Authority and its Implementation in Enterprise Information Systems

Alexei Sharpanskykh

Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands,
sharp@few.vu.nl

Abstract. The concept of power is inherent in human organizations of any type. As power relations have important consequences for organizational viability and productivity, they should be explicitly represented in enterprise information systems (EISs). Although organization theory provides a rich and very diverse theoretical basis on organizational power, still most of the definitions for power-related concepts are too abstract, often vague and ambiguous to be directly implemented in EISs. To create a bridge between informal organization theories and automated EISs, this paper proposes a formal logic-based specification language for representing power- (in particular authority) relations. The use of the language is illustrated by considering authority structures of organizations of different types. Moreover, the paper demonstrates how the formalized authority relations can be integrated into an EIS.

1 Introduction

The concept of *power* is inherent in human organizations of any type. Power relations that exist in an organization have a significant impact on its viability and productivity. Although the notion of power is often discussed in the literature in social studies (Bacharach and Aiken, 1977; Blau and Scott, 1962; Clegg, 1989; Friedrich, 1958; Gulick and Urwick, 1937; Hickson et al., 1971; Parsons, 1947; Peabody, 1964), it is only rarely defined precisely. In particular, power-related terms (e.g., control, authority, influence) are often used interchangeably in this literature. Furthermore, the treatment of power in different streams of sociology differs significantly. One of the first definitions for power in the modern sociology was given by Max Weber (1958): *Power is the probability that a person can carry out his or her own will despite resistance*. Weber and his followers (Dahl, Polsby) considered power as an inherently coercive force that implied involuntary submission and ignored the relational aspect of power. Other sociologists (Bierstedt, Blau) considered power as a force or the ability to apply sanctions (Blau and Scott, 1962). Such view was also criticized as restrictive, as it did not pay attention to indirect sources and implications of power (e.g., informal influence in decision making) and subordinate's acceptance of power. Parsons (1947) considered power as "*a specific mechanism to bring about changes in the action of organizational actors in the process of social interaction*".

Most contemporary organization theories explore both formal (normative, prescribed) and informal (subjective, human-oriented) aspects of power (Clegg, 1989; Peabody, 1964; Scott, 2001). Formal power relations are documented in many modern organizations and, therefore, can be explicitly represented in models on which enterprise information systems (EISs) are based. The representation of formal power in EISs has a number of advantages. First, it allows a clear definition of rights and responsibilities for organizational roles (actors) and a power structure. Second, based on the role specifications, corresponding permissions for information, resources and actions can be specified for each role. Third, explicitly defined rules on power enable the identification of violations of organizational policies and regulations. Fourth, data about power-related actions (e.g., empowerment, authorization) can be stored in an EIS for the subsequent analysis.

For modeling of power relations the rich theoretical basis from social science can be used. Notably many modern EISs implement no or very simplified representations of power relations and mechanisms. In particular, the architecture ARIS (Scheer and Nuettgens, 2000) used for development of EISs identifies responsibility and managerial authority relations on organizational roles, however, does not provide general mechanisms for representing such relations and does not address change of these relations over time. The enterprise architecture CIMOSA (1993) distinguishes responsibilities and authorities on enterprise objects, agents, and processes/activities. However, no precise meaning (semantics) is attached to these concepts, which may be interpreted differently in different applications. Also, different aspects of authorities are not distinguished both in ARIS and in CIMOSA (e.g., authority for execution, authority for supervision, authority for monitoring).

Often EISs realize extensive access schemata that determine allowed actions for roles and modes of access of roles to information (Bernus, Nemes and Schmidt, 2003). Normally such schemata are based on power relations established in organizations. Thus, to ensure consistency, unambiguousness and completeness of EISs' access schemata, organizational power relations should be precisely identified and specified using some (formal) language. To this end, theoretical findings on organization power from social science are useful to consider. However, there is an obstacle to the direct implementation of this knowledge in EISs – the absence of operational definitions of power-related concepts in social theories.

The first step to make the concept of power operational is to provide a clear and unambiguous meaning for it (or for its specific aspects). In this paper this is done by identifying the most essential characteristics and mechanisms of power described in different approaches and by integrating them into two broad categories: formal power (or authority) and informal power (or influence), which are described in Section 2. Further this paper focuses on the formal representation of authority, for which a formal language is described in Section 3. Moreover, Section 3 illustrates how the introduced formal language can be used to model authority systems of different types of organizations. Section 4 discusses the integration of formal authority relations into an automated EIS. Finally, the paper concludes with a discussion in Section 5.

2 Power, authority and influence

As in many contemporary social theories (Clegg, 1989; Peabody, 1964), we assume that power can be practiced in an organization either through (formal) *authority* or through (informal) *influence relations*. Authority represents formal, legitimate organizational power by means of which a regulated normative relationship between a superior and a subordinate is established. Usually authority is attached to positions in organizations. For example, authority of some managerial positions provides power to hire or to fire; to promote or to demote; to grant incentive rewards or to impose sanctions. In many approaches it is assumed that authority implies involuntary obedience from subordinates. Indeed, as authority has a normative basis that comprises formal, explicitly documented rules, it is expected that subordinates, hired by the organization, should be aware of and respect these rules, which implies the voluntary acceptance of authority.

All manifestations of power that cannot be explained from the position of authority fall into the category of influence. In contrast to authority, influence does not have a formal basis. It is often persuasive and implies voluntary submission. Some of the bases of influence are technical knowledge, skills, competences and other characteristics of particular individuals. Influence is often exercised through mechanisms of leadership; however, possession of certain knowledge or access to some resources, as well as different types of manipulation may also create influence. Influence may be realized in efforts to affect organizational decisions indirectly.

Although authority and influence often stem from different sources, they are often interrelated in organizations. For example, the probability of the successful satisfaction of organizational goals increases, when a strong leader (meaning a leader that has a great value of influence) occupies a superior position of authority. Furthermore, sometimes patterns of influence that frequently occur in an organization may become institutionalized (i.e., may become authority relations).

Modeling methods for authority and influence are essentially different. While authority relations are often prescriptive and explicitly defined, influence relations are not strictly specified and may vary to a great extent. Therefore, whereas authority relations can be generally represented in EISs, the specification of influence relations is dependant on particular (cognitive) models of agents that represent organizational actors. Relations between authority and influence can be studied by performing simulation with different types of agents situated in different organizational environments. The focus of this paper is on modeling of formal authority relations. Influence relations and relations between authority and influence will be considered elsewhere.

3. Authority: a Formal Approach

First, in Section 3.1 a formal language for specifying authority-related concepts and relations is introduced. Then, Section 3.2 discusses how the introduced language can be used for representing authority structures of organizations of different types.

3.1 A Formal Language

Simon (1957) describes three contributions of authority for an organization: (1) the enforcement of responsibility, (2) the specialization of decision-making, and (3) the coordination of activity. Based on this and other theoretical findings that describe power, duties and responsibilities of organizational positions (Mintzberg, 1979), a number of relations for the specification of formal authority can be identified. These relations are defined on positions (or roles), without considering particular agents (individuals). The relations are formalized using the order sorted-predicate language (Manzano, 1996) and are presented graphically in Fig.1.

We represent all activities of an organization (including decision making and personnel-related activities) by processes. Each organizational role is associated with one or more process. Roles may have different rights and responsibilities with respect to different aspects of the process execution. Furthermore, often several roles may potentially execute or manage certain processes. This is represented by the relation

is_authorized_for: r:ROLE x aspect: ASPECT x a:PROCESS, where aspect has one of the values {execution, monitoring, consulting, tech_des (making technological decisions), manage_des (making managerial decisions), user_defined_aspect}.

All types of decisions with respect to a particular process can be divided into two broad groups: *technological* and *managerial decisions* (inspired by (Bacharach and Aiken, 1977)). Technological decisions concern technical questions related to the process content and are usually made by technical professionals. Managerial decisions concern general organizational issues related to the process (e.g., the allocation of employees, process scheduling, the establishment of performance standards, provision of resources, presenting incentives and sanctions). Managers of different levels (i.e., from the lowest level line managers to strategic apex (top) managers) may be authorized for making different types of managerial decisions varying from in scope, significance and detail. A particular decision type is specified as an aspect in the is_authorized_for relation. The same holds for technological decisions. Whereas *consulting* has a form of recommendation and implies voluntary acceptance of advices, decisions imposed on a role(s) that execute(s) the process are considered as imperatives with corresponding implications.

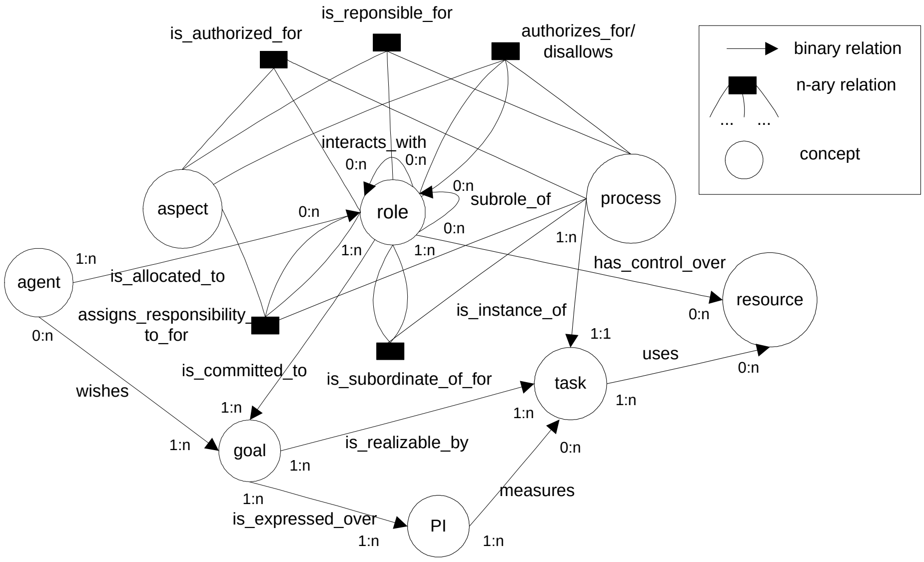

Fig. 1. Graphical representation of the concepts and relations of the language used for specifying formal authority relations

Authorization for *execution* implies that a role is allowed to execute the process according to existing standards and guidelines. Whenever a problem, a question or a deviation from the standard procedures occurs, the role must report about it to the role(s) authorized for making technological/managerial (depending on the problem type) decisions and must execute the decision(s) that will follow.

Monitoring implies passive observation of (certain aspects of) process execution, without intervention.

Notice that other aspects of process execution described in the managerial literature (e.g., control, supervision) can be represented as a combination of already introduced aspects. In particular, control can be seen as the conjunction of monitoring and making technological and/or managerial decisions aspects; supervision can be defined as the combination of consulting and control. Furthermore, the designer is given the possibility to define his/her own aspects and to provide an interpretation to them.

Although several roles in an organization may be authorized for a certain aspect related to some process, only one (or some) of them will be eventually (or are) responsible for this aspect. For example, the responsibility of a certain role with respect to the process execution means that the role is actually the one who will be performing the process and who holds accountability of the process execution. Furthermore, responsibility for the process execution implies allowance to use resources required for the process performance. The responsibility relation is specified as:

is_responsible_for: r:ROLE x aspect:ASPECT x a:PROCESS: process a is under responsibility of role r with respect to aspect (defined as for authorized_for)

Some roles are authorized to make managerial decisions for authorizing/disallowing other roles for certain aspects with respect to process execution. The authorization/ disallowance actions are specified by the following relations:

authorizes_for: r1:ROLE x r2:ROLE x aspect: ASPECT x a:PROCESS: role r1 gives the authority for aspect of process a to role r2.

disallows: r1:ROLE x r2:ROLE x aspect: ASPECT x a:PROCESS: role r1 denies the authority for aspect of process a for role r2.

However, to make a role actually responsible for a certain aspect of the process, another role besides the authority to make managerial decisions should also be the superior of the role with respect to the process. Superior-subordinate relations with respect to organizational processes are specified by: is_subordinate_of_for: r1: ROLE x r2: ROLE x a:PROCESS. Then, responsibility is assigned/retracted using the following relations:

assigns_responsibility_to_for: r1: ROLE x r2:ROLE x aspect: ASPECT x a:PROCESS: role r1 assigns the responsibility for aspect of process a to role r2.

retracts_responsibility_from_for: r1: ROLE x r2:ROLE x aspect: ASPECT x a:PROCESS: role r1 retracts responsibility from role r2 for aspect of process a.

Using these relations superiors may delegate/retract (their) responsibilities for certain aspects of processes execution to/from their subordinates, and may restrict themselves only to control and making decisions in exceptional situations.

In (Hickson et al., 1971) control over resources is identified as an important source of power. Therefore, it is useful to identify explicitly which roles control resources by means of the relation has_control_over: r1: ROLE x res:RESOURCE. In the proposed modeling framework the notion of resource includes both tangible (e.g., materials, tools, products) and abstract (information, data) entities.

Our treatment of authority is different from both formal approaches that consider authority as an attribute or a property inherent in an organization (Gulick and Urwick, 1937; Weber, 1958) and from the human-relation view that recognizes authority as an informal, non-rational and subjective relation (e.g., Follett, Mayo, cf. (Clegg, 1989)). As many representatives of converging approaches (e.g., C.I. Barnard, Simon (1957)) we distinguish between the formal authority prescribed by organizational policies and actual authority established between a superior and his/her subordinate in the course of social interactions. In the latter case a special accent lies on the acceptance of authority by a subordinate. In (Clegg, 1989) different cases of the authority acceptance are discussed: orders anticipated and carried out (anticipation); acceptance of orders without critical review; conscious questioning but compliance (acceptance of authority); discusses but works for changes; ignores, evades, modifies orders (modification and evasion); rejection of authority (appeals to co-workers or higher rank for support). Depending on the organizational type, varying administrative sanction may be applied in case an employee does not accept an authoritative communication, when he/she: (a) correctly understands/interprets this communication; (b) realizes that this communication complies with formal organizational documents and/or is in line with organizational goals; (c) is mentally and physically able to perform the required actions. In many modern organizations rewards and sanctions form a part of authority relation, thus, explicitly defined:

grants_reward_to_for: r1: ROLE x r: REWARD x r2: ROLE x reason: STRING: role r1 grants reward r to role r2 for reason

imposes_saction_on_for: r1: ROLE x s: SANCTION x r2: ROLE x reason: STRING: role r1 imposes sanction s to role r2 for reason

Sometimes authority relations may be defined with respect to particular time points or intervals (e.g., responsibility for some aspect of a process may be provided for some time interval). To express temporal aspects of authority relations the Temporal Trace Language (TTL) (Jonker and Treur, 2003) is used.

TTL allows specifying a temporal development of an organization by a trace. A trace is defined as a temporally ordered sequence of states. Each state corresponds to a particular time point and is characterized by a set of state properties that hold in this state. State properties are formalized in a standard predicate logic way (Manzano, 1996) using state ontologies. A state ontology defines a set of sorts or types (e.g., ROLE, RESOURCE), sorted constants, functions and predicates.

States are related to state properties via the formally defined satisfaction relation |=: state(γ, t) |= p, which denotes that state property p holds in trace γ at time t. For example, state(γ1, t1) |= is_responsible_for(employee_A, execution, p1) denotes that in trace γ1 at time point t1 the employee_A is responsible for the execution of process p1.

Dynamic properties are specified in TTL by relations between state properties. For example, the following property expresses the rule of a company's policy that an employee is made responsible for making technological decisions with respect to process p1 after s/he have been executing this process for two years (730 days):

∀γ: TRACE ∀t1:TIME ∀empl: EMPLOYEE state(γ, t1) |= is_responsible_for(empl, execution, p1) & ∃t2: TIME state(γ, t2) |= assigns_responsibility_to_for(management, empl, execution, p1) & t1-t2 = 730
⇒ state(γ, t1) |= assigns_responsibility_to_for(management, empl, tech_des, p1)

Other specific conditions (e.g., temporal, situational) under which authority relations may be created/maintained/dissolved are defined by executable rules expressed by logical formulae. The specification of these rules will be discussed in Section 4.

3.2 Modeling Authority Relations in Different Types of Organizations

Authority is enforced through the organizational structure and norms (or rules) that govern the organizational behavior. In general, no single authority system can be equally effective for all types of organizations in all times. An organizational authority system is contingent upon many organizational factors, among which organizational goals; the level of cohesiveness between different parts of an organization, the levels of complexity and of specialization of jobs, the level of formalization of organizational behavior, management style (a reward system, decision making and coordination mechanisms), the size of an organization and its units. Furthermore, the environment type (its uncertainty and dynamism; the amount of competitors), as well as the frequency and the type of interactions between an organization and the environment exert a significant influence upon an organizational authority structure.

In the following it will be discussed how authority is realized in some types of (mostly industrial) organizations and how it can be modeled using relations introduced in the previous Section 3.1.

Authority in small firms of the early industrial era was completely exercised by their owners through mechanisms of direct personal control. Firm owners were managers and technical professionals at the same time, and, therefore, had authority and responsibility for all aspects related to processes, except for their execution, responsibility for which was assigned to hired workers. This can be expressed using the introduced formal language as follows:

∀p: PROCESS ∀t: TIME ∀γ: TRACE ∃empl: HIRED_EMPLOYEE state(γ, t) |= [is_responsible_for(firm_owner, control, p) & is_responsible_for(firm_owner, supervision, p) & is_responsible_for(empl, execution, p)]

The owners controlled all resources (∀r: RESOURCE ∀t: TIME ∀γ: TRACE state(γ, t) |= has_control_over(firm_owner, r)). Currently similar types of organizations can be found in family business and small firms.

With the growth of industry, which caused joining of small firms into larger enterprises, owners were forced to hire subcontractors, who took over some of their managerial functions. This can be modeled using the introduced language as assigning responsibility to subcontractors by the owner for some managerial and technological decisions, as well as monitoring and consulting of workers with respect to some processes execution. For example, the responsibility assignment to role subcontractor_A for making managerial and technological decisions related to the process p1 is expressed as

∀γ: TRACE ∃t: TIME state(γ, t) |= [assigns_responsibility_to_for(firm_owner, subcontractor_A, tech_des, p1) ∧ assigns_responsibility_to_for(firm_owner, subcontractor_A, manage_des, p1)]

The owner reserved often the right to control for himself, which included granting rewards and imposing sanctions to/on subcontractors and workers, realized through superior-subordinate relations. For example, the following rule describes the superior-subordinate relations between the firm owner and subcontractor_A, responsible for making technological decisions related to process p1 and employee_A responsible for execution of process p1.

∀γ: TRACE ∀t: TIME state(γ, t) |= is_subordinate_of_for(subcontractor_A, firm_owner, p1) & is_subordinate_of_for(employee_B, firm_owner, p1)

Organizational resources were usually controlled by the owner.

Large industrial enterprises of XX century are characterized by further increase in number of managerial positions structured hierarchically by superior-subordinate relations. Such organizations are often defined as mechanistic (Scott, 2001) and have the following typical characteristics: strong functional specialization, a high level of processes formalization, a hierarchical structure reinforced by a flow of information to the top of the hierarchy and by a flow of decisions/orders from the top. Responsibilities were clearly defined for every position in a hierarchy. In most organizations of this type responsibility for execution was separated from responsibilities to make decisions. Managerial positions differed in power to make decisions depending on the level in the hierarchy. Often, technological decisions were made by managers of lower levels (or even by dedicated positions to which also execution responsibilities were assigned), whereas managerial decisions were made by managers at the apex. For example, the following formal expression identifies one of the upper managers responsible for making strategic decisions related to process p, one of the middle level managers responsible for making tactical decisions related to p and one of the first level managers responsible to making technological decisions related to p:

∃manager1: UPPER_MANAGER ∃manager2: MIDDLE_LEVEL_MANAGER ∃manager3: FIRST_LEVEL_MANAGER ∀γ: TRACE ∀t: TIME state(γ, t) |= [is_responsible_for(manager1, making_strategic_decisions, p) ∧ is_responsible_for(manager2, making_tactical_decisions, p) ∧ is_responsible_for(manager3, tech_des, p)]

In many of such organizations managers at the apex shared responsibility for making (some) decisions with lower-level managers. Therefore, decisions that were usually proposed by lower level managers had to be approved by the apex managers. In connection to the previous example the following superior-subordinate relations can be identified: is_subordinate_of_for(manager2, manager1, p) & is_subordinate_of_for(manager3, manager2, p).

Initially such enterprises operated in relatively stable (however, sometimes complex) environmental conditions that reinforced their structure. However, later in the second half of XX century to survive and to achieve goals in the changed environmental conditions (e.g., a decreased amount of external resources; increased competition; diversification of markets) enterprises and firms were forced to change their organizational structure and behavior. In response to the increased diversity of markets, within some enterprises specialized, market-oriented departments were formed. Such departments had much of autonomy within organizations. It was achieved by assigning to them the responsibility for most aspects related to processes, which created products/services demanded by the market. Although department heads still were subordinates of (apex) manager(s) of the organization, in most cases the latter one(s) were restricted only to general performance control over departments. Often departments controlled organizational resources necessary for the production and had the structure of hierarchical mechanistic type.

Although a hierarchical structure proved to be useful for coordination of activities of organizations situated in stable environments, it could cause significant inefficiencies and delays in organizations situated in dynamic, unpredictable environmental conditions. Furthermore, the formalization and excessive control over some (e.g., creative and innovative) organizational activities often can have negative effects on productivity. Nowadays, large enterprises often create project teams or task forces that are given complex, usually innovative and creative tasks without detailed descriptions/prescriptions. As in the case with departments, teams are often assigned the responsibility to make technological and (some) managerial decisions and are given necessary resources to perform their tasks. For example, the following formal expression represents the responsibility assignment to the team_A for making technological and strategic managerial decisions related to the process of development of a design for a new product.

∀γ: TRACE ∃t: TIME state(γ, t) |= [assigns_responsibility_to_for(management, team_A, tech_des, develop_design_new_product_A) ∧ assigns_responsibility_to_for(management, team_A, strategic_managerial_des, develop_design_new_product_A)]

Usually teams have highly cohesive plain structures with participants selected from different organizational departments based on knowledge, skills and experience required for the processes assigned to these teams. Although many teams implement informal communication and participative decision making principles (Lansley, Sadler and Webb, 1975), also formal authority relations can be found in teams. In particular, in some project teams superior-subordinate relations exist between the team manager and team members. In this case, whereas responsibility for making technological decisions is given to team members, the responsibility for most managerial decisions is assigned to the team manager. Then, the members of such teams, being also members of some functional departments or groups, have at least two superiors. In other teams the team manager plays the integrator role and does not have formal authority over team members. In this case the responsibility for decisions made by a team lies on all members of the team. Sometimes to strengthen the position of a team manager, s/he is given control over some resources (e.g., budgets) that can be used, for example, to provide material incentives to the team members.

The principles on which teams are built come close to the characteristics of the organic organizational form (Scott, 2001). Some of such organizations do not have any formal authority structure, other allow much flexibility in defining authority relations between roles. In the former case formal authority is replaced by socially created informal rules. In the latter case, authority may be temporally provided to the role that has the most relevant knowledge and experience for current organizational tasks. In many organic organizations formal control and monitoring are replaced by informal mutual control and audit. For the investigation of dynamics of organic organization, informal aspects such as influence, leaderships, mental models of employees are highly relevant, which will be discussed elsewhere. Often interactions between organic organizations (e.g., of network type) are regulated by contracts. Usually contracts specify legal relationships between parties that explicitly define their rights and responsibilities with respect to some processes (e.g., production, supply services). Several organizations may be involved in the process execution (e.g., supply chains for product delivery); therefore, it is needed to identify particular aspects of responsibility in contracts for such processes. The introduced language may be used for specifying such responsibilities and their legal consequences through reward/sanctions mechanisms.

4. Integration of Authority Relations into an EIS

In our previous work a general framework for formal organizational modeling and analysis is introduced (Popova and Sharpanskykh, 2007c). It comprises several perspectives (or views) on organizations, similar to the ones defined in the Generalized Enterprise Reference Architecture and Methodology (GERAM) (Bernus, Nemes and Schmidt, 2003), which forms a basis for comparison of the existing architectures and serves as a template for the development of new architectures. In particular, *the performance-oriented view* (Popova and Sharpanskykh, 2007b) describes organizational goal structures, performance indicators structures, and relations between them. *The process-oriented view* (Popova and Sharpanskykh, 2007a) describes task and resource structures, and dynamic flows of control. In *the agent-oriented view* different types of agents with their capabilities are identified and principles for allocating agents to roles are formulated. Concepts and relations within every view are formally described using dedicated formal predicate-based languages. The views are related to each other by means of sets of common concepts. The developed framework constitutes a formal basis for an automated EIS.

To incorporate the authority relations introduced in this paper into this framework, both syntactic and semantic integration should be performed. The syntactic integration is straightforward as the authority relations are expressed using the same formal basis (sorted predicate logic) as the framework. Furthermore, the authority relations are specified on the concepts defined in the framework (e.g., tasks, processes, resources, performance indicators). For the semantic integration rules (or axioms) that attach meaning, define integrity and other types of organization constraints on the authority relations should be specified. A language for these rules is required to be (1) based on the sorted predicate logic; (2) expressive enough to represent all aspects of the authority relations; (3) executable, to make constraints (axioms) operational. Furthermore, as authority relations are closely related to dynamic flows of control that describe a temporal ordering of processes, a temporal allocation of resources etc., a language should be temporally expressive. A language that satisfies all these requirements is the Temporal Trace Language (TTL). In (Sharpanskykh and Treur, 2006) it is shown that any TTL formula can be automatically translated into executable format that can be implemented in most commonly used programming languages.

In the following the semantic integration rules and several examples of constraints defined for particular organizations are considered.

The first axiom on the authority relations expresses that roles that are responsible for a certain aspect related to some process should be necessarily authorized for this:

Ax1: ∀r ROLE ∀a:PROCESS ∀aspect:ASPECT ∀γ:TRACE ∀t:TIME state(γ, t) |= [responsible_for(r, aspect, a) ⇒ authorized_for(r, aspect, a)]

Another axiom expresses the transitivity of the is_subordinate_of_for relation: r1: ROLE x r2: ROLE x a:PROCESS:

Ax2: ∀r1, r2, r3: ROLE ∀a:PROCESS ∀γ, t state(γ, t) |= [is_subordinate_of_for(r2, r1, a) ∧ is_subordinate_of_for(r3, r2, a)] ⇒ is_subordinate_of_for(r3, r1, a)]

One more axiom (**Ax3**) that relates the interaction (communication) structure of an organization with its authority structure based on superior-subordinate relations expresses that there should be specified a communication path between each superior role and his/her subordinate(s). Such a path may include intermediate roles from the authority hierarchy and may consist of both interaction and interlevel links.

The following axiom expresses that only roles that have the responsibility to make managerial decision with respect to some process are allowed to authorize other roles for some aspect of this process:

Ax4: ∀r1,r2:ROLE ∀a:PROCESS ∀asp:ASPECT ∀γ, t state(γ, t) |=
[authorizes_for(r1, r2, asp, a) ⇒ is_responsible_for(r1, manage_des, a)]

In general, rules that describe processes of authorization, assigning/retracting of responsibilities may have many specific conditions. However, to assign responsibility for some aspect of a process a role should necessarily have at least the responsibility to make managerial decisions and be the superior (with respect to this process) of a role, to which the responsibility is assigned. All other conditions may be optionally specified by the designer. Responsibility may be assigned on a temporal basis. To specify that a responsibility relation holds in all states that correspond to time points in the time interval limit, a responsibility persistency rule should be defined:

C1: ∀asp: ASPECT ∀r1,r2:ROLE ∀a:PROCESS ∀γ, ∀t1, t2:TIME state(γ, t1) |= is_responsible_for(r1, asp, a) & state(γ, t2) |= assigns_responsibility_to_for(r1, r2, asp, a) & (t1-t2) < limit

⇒ state(γ, t1+1) |= is_responsible_for(r1, asp, a)

Using concepts and relations from other organizational views, more complex constraints related to formal authority can be described. For example, “the total amount of working hours for role r1 should be less than a certain limit”:

C2: sum([a:PROCESS], case(∃t1 state(γ, t1) |= is_responsible_for(r1, execution, a), a.max_duration, 0)) < limit

This property can be automatically verified every time when roles are assigned additional responsibilities for some processes. This is particularly useful in matrix organizations (Scott, 2001), in which roles often combine functions related to different organizational formations (departments, teams), and, as a result, their actual workload may not be directly visible.

Another constraint expresses that when the execution of a process begins, for each of the basic aspects for this process (execution, tech_des, and manage_des) a responsible role should be assigned:

C3: ∀a:PROCESS ∀γ, t state(γ, t) |= process_started(a)
⇒ ∃r1,r2,r3: ROLE state(γ, t) |= [is_responsible_for(r1, manage_des, a) ∧ is_responsible_for(r2, tech_des, a) ∧ is_responsible_for(r3, execution, a)]

Another example is related to rewards/sanctions imposed on a role depending on the process execution results. As shown in (Popova and Sharpanskykh, 2007b), performance indicators (PIs) may be associated with organizational processes that represent performance measures of some aspects of the tasks execution. Depending on the PIs values, a company may have regulations to provide/impose some rewards/sanctions for roles (agents) responsible for the corresponding processes. Although such rules are rarely completely automated, still an EIS may signal to managers about situations, in which some rewards/sanctions can be applied. For example, the system may detect and propose a reward granting action to the manager, when a role has been keeping the values of some PI(s) related to its process above a certain threshold for some time period [period_start, period_end]. In TTL:

C4: ∀γ, t1 t1 ≥ perod_start & t1 ≤ perod_end & state(γ, t1) |= [is_responsible_for(r2, execution, a1) ∧ measures(PI1, a1) ∧ is_subordinate_of_for(r2, r1, a1) ∧ PI1.value > limit]
⇒ state(γ, period_end+1) |= grants_reward_to_for(r1, bonus_5_procent, r2, excellent_performance_of_a1)

The axioms Ax1-Ax4 can be checked on a specification of organizational formal authority relations. To this end, simple verification algorthims have been implemented. Whereas the constraints C1-C4 and similar to them need to be checked on actual executions of organizational scenarios (e.g., traces obtained from an EIS). An automated method that enables such types of analysis is described in (Popova and Sharpanskykh, 2007a).

Furthermore, the identified rules can be used to determine for each user of an EIS relevant to him/her information and a set of allowed actions that are in line with his/her (current) responsibilities defined in the system. Moreover, (possible) outcomes of each action of the user can be evaluated on a set of (interdependent) authority-related and other organizational constraints, and based on this evaluation the action is either allowed or prohibited.

5. Discussion

This paper makes the first step towards defining the formal operational semantics for power-related concepts (such as authority, influence, control), which are usually vaguely described in organization theory. In particular, this paper addresses formal authority, different aspects of which are made operational by defining a dedicated predicate logic-based language. It is illustrated how the introduced relations can be used for representing authority structures of organizations of different types.

Modern enterprises can be described along different dimensions/views: e.g., human-oriented, process-oriented and technology-oriented. However, most of the existing EISs focus particularly on the process-oriented view. An extension of the models on which EISs are built with concepts and relations defined within the human-oriented view allows conceptualizing more static and dynamic aspects of organizational reality, thus, resulting in more feasible enterprise models. Among the relations between human actors authority deserves a special attention, as it is formally regulated and may exert a (significant) influence on the execution of enterprise processes. This paper illustrates how the concepts and relations of authority can be formally related to other organizational views, thus resulting into an expressive and versatile enterprise model. The introduced authority relations may be also incorporated into other existing enterprise architectures that comply with the requirements of the GERAM (e.g., CIMOSA), based on which modern EISs are built. However, to enable semantic integration of the authority concepts, an EIS is required to have formal foundations, which are missing in many existing enterprise architectures and systems.

In the future it will be investigated how the proposed authority modeling framework can be applied for the development of automated support for a separation task (i.e., maintaining a safe distance between aircrafts in flight) in the area of air traffic control. Originally this task was managed by land controllers, who provided separation instructions for pilots. With the increase of air traffic, the workload of controllers rose also. To facilitate the controllers's work, it was proposed to (partially) delegate the separation task to pilots. This proposal found supporters and opponents both among controllers and pilots. The resistance to a large extent was (is) caused by ambiguity and vagueness of issues related to power mechanisms. Such questions as "whom to blame when an incident/accident occurs?", "which part of the task may be delegated?", "under which environmental conditions the task can be delegated?" still remain open. By applying the framework proposed in this paper one can precisely define responsibilities of both controllers and pilots and conditions under which the responsibility can be assigned/retracted. Notice that these conditions may include relations from different views on organizations (e.g., "current workload is less than x", "has ability a"), which allows a great expressive power in defining constraints.

References

Bacharach, S.B. & Aiken, M. (1977). Communication in administrative bureaucracies. Academy of Management Journal. 18, 365-377.

Bernus, P., Nemes, L. & Schmidt, G. (2003) Handbook on Enterprise Architecture. Springer-Verlag.

Blau, P.M. & Scott, W.R. (1962). Formal Organizations. Chandler Publishing.

CIMOSA. (1993). CIMOSA – Open System Architecture for CIM. ESPRIT Consortium AMICE. Springer-Verlag, Berlin.

Clegg, S.R. (1989). Frameworks of Power. London: Sage.

Friedrich, C.J. (ed.). (1958). Authority. Cambridge: Harvard University Press.

Gulick, L.H. & Urwick, L.F. (eds.). (1937). Papers on the Science of Administration. Institute of Public Administration, New York.

Hickson, D.J., Hinings,C.R., Lee,C.A., Schneck, R., & Pennings, J.M. (1971). A strategic contingency theory of intraorganizational power. Administrative Science Quarterly. 16, 216-229.

Jonker, C.M. & Treur, J. (2003). A temporal-interactivist perspective on the dynamics of mental states. Cognitive Systems Research Journal. 4, 137-155.

Lansley, P., Sadler, P.J., & Webb, T.D. (1975). Organization Structure, Management Style and Company Performance, Omega, London.

Manzano, M. (1996). Extensions of First Order Logic. Cambridge University Press.

Mintzberg, H. (1979). The Structuring of Organizations. Prentice Hall, Englewood Cliffs.

Parsons T. (1947). The institutionalization of Authority. In: Max Weber, The Theory of Social and Economic organization. Oxford University Press, New York.

Peabody, R.L. (1964). Organizational authority: superior-subordinate relationships in three public service organizations. Atherton Press, New York.

Popova, V. & Sharpanskykh, A. (2007a). Process-Oriented Organization Modeling and Analysis. In: J.C.Augusto, J. Barjis, U. Ultes-Nitsche (eds.), Proceedings of the 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS 2007), INSTICC Press, 114-126.

Popova, V. & Sharpanskykh, A. (2007b). Modelling Organizational Performance Indicators. In: Barros, F. et al. (eds.): Proceedings of the International Modeling and Simulation Multi-conference IMSM'07. SCS Press, 165-170.

Popova, V. & Sharpanskykh, A. (2007c). A Formal Framework for Modeling and Analysis of Organizations. In: Ralyte, J., Brinkkemper, S., Henderson-Sellers, B. (eds.), Proceedings of the Situational Method Engineering Conference, ME'07. Springer Verlag, 343-359

Scheer, A.-W. & Nuettgens. M. (2000). ARIS Architecture and Reference Models for Business Process Management. In: van der Aalst, W.M.P. et al. (eds.), LNCS 1806, Berlin, 366-389

Scott, W.R. (2001). Institutions and organizations. SAGE Publications. Thousand Oaks.

Sharpanskykh, A. & Treur, J. (2006) Verifying Interlevel Relations within Multi-Agent Systems. In Proceedings of the 17th European Conf. on AI, ECAI'06. IOS Press, 290-294.

Simon, H.A. (1957). Administrative Behavior. 2nd edn. Macmillan Co. New York

Weber, M. (1958). From Max Weber: Essays in Sociology. In: Gerth, H.H. and Wright Mills, C. (ed.) Oxford University Press, New York.