in the safety-conscious and the cost-conscious decision makers. The definitions of these roles can be seen in Algorithm 5. They have available several operations, such as *check-readiness* in the *decision-maker*. All of them produce pieces of information according to the available state of the shuttle. The operation *decide-launch* takes this information to approve or deny the launching of the shuttle, which satisfies the objective *take-decision*.

Algorithms 3, 4, and 5 describe the general behaviour of the activity system. Then, the SM specifies the actual individuals implementing it, and the IM how they carry out the activity.

In this case, the SM just considers one instance per artefact and one agent for the decision maker (in this case an *engineer*) playing the two possible roles, the safety-conscious and the cost-conscious decision maker. This models the kind of situation that faced NASA engineers about launching the shuttle in the presence of non-optimal conditions.

The final element of the specification is the IM in Algorithm 6. It uses the previous instances to implement the activity system. The protocol establishes the sequences of steps that the roles can perform to execute the activity. The operations have been already discussed in this section.

The verification of the previous specification is the second step of the analysis with ATCAS-IL. It finds out the double-bind contradiction pointed out in [19]. When the engineer needs to abort the launching because it does not meet the *safety-first* condition, there is a violation of the *timely-flight* condition as no launching time is scheduled before the planned time. If the engineer decides launching anyway, there is a violation of the *safety-first* condition. This inability to take an action without violating constraints corresponds to the double-bind contradiction [20]. Thus, the verification has been able to automatically find the contradiction and to point out the conflicting properties. Details on the analysis process can be found in [17].

The presentation of the case study has used a textual specification with ATCAS-IL. OperettA supports visual modelling as long as the corresponding Eclipse models with the abstract and concrete syntaxes of the language are available. This step is required to complete a DSL suitable for social science researchers. They can easily grasp the primitives related with AT concepts, as they correspond to a structured textual representation of activity systems. However, the deontic logic used to specify the low-level details do not belong to their standard background. The use of macros and a graphical notation can reduce their difficulties to use it in their models.

VI. Related work

The field of ABM shares with general agent research many of its limitations. The review of Gilbert and Terna [5] about the implementation of ABM points out the lack of conceptually well-defined blocks for modelling and implementation guidelines.

ABM applies models to a large extent as a conceptual tool with only a swallow agreement about what an agent is [3]. Just to discuss some examples, the research in [11], [22], [6], [12] can be considered. The agents in [6] are modelled as sets of simple variables and rules to modify them. There is a set of external parameters not modifiable by agents, and a set of internal parameters that agents can modify with actions. Actions are rules triggered by certain conditions. There are also information links between agents that indicate when changes in their variables are propagated to other agents. The model can consider some noise in the communications to allow non-modelled environmental effects. Works as [22] have a more complex representation of the involved agents. Their condition-action rules include symbolic representations of elements such as goals, tests or capabilities, being therefore closer to traditional MAS. However, they do not represent key elements in MAS like the society, norms or decision making. Some researchers [11], [12] advocate the use of common MAS for ABM. This approach allows for the richer and more abstract modelling of individuals, but overlooks several relevant facts. The first is that MAS abstractions come from software development, and therefore they are not well-suited for social researchers. The second is that ABM usually deals with huge populations that cannot be implemented with computationally expensive agents.

ATCAS-IL aligns with research that promotes MAS for ABM, as it allows making modelling as complex as required. However, its language foundation is in AT, and thus in Social Sciences instead of AOSE. Moreover, ATCAS-IL relies on automated transformations of models to manage the implementation and its tradeoffs. This is an approach already pointed out in [12], though it proposes the use of programming languages instead of model transformation languages [9] as ATCAS-IL does. This last approach is intended to be closer to the concepts of the domain.

The issue of the lack of implementation guidelines was already mentioned in the introduction. ABM methodologies focus on capturing the information of the social systems at a very abstract level, but disregard how to migrate from these conceptual models to computational ones [3]. From the already mentioned works, [11], [22], [6] only consider the general features of agents in their models and the results of their simulations. The absence of general implementation details suggests that they rely on specific implementations for the problem at hand. This constitutes a major problem to validate the models, as it is not easy to know whether some of the results come from unintended features of the coding [4]. Approaches emerging from MDE and promoting the automated transformation of models [11], [12] facilitate to some extent this validation, as the same transformations are applied to different models. Nevertheless, the full validation of models in these example approaches still requires a complete understanding of the program code involved in the transformation. For this reason, model transformation languages [9], which work at the level of models, seem a more suitable approach for this task. Nevertheless, complete development

processes for ABM that guide researcher in the modelling process are still an open issue.

VII. Conclusion

This paper has introduced ATCAS-IL as part of the ATCAS framework for ABM. ATCAS-IL is a DSL to represent social constructions. It is based on two main sources: the socio-psychological AT to define its conceptual framework; the OperA framework for agent organizations for its formal definition. This foundation pursues two objectives. First, it tries to increase the autonomy of social researchers in ABM by directly applying their own concepts. This reduces the misunderstandings that inevitably appear when researchers need to rely on engineers without a background in Social Sciences for the modelling. Second, the formal definition of the language enables the automated processing of its models. ATCAS-IL proposes making these transformations through standard model transformation languages. This implies that the transfer of information between models is partly specified at the level of abstraction of models, and not with code. It is also expected to improve comparability between models: the application of the same transformations to different models of the same hypotheses should generate equivalent results.

The paper has illustrated the use of ATCAS-IL with the problem of the identification of the contradictions that led to a well-known failure in the space shuttle program. While the original AT work relies on the human analysis of data, a suitable model allows the automated discovery of the problem and its potential reasons.

ATCAS-IL has currently three main limitations. First, the domain-specific primitives are not enough to model a complete system. As the case study illustrates, low level-details have to be expressed with logics, which are not suitable for social researchers. Ongoing work is intended to determine the additional primitives required in the language according to AT and agent research. AT also considers recurrent social patterns [16], [14] that can be described with reusable macros. Second, textual specifications are too verbose given the amount of details required in the models. The extensions of the DSL can help to solve this issue. Besides, the use of the visual modelling capabilities of OperettA can simplify the development of the specifications, hiding the repetitive details of modelling. Third, facilitating the development of the automated model transformations is still an open issue, as social researchers are not expected to be experts in transformation languages. Approaches based on the automated generation of transformations from model prototypes are a potential solution.

Acknowledgment

This work has been done in the context of the project "Collaborative Ambient Assisted Living Design (ColoSAAL)" (grant TIN2014-57028-R) supported by the Spanish Ministry for Economy and Competitiveness, the research programme MOSI-AGIL-CM (grant S2013/ICE-3019) supported by the Autonomous Region of Madrid and co-funded by EU Structural Funds FSE and FEDER, and the "Programa de Creación y Consolidación de Grupos de Investigación" (UCM-BSCH GR35/10-A).

References

[1] M. W. Macy and R. Willer, "From factors to actors: computational sociology and agent-based modeling," *Annual Review of Sociology*, vol. 28, pp. 143–166, 2002. [Online]. Available: http://www.jstor.org/stable/3069238

[2] R. Axelrod, "Advancing the art of simulation in the social sciences," in *Simulating social phenomena*. Springer, 1997, pp. 21–40.

[3] A. Drogoul, D. Vanbergue, , and T. Meurisse, "Multi-agent based simulation: where are the agents?" in *Multi-Agent-Based Simulation II*, vol. 2581. Springer, 2003, pp. 43–49.

[4] R. L. Axtell and E. J. M., "Agent-based modeling: understanding our creations," *The Bulletin of the Santa Fe Institute*, vol. 9, no. 2, pp. 28–32, 1994.

[5] N. Gilbert and P. Terna, "How to build and use agent-based models in social science," *Mind & Society*, vol. 1, no. 1, pp. 57–72, 2000.

[6] A. Pyka and G. Fagiolo, "Agent-based modelling: a methodology for neo-schumpeterian economics," *Volkswirtschaftliche Diskussionsreihe*, vol. 272, pp. 1–26, 2005. [Online]. Available: http://www.wiwi.uni-augsburg.de/vwl/institut/paper/272.pdf

[7] B. Henderson-Sellers and P. Giorgini, Eds., *Agent-oriented methodologies*. IGI Global, 2005.

[8] N. M. Gotts, J. G. Polhill, , and A. N. R. Law, "Agent-based simulation in the study of social dilemmas," *Artificial Intelligence Review*, vol. 19, pp. 3–92, 2003.

[9] R. France and B. Rumpe, "Model-driven development of complex software: a research roadmap," in *Proceedings of the 2007 Future of Software Engineering Conference (FOSE 2007)*. IEEE Computer Society, 2007, pp. 37–54.

[10] A. van Deursen and J. Visser, "Domain-specific languages: an annotated bibliography," *ACM Sigplan Notices*, vol. 35, no. 6, pp. 26–36, 2000.

[11] S. Hassan, R. Fuentes-Fernández, J. M. Galán, A. López-Paredes, , and J. Pavón, "Reducing the modeling gap: On the use of metamodels in agent-based simulation," in *Proceedings of the 6th Conference of the European Social Simulation Association*, 2009, pp. 1–12.

[12] C. Sansores, J. Pavón, and J. J. Gómez-Sanz, "Visual modeling for complex agent-based simulation systems," in *Multi-Agent-Based Simulation VI*, vol. 3891. Springer, 2006, pp. 174–189.

[13] L. S. Vygotsky, Ed., *Mind and Society*. Harvard University Press, 1978.

[14] G.-S. J. Fuentes-Fernández, R. and J. Pavón, "Managing contradictions in multi-agent systems," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 8, pp. 1243–1250, 2007. [Online]. Available: http://search.ieice.org/bin/summary.php?id=e90-d_8_1243

[15] V. Dignum, F. Dignum, , and J. Meyer, "An agent-mediated approach to the support of knowledge sharing in organizations," *The Knowledge Engineering Review*, vol. 19, no. 2, pp. 147–174, 2005.

[16] Y. Engeström, Ed., *Learning by expanding: an activity-theoretical approach to developmental research*. Orienta-Konsultit, 1987.

[17] P. Moreno-Ger, S.-R. J. Fuentes-Fernández, R., and B. Fernández-Manjón, "Model-checking for adventure videogames," *Information and Software Technology*, vol. 51, no. 3, pp. 564–580, 2009.

[18] A. N. Leontiev, Ed., *Activity, Consciousness, and Personality*. Prentice-Hall, 1978.

[19] G. R. Holt and A. W. Morris, "Activity theory and the analysis of organizations," *Human Organization*, vol. 52, no. 1, pp. 97–109, 1993.

[20] G. Bateson, Ed., *Steps to an Ecology of Mind*. Ballantine Books, 1987.

[21] D. Okouya and V. Dignum, "Operetta: a prototype tool for the design, analysis and development of multi-agent organizations," in *AAMAS 2008 demo papers*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 1677–1678.

[22] M. K.-K. T. Murakami, Y. and T. Ishida, "Multi-agent simulation for crisis management," in *Proceedings of the IEEE Workshop on Knowledge Media Networking*. IEEE Computer Society, 2002, pp. 135–139.