Automated Team Selection and Compliance Checking in Information Systems

Cristina Cabanillas
Vienna University of Economics and Business
Vienna, Austria
cristina.cabanillas@wu.ac.at

Manuel Resinas
Universidad de Sevilla
Sevilla, Spain
resinas@us.es

Jan Mendling
Vienna University of Economics and Business
Vienna, Austria
jan.mendling@wu.ac.at

Antonio Ruiz-Cortés
Universidad de Sevilla
Sevilla, Spain
aruiz@us.es

ABSTRACT
Plenty of activities in many business contexts must be performed collaboratively, e.g., in a hospital or when organising a conference. Tasks such as team composition and allocation are usually performed manually and on the ground of limited criteria such as individual skills, a.o. because adequate automatic support is missing. This paper addresses this shortcoming. We present an approach for team selection and compliance checking in process-aware information systems, which includes (i) a language for describing teams; (ii) a way to define team selection conditions and team-related policies; and (iii) a mechanism for the automatic resolution of the team selection conditions and for team-aware compliance checking based on formal ontologies.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous; D.2.8 [Software Engineering]: Metrics—complexity measures, performance measures

Keywords
PAIS, RALTeam, resource assignment, team selection, team-aware compliance checking

1. INTRODUCTION
Although resource management in Process-Aware Information Systems (PAIS) has gained increasing attention in recent years, there is hardly any approach that supports the assignment and allocation of a work item to teams that take part in a Business Process (BP). This comes as a surprise as there are areas such as healthcare where daily activities like surgeries typically require the availability of more than one person. As a consequence, standard workflow concepts cannot be directly applied since they assume a 1:1 relation between work item and worker.

The management of teams relates to a broad spectrum of issues, which are partially discussed in the area of agent and multi-agent systems [30], distributed systems [16] and social computing [21]. These include, e.g., team composition considering availability and preferences, constraints on team selection in relation to a task, conflicts of interest, optimal scheduling, or strategies to improve team performance. The problem is, however, that automatic support for the allocation of suitable teams to BP activities in process-oriented organisations is missing.

In this paper, we address this research problem by tackling team selection and team-aware compliance checking. Our contribution is a language for the description of teams called RALTeam, grounded on a team-aware organisational metamodel. Extending such language it is possible to define team selection conditions for assigning teams to process activities, and team-aware policies that specify constraints over the composition of teams in an organisation. The semantics of RALTeam are then formalized using Description Logics (DLs), which facilitates the automatic resolution of the selection conditions during process execution for team allocation, as well as the automatic checking of such conditions against the team-aware policies defined in the company in order to ensure compliance. All constructs of the language are motivated by projects that we have been involved with or which are discussed in the literature.

Against this background, the rest of the paper is structured as follows. Section 2 describes the research problem using a scenario from the healthcare domain. Section 3 presents an organisational metamodel that explicitly captures team-related concepts. Section 4 defines RALTeam as a language for team description. Section 5 explains how the language
can be extended to define team selection conditions and
team-related policies. Section 6 defines the semantics of the
language and the mechanism for team selection and compli-
ance checking. Section 7 outlines our proof-of-concept im-
plementation. Section 8 discusses related work before Section
9 concludes the paper.

2. MOTIVATING SCENARIO
One domain in which team work occurs is healthcare, from
which we adapt a scenario presented in [24]. Fig. 1 shows
the BP for patient diagnosis modelled with Business Process
Model and Notation (BPMN). Resource assignments are de-
finied in terms of organisational roles, along with conditions
that must hold for department members to participate in the
activities. First, the patient is registered by a clerk. Then,
a doctor and an assistant conduct a physical examination
while in parallel a nurse prepares the required documents.

Sometimes, further tests must be performed by the same
doctor and assistant with the help of a nurse. When these
activities are completed, the doctor assesses the results of
the test(s) and decides which information the nurse has to
give to the patient.

Now, let us assume that the process is executed within
the Department of Gynaecology (DoG), whose organisational
structure is shown in Fig. 2. It is organised on the ground of
a hierarchy of positions that are occupied by the members of
the department. The head is a doctor called Nick, who can
delegate work to all the resources occupying lower positions
in the hierarchy, i.e., to all the members of the department.
Below, there is an administrative assistant (Kate) and an-
other doctor (Marc), who report to the department head.
Subordinates of doctors are interns (Jane and Philip) and
nurses (Sue and Joe). The table attached to the hierarchy
shows the roles associated to each of the positions in this or-
ganisational unit, which typically establish the privileges for
the execution of activities and the access to data. Further-
more, as many activities in the department are collabora-
tive, there are some work teams already composed which are
usually directly used for assignment. Jane and doctor Nick
form a team called Perm_RE_1, where RE stands for Rout-
tine Examination. There is a rule in the hospital stating that
there must be at least one doctor in each team. In that
team, Nick plays the role of a coordinator and both of them
are implementers. Similarly, Philip is supervised by doctor
Marc, such that they form another team called Perm_RE_2.
There are also two teams of three members, Nick, Jane and
Sue form team Perm_AT_1, and Marc, Philip and Joe for
team Perm_AT_2. AT stands for Advanced Tests, and all
AT teams must have at least three members, a doctor and
a nurse among them.

The only accurate way to model resources in BPMN is by
means of its XML syntax, by default using XPath. Teams
are neither captured in the BPMN metamodel nor in often
widely used BP modelling notations. We have used the
BPMN Group and Text Annotation artifacts for the sake of
clarity. Nonetheless, there are constraints specified in the
description of the BP that could not be represented in the
model, nor can be defined with XPath, e.g., the fact that the
doctor that takes part in the performance of the advanced
tests is the same as the doctor conducting the physical ex-
amination, or the fact that the nurse delivering information
to the patient is the same who made the documents. Such
constraints are fundamental in order to select appropriate
teams or individuals. With the organisational structure and
the teams in the DoG, there are two possible teams for ac-
tivity Conduct physical examination, and two possible teams
for activity Conduct advanced tests. However, selecting a
proper team for activity Conduct advanced tests depends on
the team that conducted the examination in that specific BP
instance.

Another domain in which collaborative work is often found
is Software (SW) development, where several teams are usually
involved in the different SW development phases. For example,
the company responsible for the music service system
Spotify published team-based structure [20]. Team com-
position and selection is also fundamental (and critical) in
spaceflight and military missions. The NASA HRP BHP is
in charge of managing the risks related to team performance
and effectiveness in spaceflight missions [32]. The Team In-
tegrated Design Environment (TIDE) is a tool for the design
of teams for military missions [22]. Emergency services also
require team work. For instance, temporary teams are or-
dinary for police and firefighters which, furthermore, sometime
must also cooperate with teams from other organisations,
e.g., to battle a blaze distributed over a canyon ridge
[8]. We use these domains as reference in the design of the
team-aware organisational metamodel presented next.

3. TEAM-AWARE ORGANISATIONAL META-
MODEL

Elements related to teams must be part of the organisa-
tional metamodel of the company, such that the assignment
of teams to activities can be easily managed. To this end,
we take the organisational metamodel described by Russell
et al. [29] as a starting point, which covers people, capabil-
A team can have a type (class TeamType) that is associated with a specific configuration of the organisational roles. For example, in the motivating scenario there are teams Perm_REAL and Perm_REAL of type Routine Examination, composed of a doctor and an assistant. Team type Advanced Tests is made up of a doctor, an assistant and a nurse. More teams of these types could be created with the same role configuration. There could also be a team type Heart Surgery made up of two doctors, two assistants and one nurse, for instance. In this way, team types provide templates for the composition of teams.

Teams can also be structured hierarchically. For example, in SW development, there are often teams of SW Analysts (composed of persons with role Analyst), teams of SW Developers and teams of SW Testers. The team of analysts delegate work to the teams of developers, which report issues and results to the former and, in turn, delegate work to the team of testers. These report the results to the developers. In this context, modes of communication between teams have to be established, which we do not directly address here.

Finally, teams are also classified according to their temporality. A PermanentTeam is defined without an expiry date, e.g., all the teams defined in Section 2. Permanent teams can be referenced by their identifier at any moment. However, in certain occasions new teams are composed for specific purposes. For instance, in emergency surgeries teams are created, modified and broken up constantly depending on the requirements of the operations. Such teams are called TemporaryTeams because they have an expiry date defined as a specific scope. The scope can be (i) a specific period of time, e.g., a team active from August 1st to August 31st to provide support during the summer holiday break; (ii) it can be associated to a single activity instance, e.g., the execution of a single surgery; or (iii) it can be related to a process instance, so that the team can be treated as a single entity during the execution of the process instance because their participation could be required at any moment. Further team classifications are proposed in literature that mostly focus on how teams organise themselves (e.g., their coordination mechanism [26]). However, they are not included in the metamodel because our focus is on those aspects that are relevant for team selection in the context of resource assignment.

A person is a team creator if she is in charge of its configuration and of recruiting its members. She is not necessarily a member of the team, though. Besides, the figure of team creator is not mandatory, as teams may be automatically composed by a system according to some properties defined for them.

It is important to remark the difference between an OrganisationalUnit and a PermanentTeam. Although both are groups of people with an indefinite duration, the former is not an entity of collaborative work with a single goal by nature, but is composed of members that participate in different activities, each of which has a specific objective. In case of assigning concrete work to an organisational unit, it is because the unit is working as a team in the context of an activity or process, i.e., there is a new team made of the...
members of the organisational unit. For instance, if a hospital is organising an event, each department (i.e., organisational unit) could form a team working on the preparation of a specific issue (i.e., in that moment all their members have a common goal). Such a distinction has been described before [9].

### 4. RALTeam FOR TEAM DESCRIPTION

Resource Assignment Language (RAL) is a Domain Specific Language (DSL) for the definition of resource selection conditions, currently focused on human resources [6]. Its current version [7] focus on the grey excerpt of the metamodel in Fig. 3 and allows expressing a great variety of conditions with a syntax similar to natural language, such as:

**RAL1:** IS Samuel

**RAL2:** NOT (IS PERSON INVOLVED IN ACTIVITY RegisterP IN ANOTHER INSTANCE)

**RAL3:** (HAS ROLE Assistant) OR (HAS POSITION DoG_Doctor)

**RAL4:** SHARES SOME ROLE WITH PERSON IN DATA FIELD Test.Doctor

**RAL5:** (HAS UNIT DoG) AND (IS PERSON RESPONSIBLE FOR ACTIVITY MakeDocs)

We have defined an extension for RAL called RALTeam to describe teams according to the team-aware organisational metamodel. Similarly to RAL, RALTeam is composed of the expressions and constraints described next, whose Extended Backus-Naur Form (EBNF) syntax is shown in Language 1. In particular, it includes eight types of expressions (RALTeamExpr), whose configuration is supported by three types of constraints.

**TeamIDExpr** (line 7) allows directly indicating a team ID.

**TeamSizeExpr** (line 9) allows specifying the number of team members with a TCardinalityConstraint (line 30), e.g., WITH AT MOST 3 MEMBERS.

**TeamRoleExpr** (line 11) allows specifying a set of team role types for a team, i.e., team role types played by some of its members.

**TeamTypeExpr** (line 13) allows specifying the type of a team among three options: (i) a specific type, (ii) the same type as another team defined by a RALTeamExpr (line 14), or (iii) a type different than the type of another team defined by a RALTeamExpr (line 15).

**TeamCreatorExpr** (line 17) specifies the creator of a team using options similar to the TeamTypeExpr, plus one option described below.

**TeamScopeExpr** (line 22) allows specifying the scope of a team using options similar to the TeamTypeExpr,
the only difference being that the specific scope can be defined according to the three types of scopes described in the metamodel (cf. Fig. 3) with a `ScopeConstraint`.

`TeamCompoundExpr` (line 24) allows combining the aforementioned expressions with the AND and OR operators.

`TeamMemberExpr` (line 4) uses a `MembershipConstraint` to provide information about the team members, and optionally the team role type(s) that they play (line 28). Specifically, it allows specifying (i) a concrete person (line 37), resulting in sentences such as "WHOSE MEMBERS INCLUDE Marc or WHOSE MEMBERS INCLUDE Marc PLAYING TEAM ROLE TYPE Implementer"; or (ii) an amount of people, e.g., "WHOSE MEMBERS INCLUDE EXACTLY 2 PEOPLE PLAYING TEAM ROLE TYPE Implementer"; or (iii) a certain number of people with specific characteristics defined with `PeopleSelection` (line 38), which include:

- properties specified with a RAL expression (line 40), e.g., "WHOSE MEMBERS INCLUDE ONLY PEOPLE WHO HAVE UNIT DoG specifies that all the member of the team belong to DoG, where HAVE UNIT DoG comes from RAL and unit refers to an organisational unit. The link with RAL involves all the RAL expressions but one (IsAssignmentExpr)."

**Language 2** RALTeam for team selection and rule definition (EBNF)

```
RALTeamSelection := TEAM RALTeamExpr

RALTeamPolicy := TEAMS RALTeamExpr MUST [NOT] BE TEAMS RALTeamExpr
```

- people that do (or do not) belong to other teams defined by a `RALTeamExpr` (line 42).

It also introduces an option in `CreatorConstraint` (line 20) to provide more details about the team creator by means of a RAL expression.

### 5. RALTEAM FOR TEAM SELECTION AND RULE DEFINITION

The concepts of RALTeam as defined above offer a mechanism for team description. Let us now consider the required language concepts for team selection and rule definition. To this end, Language 2 introduces two additional expressions:

- `RALTeamSelection` (line 1) allows defining conditions for the selection of teams. Hence, it allows assigning teams to
BP activities by specifying the conditions that they must fulfill, e.g.:

TEAM CONTAINING TEAM ROLE TYPE Coordinator would return at least teams Perm_{RE,1} and Perm_{RE,2} in our scenario.

TEAM OF TYPE LIKE TEAM ((Perm_{RE,1}) OR (Temp_{AT,2})) selects teams Perm_{RE,2} and team Temp_{AT,1} according to our scenario.

TEAM WITH SCOPES ACTIVE DURING THE EXECUTION OF PROCESS bp1 selects all the permanent teams and the temporary teams whose scope fits with the one specified in the expression, i.e., teams active while bp1 is running.

TEAM (WITH AT LEAST 4 MEMBERS) OR (DF TYPE Advanced Tests) selects teams Perm_{AT,1} and Temp_{AT,2}.

ERALTeamPolicy. (line 4) allows defining policies related to teams that must hold in the organisation, such as:

TEAMS WITH BETWEEN 5 AND 10 MEMBERS MUST BE TEAMS-containing TEAM ROLE TYPE Coordinator.

TEAMS CREATED BY SOMEONE WHO HAS ROLE Assistant MUST NOT BE TEAMS OF TYPE Routine Examination.

Applying formal semantics to all the expressions described above, the resolution of the conditions for resource selection, which return a set of teams that are potential performers of a BP activity; and the checking of compliance between the existing teams and the policies defined by the company, can be automated.

6. AUTOMATED TEAM SELECTION AND COMPLIANCE CHECKING

Following the same approach as in RAL [5], RALTeam semantics are defined by means of a mapping to DLs [2]. Knowledge representation systems based on DLs involve two components: TBox and ABox. The TBox describes terminology, i.e., the ontology in the form of concepts and properties (relations between the concepts) and their relationships, while the ABox contains assertions about individuals (instances of concepts) using the terms from the ontology [2].

The mapping is a function \( \mathcal{T} \) that maps the team-aware organisational metamodel, its instantiation for a specific organisation and the RALTeam expressions to DL axioms and concept descriptions.

The mapping of the team-aware organisational metamodel is straightforward: metamodel classes and associations are mapped as concepts and properties in the Knowledge Base (KB), respectively, and cardinality restrictions are mapped as axioms such as Team \( \sqsupseteq 1 \text{hasTeamType.(TeamType)} \). There is only one consideration to this mapping. In the metamodel, the relationship between Person, Team and TeamRoleType is modelled with class TeamRole. However, DLs allow a more convenient way of expressing such a relationship by using hierarchies of properties. In this case the mapping involves adding a property hasMember from Team to Person and defining each TeamRoleType as a new subproperty of hasMember. In addition, a new property roleType is added from Team to TeamRoleType. This avoids introducing an "artificial" concept to define the ternary relationship of the metamodel, hence minimising the number of constructs as suggested in the Conceptualisation Principle described by ter Hofste and Proper [37].

The instantiation of the metamodel is mapped as follows. Class instances and their relations are mapped as individuals and relations between them except for TeamRole instances, which are mapped by means of hasMember sub-properties as described above in order to make it easier to build DL expressions with them. In addition, we assume that we have complete knowledge about the organisational model. Therefore, a mechanism to deal with the open world assumption of DLs should be provided. The open world assumption means that DLs assume that the knowledge may be incomplete, and hence, the absence of a property assertion stating that hasMember(Perm_{RE,1}^T, Jane) does not mean that Jane does not belong to team Perm_{RE,1}.

The solution proposed is an usual way to deal with the open world assumption, which involves that the mapping must include assertions that explicitly state that each individual has exactly the properties specified and no more (e.g. Team Perm_{RE,1} has exactly two hasMember relationships: Perm_{RE,1} \sqsubseteq 2 \text{hasMember(Person)}).

Finally, RALTeam expressions are mapped into DL concept descriptions, which are all subconcepts of Team and are defined in a way such that for every team \( t \) that satisfies a RALTeam expression expr, it holds that \( t^T \in \text{expr}^T \). Table 1 details the mapping for most RALTeam expressions. Expressions that involve TeamMemberExpr require an additional mapping (cf. Table 2) to obtain DL concepts from people selection expressions.

All of the DL concept descriptions used in this mapping belongs to the direct model-theoretic semantics of OWL 2, which extends the semantics of the description logics SROIQ with datatypes and punning [27]. In particular, note that the kind of reasoning used for date scopes in temporary teams do not require the use of temporal DL. Instead, dates are used as if they were integer numbers, i.e., simple datatypes. This means that any DL reasoner that can handle OWL 2 semantics can be used to reason about teams.

In fact, let \( \mathcal{K} \) be a KB obtained after mapping the elements of the team-aware organisational metamodel, its instantiation for a specific organisation and the RALTeam expressions using mapping \( \mathcal{T} \), both team selection and team compliance checking can be formulated in terms of standard DL reasoning tasks on \( \mathcal{K} \) that are implemented by most DL reasoners. In particular, two DL reasoning tasks are used, namely: concept retrieval, which is the problem of computing the set containing exactly every instance of a concept with respect to a KB \( \mathcal{K} \), and consistency, which is the problem of deciding whether a KB \( \mathcal{K} \) is consistent. We denote the former reasoning task as \( \text{individuals}_{\mathcal{K}} \) and the latter as \( \text{consistent}_{\mathcal{K}} \).

Team selection. This operation involves obtaining all teams defined in the organisation that satisfy a given RALTeam expression expr. Therefore, it can be expressed using the
We have evaluated the viability of the concepts covered by RALTeam with a prototypical application implemented using Java and the OWL API, and tested using the HermiT OWL reasoner. The mappings for all RALTeam expressions and the Java application can be found at [http://www.isa.us.es/cristal](http://www.isa.us.es/cristal). Using the implemented concepts, we are able to express the team assignments of our motivating scenario as follows:

Activity "Conduct physical examination" must be performed by an RE team can be defined using RALTeam as follows:

\[
\begin{align*}
\text{TEAM OF TYPE RoutineExamination} & \\
\text{AT LEAST 1 PERSON WHO } ps \text{ PLAYING TEAM ROLE TYPE } teamRoleTypeId \text{ (ps')} \\
\end{align*}
\]

The selection of teams that fulfill this RALTeam expression can be done by means of the following DL reasoning task:

\[
\text{individuals}_{\text{R}}(\exists \text{hasType}.\{\text{RoutineExamination}\})
\]

Activity "Conduct advanced tests" must be done by an AT team whose doctor took part in activity Conduct physical examination in that BP instance\(^1\) can be defined using RAL-Team as follows:

\[
\begin{align*}
\text{(TEAM OF TYPE AdvancedTests) AND } \text{(TEAM( WHOSE MEMBERS INCLUDE (AT LEAST 1 PERSON WHO \((\text{IS ANY PERSON INVOLVED IN ACTIVITY ConductPhysicalExamination}) \text{ AND (HAS ROLE Doctor)})\)))}
\end{align*}
\]

whose team selection can be done by means of the following DL reasoning task:

\[
\text{individuals}_{\text{R}}(\exists \text{hasType}.\{\text{AdvancedTests}\}) \cap \\
\text{((Team } \cap \exists \text{hasMember}(\text{IS ANY PERSON...}^{\text{R}})) \cap \\
\text{(Team } \cap \exists \text{hasMember}(\text{HAS ROLE Doctor}^{\text{R}})))}
\]

\(^1\)This selection condition has been shortened due to space limitations (cf. Section 2).

### Table 1: Excerpt of the mapping of the RALTeam expressions to DL concepts

<table>
<thead>
<tr>
<th>Type</th>
<th>RALTeam Expr ((expr))</th>
<th>DL Concept Description ((expr^T))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team</td>
<td>teamID</td>
<td>{teamID}</td>
</tr>
<tr>
<td>Size</td>
<td>AT LEAST n MEMBERS</td>
<td>\text{Team} \geq n \text{hasMember}</td>
</tr>
<tr>
<td></td>
<td>EXACTLY n MEMBERS</td>
<td>\text{Team} = n \text{hasMember}</td>
</tr>
<tr>
<td></td>
<td>BETWEEN n AND m</td>
<td>\text{Team} \geq n \text{hasMember} \land \leq m \text{hasMember}</td>
</tr>
<tr>
<td>Role</td>
<td>TEAM ROLES typeList</td>
<td>\exists \text{roleType}({\text{typeList}})</td>
</tr>
<tr>
<td>Type</td>
<td>teamTypeID</td>
<td>\exists \text{hasType}.{\text{teamTypeID}}</td>
</tr>
<tr>
<td></td>
<td>LIKE (expr)</td>
<td>\exists \text{hasType}.{\exists \text{hasType}.{\text{expr}}}</td>
</tr>
<tr>
<td></td>
<td>UNLIKE (expr)</td>
<td>\text{Team}(\neg \exists \text{hasType}.{\exists \text{hasType}.{\text{expr}}})</td>
</tr>
<tr>
<td>Creator</td>
<td>personId</td>
<td>\exists \text{hasCreator}.{\text{personId}}</td>
</tr>
<tr>
<td>Scope</td>
<td>ACTIVE BETWEEN start AND end</td>
<td>\text{PermanentTeam} \sqcap \text{formedWithin}.(\text{TemporalScope}) \sqcap (\text{start} \leq \text{startDate}) \land (\text{endDate} \geq \text{end})</td>
</tr>
<tr>
<td></td>
<td>ACTIVE DURING THE EXECUTION OF PROCESS pId</td>
<td>\text{PermanentTeam} \sqcap \text{formedWithin}.(\text{ProcessInstanceScope} \sqcap \exists \text{pId})</td>
</tr>
<tr>
<td>Comp.</td>
<td>expr1 AND expr2</td>
<td>\text{expr}_1 \sqcap \text{expr}_2</td>
</tr>
<tr>
<td></td>
<td>expr1 OR expr2</td>
<td>\exists \text{expr}_1 \sqcap \exists \text{expr}_2</td>
</tr>
<tr>
<td>Member</td>
<td>ONLY PEOPLE WHO ps</td>
<td>\text{Team}(\forall \text{hasMember}.{\text{ps}})</td>
</tr>
<tr>
<td>AT LEAST 1 PERSON WHO ps PLAYING TEAM ROLE TYPE teamRoleTypeId</td>
<td>\text{Team}(\exists \text{hasMember}.(\text{ps}'))</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: Mapping of RALTeam PeopleSelection to DL concept descriptions. Function \(\text{\text{R}}\) stands for the RAL mapping detailed in [5]

<table>
<thead>
<tr>
<th>People selection ((ps))</th>
<th>DL Concept Description ((ps'))</th>
</tr>
</thead>
<tbody>
<tr>
<td>SelectionExpr</td>
<td>SelectionExpr^R</td>
</tr>
<tr>
<td>IS MEMBER OF expr</td>
<td>\exists \text{hasMember}^-.{\text{expr}}</td>
</tr>
<tr>
<td>IS NOT MEMBER OF expr</td>
<td>\text{Person} \sqcap \neg \exists \text{hasMember}^-.{\text{expr}}</td>
</tr>
</tbody>
</table>

### 7. PROOF OF CONCEPT

We have evaluated the viability of the concepts covered by RALTeam with a prototypical application implemented using Java and the OWL API, and tested using the HermiT OWL reasoner. The mappings for all RALTeam expressions and the Java application can be found at [http://www.isa.us.es/cristal](http://www.isa.us.es/cristal). Using the implemented concepts, we are able to express the team assignments of our motivating scenario as follows:

Activity "Conduct physical examination" must be performed by an RE team can be defined using RALTeam as follows:

\[
\begin{align*}
\text{TEAM OF TYPE RoutineExamination} & \\
\text{AT LEAST 1 PERSON WHO } ps \text{ PLAYING TEAM ROLE TYPE } teamRoleTypeId \text{ (ps')} \\
\end{align*}
\]

The selection of teams that fulfill this RALTeam expression can be done by means of the following DL reasoning task:

\[
\text{individuals}_{\text{R}}(\exists \text{hasType}.\{\text{RoutineExamination}\})
\]

Activity "Conduct advanced tests" must be done by an AT team whose doctor took part in activity Conduct physical examination in that BP instance\(^1\) can be defined using RAL-Team as follows:

\[
\begin{align*}
\text{(TEAM OF TYPE AdvancedTests) AND } \text{(TEAM( WHOSE MEMBERS INCLUDE (AT LEAST 1 PERSON WHO \((\text{IS ANY PERSON INVOLVED IN ACTIVITY ConductPhysicalExamination}) \text{ AND (HAS ROLE Doctor)})\)))}
\end{align*}
\]

whose team selection can be done by means of the following DL reasoning task:

\[
\text{individuals}_{\text{R}}(\exists \text{hasType}.\{\text{AdvancedTests}\}) \cap \\
\text{((Team } \cap \exists \text{hasMember}(\text{IS ANY PERSON...}^{\text{R}})) \cap \\
\text{(Team } \cap \exists \text{hasMember}(\text{HAS ROLE Doctor}^{\text{R}})))}
\]

\(^1\)This selection condition has been shortened due to space limitations (cf. Section 2).
These assignments can then be used by the hospital to support team selection and scheduling at run time. The rules mentioned in Section 2 are defined as follows:

There must be at least one doctor in each routine examination team:

Teams of Type RoutineExamination Must Be Teams Whose Members Include At Least One Person Who Has Role Doctor

In DL, this involves adding the following axiom to the KB $K$:

$$\exists \text{hasType} \langle \text{RoutineExamination} \rangle \sqsubseteq (\text{Team} \cap \exists \text{hasMember} \langle \text{HAS ROLE Doctor}^R \rangle)$$

All advanced tests teams must have at least three members.

Teams of Type AdvancedTests Must Be Teams With At Least 3 Members

In DL, this involves adding the following axiom to the KB $K$:

$$\exists \text{hasType} \langle \text{AdvancedTests} \rangle \sqsubseteq (\text{Team} \cap \geq 3 \text{hasMember})$$

Finally, compliance between the teams defined in the company and these rules can then be checked with the DL query: consistency.

8. RELATED WORK

The necessity to deal with individual and collaborative tasks in the same environment has been identified and some partial solutions have been proposed in Computer Supported Collaborative Work (CSCW) [28, 18]. In the OSSAD method, collaborative tasks are supported by the concept of “horizontal macro-operation” [28]. None of the approaches found in this field pursues our goal. We tackle the challenge identified by in a survey on team work over the past fifty years [30] related to the assignment of teams to activities, i.e., team selection. Their notion of adaptive teams is closely related to our concept of temporary team. Next, we discuss approaches related to team work from several domains.

Team Modelling: STEAM [35] defines an organisational metamodel to support hierarchies of teams, composed of individuals. Both teams and people can be associated to roles according to their capabilities. Roles can be persistent or task-specific. Tambe et al. [36, 19] investigated how that metamodel worked in building agent-teams in the simulation league for Robocup, and how agents learn specific skills. Van der Aalst and Kumar focused on modelling organisational structures and work distribution in the context of team work [38]. Their team_type is our TeamRole, their team_position is our Role, and their role is our Position. Temporality in teams is not considered in their approach. Dustdar developed Caramba [14], a PAIS to integrate artifacts, resources and processes [15] that emphasizes communication and interaction but disregards teams.

Team Composition and Selection: Most approaches dealing with team composition and selection address the problem of finding the best match of experts to required skills [14, 17, 3]. In this context, several approaches study connectivity and social aspects for team composition, e.g., social distance between people [39]. Dorn et al. [12] highlight physical location and communication capabilities between team members as relevant. They present an approach for deriving user profiles from social networks and create virtual teams in which there is balance between skills and connectivity. This is extended towards a skill-dependent recommendation model for team composition [13]. Some other approaches considering both skills and connectivity are [21, 33, 10]. RALTeam takes into account skills and geopositions of people. Social connectivity is not considered due to its intra-organisational focus, but it could be extended to deal with social aspects as well. Some of the Advanced Resource Patterns (ARPs) described by Meyer [25] are related to team selection, namely Single Entity and Restricted Team Size. Both are supported by RALTeam, as it treats teams as a single entity for resource assignment and allows defining the team size with the TeamSizeExpr.

Team Allocation: Partially orthogonal to our work is team allocation. Mans et al. introduced an approach [24] that allocates people to BP activities considering their calendars, the calendars of the people they have to collaborate with in the BP activities, and the ongoing execution of the BP, so that everything is completed on-time. This approach is also used in Procles [23], a framework that provides support for the modelling and execution of “non-monolithic” processes, i.e., unstructured processes with complex interactions between participants, where activity execution is sometimes collaboratively performed by several people. Such features are not supported by most of the current PAIS. Our approach could be combined with schedule-based allocation approaches.

Team Cooperation and Performance: Several literature reviews and surveys have been conducted on this topic [9, 34, 31]. Moe et al. argued that traditional teams follow a plan-driven model, whereas self-managing agile teams face change-driven development. They studied work cooperation and performance in self-managed agile teams [26], applying the Dickinson and McIntyre’s team work model [11] to a real case where teams used Scrum. Caramba [14] supports the collaboration of virtual teams in adaptive workflow management systems, i.e., processes that are not perfectly defined from the beginning but are reconfigured on-the-fly. The inContext Pervasive Collaboration Services Architecture (PCSA) [16] aims at supporting highly dynamic forms of human collaboration such as Nimble (short-lived collaboration), Virtual (spanning different geographical places) and Mobile (collaboration with mobility capabilities) teams.

9. CONCLUSIONS AND FUTURE WORK

The integration of team work in business processes is still limited. In this paper, we have addressed this research problem by introducing a language to describe teams and its applicability for the definition of team selection conditions and team-related policies. The DL-based semantics of the language have been used to automate the resolution of team selection conditions and for compliance checking with team
We deem our approach not only relevant from a research angle, but it paves the way for automatically resolving higher level queries with strong practical applications such as “do we have the necessary human resources to conduct a surgery on trauma?”. We aim to conduct case studies in different domains to identify those higher level queries and to further validate RALTeam expressiveness. Furthermore, extending RALTeam to support on-the-fly team composition at run time, the composition and selection of virtual or distributed teams, and the integration of these results with other approaches such as schedule-aware workflow management systems [24] are part of our planned future work as well.

10. REFERENCES


