RNS – A Schema for Specifying Computational Autonomy

Matthias Nickles, Michael Rovatsos, Gerhard Weiß

Computer Science Department
Technical University of Munich
{nickles,rovatsos,weissg}@in.tum.de
Roadmap

- Motivation
- Basic Specification Constructs of RNS
- Specifying Norms and Sanctions with RNS
- Specifying Activities with RNS
- Discussion
- Conclusion
Motivation

Autonomy is a key property of computational agency
Motivation

Autonomy is a key property of computational agency

Potential to be enabling technology for broad range of important applications
- telecommunications, logistics, e/m-commerce, pervasive and ubiquitous computing
- open, dynamic, networked, decentralized, unpredictable applications
Motivation

- Specification of kind and level of autonomy is most critical engineering challenges

- Autonomy specification dilemma:
  - too rigid → suppression of necessary action choice
  - too generous → admission of unnecessary action choice

Strong need for techniques (methods, formalisms, tools, etc.) to specify computational autonomy

RNS ("Roles, Norms, Sanctions") developed in response to this need
Motivation

- Specification of kind and level of autonomy is most critical engineering challenges

- Autonomy specification dilemma:
  - too rigid → suppression of necessary action choice
  - too generous → admission of unnecessary action choice

- Strong need for techniques (methods, formalisms, tools, etc.) to specify computational autonomy

- RNS ("Roles, Norms, Sanctions") developed in response to this need
Agents are embedded in a social frame which regulates (but does not fully constrain) their behavior.

The social frame, called role space, is composed of roles:

```
ROLE_SPACE role_space_id { role_id_list }
```

where

- `role_space_id` = unique role space identifier
- `role_id_list` = list of unique role identifiers
Agents must act as role owners, they can try to achieve their goals through playing roles

Conceptually, roles

- serve as a means for specifying desired behavior
- are *not* viewed as a means for fully constraining behavior (they leave room for individuality!)
Agents must act as role owners, they can try to achieve their goals through playing roles

Conceptually, roles
- serve as a means for specifying desired behavior
- are *not* viewed as a means for fully constraining behavior (they leave room for individuality!)

Specifying a role requires to specify the activities that are of relevance to an agent playing this role

```
ROLE role_id { activity_list }
```
Agents must act as role owners, they can try to achieve their goals through playing roles.

Conceptually, roles serve as a means for specifying desired behavior, are *not* viewed as a means for fully constraining behavior (they leave room for individuality!)

Specifying a role requires to specify the activities that are of relevance to an agent playing this role:

```
ROLE role_id { activity_list }
```

Specifying an activity requires to specify the norms and sanctions the activity is subject to.
Pairwise specification of norms and sanctions:

\[ \text{NORM} \langle \text{norm\_type} \rangle \langle \text{condition} \rangle \; + \; \text{SANC} \langle \text{sanction\_type} \rangle \langle \text{sanction} \rangle \]

- Norm specification
- Sanction specification
- Norm-sanction pair
Pairwise specification of norms and sanctions:

\[
\text{NORM} \ <\ \text{norm}\_type> \ <\ \text{condition}> \ + \ \text{SANC} \ <\ \text{sanction}\_type> \ <\ \text{sanction}>
\]

- **norm specification**
- **sanction specification**
- **norm-sanction pair**

Three types of norms:
- permissions (P), “agent *may* do x”
- obligations (O), “agent *must* do x”
- interdictions (I), “agent *must not* do x”
Pairwise specification of norms and sanctions:

\[
\text{NORM} \langle \text{norm\_type} \rangle \langle \text{condition} \rangle \; + \; \text{SANC} \langle \text{sanction\_type} \rangle \langle \text{sanction} \rangle
\]

Three types of norms:
- permissions (P), “agent may do x”
- obligations (O), “agent must do x”
- interdictions (I), “agent must not do x”

Two types of sanctions:
- reward (RE) of norm-conforming behavior
- punishment (PU) of norm-deviating behavior
Two types of norm-sanction pairs
Two types of norm-sanction pairs

- “DEPENDENT”: become relevant after explicit request for (not) executing the activity

```<DEP role_id> : norm-sanction_pair```

status type
Two types of norm-sanction pairs

- "DEPENDENT": become relevant after explicit request for (not) executing the activity

  \[
  <\text{DEP} \ role\_id> : \text{norm-sanction\_pair}
  \]

  status type

- "INDEPENDENT": valid regardless of requests

  \[
  <\text{IND}> : \text{norm-sanction\_pair}
  \]

  status type
Specifying Norms and Sanctions

- General syntactic form of dependent and independent norm-sanction pairs:
  \[
  \text{status\_type} : \text{norm-sanction\_pair}
  \]

- Specifications of this form are called status statements
Specifying Norms and Sanctions

General syntactic form of dependent and independent norm-sanction pairs:

<status_type> : norm-sanction_pair

status_statement

Specifications of this form are called status statements

The list of all status statements attached to an activity is called the activity’s status range

STATUS_RANGE status_statement_list
Specifying Norms and Sanctions

General syntactic form of dependent and independent norm-sanction pairs:

\[
\text{status statement} : \text{status_type} : \text{norm-sanction_pair}
\]

Specifications of this form are called status statements.

The list of all status statements attached to an activity is called the activity’s status range.

Example:

\[
\text{STATUS RANGE status_statement_list}
\]

\[
\text{STATUS RANGE}
\]

\[
\text{<IND> : NORM <P> <NO> + SANC <NO> <NO>}
\]

\[
\text{<DEP EACH> : NORM <O> <quantity \leq 100> + SANC <PU> <withdraw_role>}
\]

\[
\text{<DEP AssemblyMg> : NORM <I> <material = \text{steel}> + SANC <PU> <pay_fine>}
\]
Specifying Activities

Four types of activities

- *basic activities*, i.e. resource and event handling
- *execution request activities*, i.e. requests for executing activities
- *sanctioning activities*, i.e. activities that result in punishment (reward) of norm-deviating (norm-conforming) behavior
- *change activities*, i.e. activities that result in changes of norms and/or sanctions
Specifying Activities

Four types of activities

- *basic activities*, i.e. resource and event handling
- *execution request activities*, i.e. requests for executing activities
- *sanctioning activities*, i.e. activities that result in punishment (reward) of norm-deviating (norm-conforming) behavior
- *change activities*, i.e. activities that result in changes of norms and/or sanctions

Complex activity constructs are possible, e.g. requests for sanction, requests for requests (for requests for ...), requests for changes of norms, ...
Basic activities have the general syntax

```plaintext
ACT activity_id ( activity_variable_list )
   { STATUS_RANGE status_statement_list }
```

where

```plaintext
STATUS_RANGE = as explained above
```
Basic activities have the general syntax

\[
\text{ACT } \text{activity\_id} (\text{activity\_variable\_list}) \\
\{ \text{STATUS\_RANGE status\_statement\_list} \}
\]

where

\[
\text{STATUS\_RANGE} = \text{as explained above}
\]

Example of a basic activity

\[
\text{ACT deliver} (\text{material,quantity}) \\
\{ \text{STATUS\_RANGE} \\
\langle \text{IND} \rangle: \text{NORM} <\text{P}> <\text{NO}> + \text{SANC} <\text{NO}> <\text{NO}> \\
\langle \text{DEP EACH} \rangle: \text{NORM} <\text{O}> <\text{quantity} \leq 100> + \text{SANC} <\text{PU}> <\text{withdraw\_role}> \\
\langle \text{DEP AssemblyMg} \rangle: \text{NORM} <\text{I}> <\text{material} = \text{steel}> + \text{SANC} <\text{PU}> <\text{pay\_fine}> \\
\}
\]
Execution request activities have the general syntax

\[
\text{ACT REQUEST} ( \text{agent_id_list} ; \text{role_id_list} ; \text{[NOT]} \text{ activity_id} ( \text{activity_variable_list} ) )
\{ \text{STATUS RANGE} \text{ status_statement_list} \}
\text{NORMATIVE IMPACT} \text{ norm_specification_list} \}
\]

where

- \text{NORMATIVE IMPACT} = \text{normative impact of the request on the requested agent(s)}
- \text{norm_specification_list} = \text{list of norm specifications of the form}
  \[
  \text{NORM} <\text{norm_type}> <\text{condition}>
  \]
Specifying Activities

**Execution request activities** have the general syntax

```
ACT_REQUEST (agent_id_list; role_id_list; [NOT] activity_id (activity_variable_list))
{ STATUS_RANGE status_statement_list
  NORMATIVE_IMPACT norm_specification_list }
```

where

- **NORMATIVE_IMPACT** = normative impact of the request on the requested agent(s)
- **norm_specification_list** = list of norm specifications of the form
  ```
  NORM <norm_type> <condition>
  ```

**Example of an execution request activity**

```
ACT_REQUEST (EACH; USsupplier, EUROsupplier; NOT deliver (material, quantity))
{ STATUS_RANGE
  <IND>: NORM <P> <(material = steel) AND (rating(material) = poor)> +
  SANC <NO> <NO>
  NORMATIVE_IMPACT
  NORM <I> <material = steel>
}
```
Specifying Activities

Sanctioning activities have the general syntax

```
ACT SANCTION ( agent_id_list ; role_id_list ; activity_id ; norm_spec )
{ STATUS_RANGE status_statement_list
  SANCTIONING_IMPACT sanction_specification_list }
Specifying Activities

Sanctioning activities have the general syntax

```
ACT SANCTION ( agent_id_list ; role_id_list ; activity_id ; norm_spec )
{  STATUS_RANGE status_statement_list
   SANCTIONING_IMPACT sanction_specification_list }
```

where

```
SANCTIONING IMPACT = list of sanction specifications of the form
   SANC <sanction_type> <sanction>
```

Example of a sanctioning activity

```
ACT SANCTION ( EACH ; EACH ; deliver ; NORM <O> <quantity < 100> )
{  STATUS_RANGE
   <IND> : NORM <P> <NO> + SANC <RE> <earn_bonus>
   <DEP RoleSpaceMg> : NORM <I> <NO> + SANC <PU> <withdraw_role>
   SANCTIONING_IMPACT
   SANC <PU> <withdraw_role>
}
```
Specifying Activities

Change activities result in changes of norms and/or sanctions by modifying the status range of activities.

Three types of changes:

- **DEL** (delete): a status statement is removed from the status range of an activity.
- **REP** (replace): a new status statement replaces an existing one.
- **ADD** (add): a new status statement is added.

Formal syntax and examples of **DEL**, **REP**, and **ADD** straightforward (cf. paper).
RNS offers several useful features

- domain- and application-independent
- neutral w.r.t. autonomy (respects autonomy specification dilemma)
- grounded in sociological role theory
- strongly expressive
  - explicit specification of sanctions
  - explicit specification of changes
  - allows for different normative impacts of the same activity (“context sensitivity”)
  - supports “composed activities” (e.g. requests for changes)
Conclusion

Key differences from other approaches:

- expressiveness
- no assumptions on agent-level (cognitive) processes
- caters for norm-violating behavior
Conclusion

Key differences from other approaches:
- expressiveness
- no assumptions on agent-level (cognitive) processes
- caters for norm-violating behavior

RNS leaves room for improvement

Major deficiencies:
- relationships among roles (e.g., generalization, aggregation, inheritance) cannot be expressed
- no support for conflict identification and resolution (e.g., permission and interdiction of the same activity)

Currently: work on the above + development of specification tool