

The NNN Formalization: Review and Development of Guideline Specification in the Care Domain

Technical Report

Georg Kaes¹, Jürgen Mangler¹, Stefanie Rinderle-Ma¹, and
Ralph Vigne²

University of Vienna, Faculty of Computer Science, Austria
{georg.kaes, juergen.mangler, stefanie.rinderle-ma}@univie.ac.at
CERN, ralph.vigne@cern.ch

1 Introduction

This technical report describes a formalization for nursing knowledge found in the NANDA, NIC and NOC (NNN for short) standards.

Data about nursing diagnoses and treatments consists of the following three NNN knowledge sources that are widely accepted and used in the field of care [1], i.e., *NANDA*¹ containing all basic information about the diagnoses, the *Nursing Interventions Classification (NIC)* containing all treatments which can be executed for improving the patients condition, and the *Nursing Outcomes Classification (NOC)* containing all nursing outcomes which can be reached after a therapy.

Based on the NNN knowledge, our overall goal is to enable the guideline-driven process development and adaptation in the care domain. Within the ACaPlan² project, we work closely together with experts from the care domain. This paper contributes a first step towards this goal by providing a formalization method for the NNN knowledge sources in such a way that this information can be directly utilized for creation and adaptation of individual patient treatment processes.

The central aspects of designing a corresponding formalization are as follows:

Methodologically, we first analyze which information of the NNN knowledge sources has to be included in the formalization based on studying NNN documentation and discussions with experts from the care domain resulting in the NNN taxonomy (contribution 1). Then we evaluate existing standards, primarily from the medical domain, i.e., GLIF [2], Asbru [3], and ARDEN Syntax [4], with respect to their support for the NNN taxonomy, illustrated by the use case FATIGUE. The resulting evaluation report and open issues (contribution 2) serve as input for the design of the NNN formalization (contribution 3). The NNN formalization is evaluated based on use case FATIGUE, results from discussing with domain experts, and possible application in other domains.

In detail, NNN knowledge sources are introduced and analyzed in Section 2. Existing standards from the medical domain are evaluated in Section 3). Section 4 presents

¹ North American Nursing Diagnosis Association, www.nanda.org

² Adaptive Care Planning: <http://cs.univie.ac.at/research/projects/projekt/infproj/1033/>

the NNN formalization. Section 5 focuses on lessons learned and a discussion about further advantages which arise from our contribution.

2 NANDA, NIC, and NOC Knowledge Sources

The goal of NANDA, NIC and NOC is the development and unification of nursing diagnoses that are the basis for all processes related to care, as they contribute a consistent terminology and ease the phrasing and documentation [5]. NANDA contains 206 nursing diagnoses [6], which are available for care attendants in various text documents. Based on the nursing diagnoses defined in NNN, nurses can determine a patients state and in further steps are able to create therapies and define intended outcomes, which are used for treating the observed symptoms.

For getting an overview over the different sections and contents of a diagnosis we studied a variety of these documents ([7, 5, 8–10]) and conferred with domain experts who showed us the most relevant parts of the given taxonomy.

As a first step, we analyzed and aggregated the necessary building blocks of NNN, which are depicted in Figure 1. In the following, these building blocks are shortly described and illustrated by the use case FATIGUE.

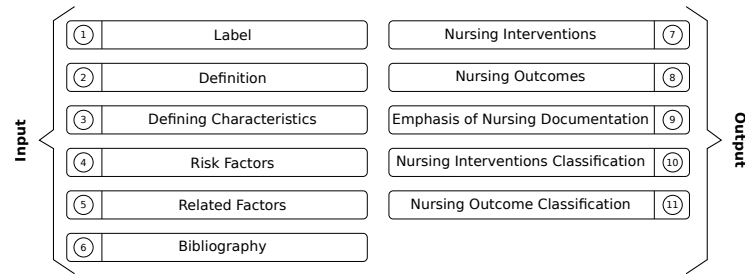


Fig. 1. Building Blocks for NNN Formalization

- ①, ②: provide identification of the diagnosis, e.g., FATIGUE, and describes the most important characteristics of the given diagnosis in one or more natural language sentences respectively. For FATIGUE, the description reads as follows: “An overwhelming sustained sense of exhaustion and decreased capacity for physical and mental work at usual level” [9].
- ③: are the symptoms which imply the fact that there is a hardship in the patients circumstances. Those characteristics, which can be either subjectively (“I feel a bit dizzy today.”) or objectively (“Your nose is bleeding.”) determined, are further informational assets for the care attendant to review or approve the diagnosis [7].
- ④: are possible causes, which may lead to the given nursing diagnosis [7]. It is a very important fact that care attendants should not concentrate on treating the symptoms a patient shows, but the causes which are the reasons for the current

circumstances. For FATIGUE, there are several potential risk factors defined, for example psychological causes such as stress, fear, or depression [7].

- ⑤: describe risk factors, defining characteristics for possible problems which may lead to future hardships for the patient. Diagnoses which contain these kind of factors only concentrate on risks, and therefore do not implement any defining characteristics (as there are none yet).
- ⑥: contains all sources which led to the development of the given taxonomy. They have to be linked to the appropriate part of the diagnosis.

After defining all information sources related to the current state of the patient, the second class of information refers to the possible treatments, their outcome, and their documentation:

- ⑦: are the concrete tasks which need to be executed for reaching the desired nursing goals given the current circumstances. They have to be defined in a standardized terminology like the Nursing Interventions Classification, or supplemented with resources.
- ⑧: define the desired outcomes after the execution of the therapy. As with Nursing Interventions, they either have to be defined within a standardized terminology like the Nursing Outcomes Classification, or supplemented with sources.
- ⑨: are not necessarily part of the NIC taxonomy itself, but nonetheless an important part of a complete definition of a given diagnosis. They are listed explicitly in specialized literature such as [7]. The emphases of nursing documentation define the most important facts for the documentation of the executed tasks. Based on the necessity of a situational and regular evaluation of the performed actions, the possibility to totally reconstruct the methods of the acting care attendants is very important. For the diagnosis of FATIGUE there exist several categories, which contain the necessary documentational elements.
- ⑩ and ⑪: list the relevant categories from the Nursing Interventions classification respectively the Nursing Outcomes Classification.

3 Evaluation of Existing Standards: GLIF, Asbru, ARDEN Syntax

Similarly to NNN knowledge supporting the work of care personnel, Clinical Practice Guidelines (CPGs) define how medical staff has to act in certain situations and are an essential part of modern medicine. During the last decades several approaches have been developed aiming at formalizing this medical knowledge in a manner which is also accessible to computers. The goal of these Computer-Interpretable Guidelines (CIGs) [11] which are a formal representation of the CPGs, is helping the doctors with their decisions in the best way possible.

In the following we evaluate three well known approaches - namely the Arden Syntax, Asbru, and GLIF as representatives for CIG standards. We chose three well known approaches which we have found in specialized literature [12] [13] very often. Of course, there do exist more approaches, like the *Evicare* project, which focuses on “providing evidence-based medicine at the point of care” [14], thus increasing quality of patient care. Their guideline formalizations are based on the *DeGeL* framework, whose

model supports elements common to clinical guidelines. [15] shows how guidelines defined in Asbru and GEM can be implemented in DeGeL.

In this section, we describe the design of the Arden Syntax, Asbru and GLIF and try a formalization of NNN in the particular approach based on the example of FATIGUE. The result is a discussion of limitations of existing CIG standards with respect to formalizing care-related diagnoses.

3.1 The Arden Syntax

The Arden Syntax is a guideline formalism implementing a language close to Pascal with the goal of formalizing medical knowledge. The basic elements are called Medical Logic Moduls (MLMS), which can be reused in several applications [16, 17]. Representing knowledge in distinct, separate modules facilitates the implementation of contents relevant to the respective institution into their own *Electronic Patient Record-System*.

Design Principle: Medical Logic Modules are the basic elements of the Arden Syntax. They contain the structure how knowledge is represented and give the medical professionals - if they are implemented in a clinical information system - informations about a patient's condition by using alerts [4].

MLMs consist of the three section *Maintenance*, *Library*, and *Knowledge*. The latter contains the actual medical knowledge encoded within different slots, i.e., key-value pairs where the value can be either text, a coded value, or structured data. The following MLM excerpt for section Knowledge implementing NANDA diagnosis FATIGUE demonstrates its basic structure (note that some simplifications were made).

Listing 1.1. Component: Knowledge

```
type: data-driven;;
data:
listless := READ {select listlessness from results where it occurred within the
  past 1 week};
features := READ {select feature from results where TirednessIndicator = true};
priority: 42;
evoke: ANY OF (listless, features)
logic:
IF features='very tired' OR listless > 84 conclude
action:
WRITE 'Assess the patient's ability to perform activities of daily living' TO
  nurse_infoscreen
```

The knowledge component contains the medical knowledge. In the data slot we have defined two variables. The first one saves a fictive indicator for listlessness, which is going to be read from a database, the second one saves a corresponding value in the key *features*. The MLMs priority is set to 42, and the MLM will be, as defined in the evoke slot, evoked as soon as one of those two variables is set. The logic slot defines the rule, which has to be evaluated for executing the action slot, if it returns true. In this example, the evoke slot returns true, if the feature is “very tired”, or the value on the listlessness scale is bigger than 84. The action defines to write an appropriate message to an info screen (named *nurse_infoscreen*).

According to the building block defined in Fig. 1 we summarize as follows:

- ① is best placed into the *title slot*.
- ② can be defined in the *explanation key* of the *Library* section.
- ③ can be considered as part of *evoke* → *keywords slot*
- ④, ⑤, ⑩ and ⑪ are not supported at all.
- ⑥ has its counterpart in the *citations slot*.
- ⑦ and ⑨ are matching with *action slots*. Although the intention is similar it should be noted that there is one major conceptual difference: the *all-or-nothing* approach of ARDEN. This makes NNN definitions very complex as each separate ⑦ spawns a new MLM.
- ⑧, although it is similar to the *library* → *purpose slot* the difference is still to big to be neglected

3.2 Asbru

Asbru is a modeling notation which emphasizes the temporal structures of medical plans [18]. The underlying *skeletal plan* leaves room for the temporal planning of the individual actions, which increases flexibility. This way, plans can be adapted to changing circumstances, for example if medical professionals are needed on other stations. Asbru consists of 2 phases. While during design phase, you can define timely confined actions and alternatives as reactions for conditions of any patient, during execution phase these resulting skeleton planes get instantiated for a concrete patient. This emphasizes the generation of practical plans. The language, which is based on XML and defined over a DTD shows a syntax close to Lisp.

Design Principles: Plans in Asbru consist of several elements such as *Preferences*, *Intentions*, and *Plan Body*. Plan hierarchies can be also defined [17]. With *AsbruView* a visualization approach for plans has been proposed [3].

For demonstrating the XML structure of Asbru, selected parts of the NNN diagnosis FATIGUE have been implemented in this notation. The following listing shows the implementation of an exemplary section of the *Intentions* part. The desired nursing outcome of this plan is to increase the energy level (a fictive scale for measuring the energy a patient currently has) of a patient to a value of at least 50 over 3 days.

Listing 1.2. Intentions

```
<intentions>
  <intention label="PowerEnergy" type="overall-state" verb="achieve" importance
    = "1">
    <temporal-pattern>
      <parameter-proposition parameter-name="patientEnergy">
        <value-description type="greater-than">
          <numerical-constant unit="E" value="50"/>
        </value-description>
        <context><context-ref name="patientEnergy"/></context>
      <time-annotation>
        <time-range>
          <duration>
            <minimum>
              <numerical-constant unit="d" value="3"/>
            </minimum>
          </duration>
        </time-range>
      </self/>
    </temporal-pattern>
  </intention>
</intentions>
```

```

        </time-annotation>
      </parameter-proposition>
    </temporal-pattern>
  </intention>
</intentions>

```

Although Asbru provides a flexible scheme for formalizing medical guidelines, it can only be of limited use to care related guidelines. In the following we list the main differences according to the building blocks defined in Fig. 1:

- ① can be defined in the *title attribute* of the plan itself.
- ③ can be defined in the *preconditions*, but it should be noted that doing so leads to the loss of support for precautions.
- ⑥ could be put into the *comment* tag which is applicable to a lot of tags.
- ⑦ and ⑨ are going to be defined as nested sub plans by using several *plan-bodys*.
- ⑧ match directly with *intentions* which additionally act as a container for ⑩.
- ②, ④, ⑤ and ⑩ can not be defined (leaving a misusing of the *comment tag* aside)

3.3 The GuideLine Interchange Format (GLIF)

GLIF (current version GLIF3) [2] has the goal of easing the interchange of guidelines between different institutions and platforms. It offers structures, which make it easy for users to understand the purpose of the guideline, as well as such, which are of use for decision support systems. In GLIF all classes are represented by UML class diagrams. The actual structure of the guidelines is defined in RDF. Other constraints are defined by in OCL [19].

Design Principles: GLIF consists of two components: the *GLIF model*, and the *GLIF syntax*. While guidelines are implemented as instances of the classes defined in the GLIF model, the representation of the knowledge covered by the guideline is specified in the GLIF syntax. *Guideline* is the class which describes a guideline. All general attributes, like the name, the author, the purpose, selection criteria (which are implemented as criterion objects), steps, the starting point, and further information (didactics) are part of this class. The class *Guideline_Collection* describes a collection of classes which belong together. The class *Supplemental Material List* can be used for describing further information about the class, like the documentation. The class *Guideline_Expressions* represents all kinds of expressions, from simple strings like “weight” to logical expressions, like “weight < 90”. Further on, GLIF proposes a 3-level-model consisting of the conceptual, computable, and the implementable level. All these levels give a different degree of abstraction, and can therefore be useful to different kinds of users (from real persons to computer programs).

For showing the structure of GLIF, again we have implemented selected parts of the NANDA diagnosis FATIGUE. In the following we show how the classes *Guideline*, *Action Step* and *Action Specification* are being implemented.

The class *Guideline* contains general information. In this example we have defined two criteria which have to evaluate to *true* before starting the guideline; these are *Lack-OfEnergy* and *NeedForEnergy*. The *Steps* defines the steps which are part of the guideline. In this case, only the two simple steps *MedicationWatch* and *DailyLivingWatch* are implemented. *First step* defines, that we first have a look on the medication.

Listing 1.3. Class: Guideline

```
name: Fatigue
author: Georg Kaes
intention: Capacity to sustain activity
eligibility\_criteria: LackOfEnergy, NeedForEnergy, [...]
didactics: An overwhelming sustained sense of exhaustion
           and decreased capacity for physical and mental work at
           usual level
step: MedicationWatch, DailyLivingWatch, ...
first step: MedicationWatch
```

The Action Steps contain general information about the actions, like which step follows the current action. Our definition implies that the step *DailyLivingWatch* is to be executed after the execution of *MedicationWatch*.

Listing 1.4. Class: Action Step

```
action: AS\_MedicationWatch
subguideline: null
next\_step: DailyLivingWatch
```

```
name: DailyLivingWatch
action: AS\_DailyLiving
subguideline: null
next\_step: [...]
```

Besides that, the *Action Specification* is defined, where further informations about the step are given.

Listing 1.5. Class: Action Specification

```
name: AS\_MedicationWatch
patient\_data: Patient\_Medication\_Overview
description: Evaluate the patient s routine
             prescription and over-the-counter medications
didactics: [list of supporting didactic materials]
```

Listing 1.6. Class: Action Specification

```
name: AS\_DailyLiving
patient\_data: Patient\_DailyLiving\_Overview
description: Assess the patient s ability to perform
             activities of daily living
didactics: [list of supporting didactic materials]
```

GLIF is a comprehensive approach, which allows medical guidelines to be modeled from various points of views. But for implementing NANDA in this formalization, we found that some important features are missing (again we reuse the building blocks defined in Fig. 1):

- ① match the *name* property of the *Guideline* class
- ② is equivalent to the *didactics* property
- ③ can be expressed as a part *eligibility_criteria* property while ④ is represented by the *intentions* property of the same class
- ④ and ⑤ have no match at all
- ⑥ can be implemented either as instance of the *Supplemental Material* class or in the corresponding didactics
- ⑦ and ⑨ are potential instances of the *action step* class
- ⑩ and ⑪ can be added to the *didactics* property of the *Action Specification* class

3.4 Summarizing the Evaluation Results

In this section three representative standards for CIGs have been evaluated based on their capability to implement the NNN building blocks ① to ⑪ as summarized in Figure 1. Apparently, none of these approaches can implement the entire structure of the NNN knowledge. Particularly, an implementation for risks and related factors of a given diagnosis are missing. Hence, in order to meet our requirement to be able to formalize the NNN knowledge in a complete manner, we introduce the NNN formalization for application in the care domain.

Table 1. Evaluation of CIG Standards along NNN Building Blocks

	①	②	③	④	⑤	⑥
Arden Syntax	+	+	+	-	-	+
Asbru	+	-	0	-	-	+
GLIF	+	+	+	-	-	+

	⑦	⑧	⑨	⑩	⑪
Arden Syntax	+	0	+	-	-
Asbru	+	+	+	-	+
GLIF	+	+	+	0	0

+: supported, -: not supported, 0: workaround

4 NNN Formalization

The following section describes a way of formalizing NNN. After a short introduction into the structure, the different parts are described in detail.

4.1 Overview

The NNN formalization (based on XML) is divided into the following three distinct sections.

1. **Meta:** Meta-Information about the guideline
2. **Custom:** institution-specific preferences for certain tasks
3. **Guideline:** information about the guideline itself

Before describing the structure itself, we want to describe some elements that may reoccur throughout the sections:

- **<hints>** can be added to various elements. Each of these elements may consist of one or more **<hint>** elements, defining its origin (*from*) and its purpose (*text*) in its attributes. Listing 1.7 defines the RNG schema for hints.
- **<examples>** can be added to support nurses when deciding which treatments to apply. Listing 1.8 defines the RNG schema for examples.
- **<inputs>** are used to enforce comprehensive documentation. A task may require more than one parameters of a specific type (e.g. natural numbers, scales, ...) for its comprehensive documentation, e.g. saving the current blood pressure (both systolic and diastolic) of a patient. Choosing the right type further enables to reuse the information defined in the NOC when scoring them. To enable this, **<outcome>** elements and **<task>** elements support **<inputs>**. To support also complex data structures (e.g. systolic and diastolic combined as blood pressure), **<inputs>** elements may nest multiple **<input>** elements, where each of them contains the attributes *label* describing the meaning of it. Listing 1.9 defines the RNG schema for inputs.
- Each element containing care relevant information defines a **score attribute** containing a natural number between 1 and 10. We intend to use this attribute to express preferences to decision support systems (DSS).

Listing 1.7. RNG schema for the hints element

```
<define name="refhints">
  <optional>
    <element name="hints">
      <oneOrMore>
        <element name="hint">
          <optional>
            <attribute name="from">
              <text/>
            </attribute>
          </optional>
          <attribute name="text">
            <text/>
          </attribute>
          <optional>
            <attribute name="score">
              <data type="integer"/>
            </attribute>
          </optional>
        </element>
      </oneOrMore>
    </element>
  </optional>
</define>
```

Listing 1.8. RNG schema for the examples element

```
<define name="refexamples">
```

```

<optional>
  <element name="examples">
    <oneOrMore>
      <element name="example">
        <attribute name="text">
          <text/>
        </attribute>
        <optional>
          <attribute name="score">
            <data type="integer"/>
          </attribute>
        </optional>
        <optional>
          <ref name="nestedexamples"/>
        </optional>
      </element>
    </oneOrMore>
  </element>
</optional>
</define>
<define name="nestedexamples">
  <optional>
    <element name="examples">
      <oneOrMore>
        <element name="example">
          <attribute name="text">
            <text/>
          </attribute>
          <optional>
            <attribute name="score">
              <data type="integer"/>
            </attribute>
          </optional>
        </element>
      </oneOrMore>
    </element>
  </optional>
</define>

```

Listing 1.9. RNG schema for the inputs element

```

<define name="refinputs">
  <optional>
    <element name="inputs">
      <oneOrMore>
        <element name="input" ns="http://relaxng.org/ns/structure/1.0">
          <attribute name="label">
            <text/>
          </attribute>
          <ref name="any"/>
        </element>
      </oneOrMore>
    </element>
  </optional>
</define>

```

Additionally to the RNG schemes, examples based on the case study of nursing diagnosis FATIGUE demonstrate several parts of our formalization. The formalized nursing knowledge can be found in NANDA diagnosis repositories like [20] and [7]. This example only serves the purpose of demonstrating the comprehensiveness of our formalization - for real life application, nursing professionals have to implement a practical formalization including relevant scales for a full documentation. This care-domain specific knowledge is out of the scope of this technical report.

4.2 Meta

This section contains general information about the guideline, including the bibliography. It therefore includes the building blocks ①, ② and ⑥ (see Fig. 1). Information about the *state* (i.e. research, implementing, testing, running or expired define by [4]), the *author*, the *validator*, the *implementer*, and various *dates* of the guideline are also part of this section.

Listing 1.10 shows the RNG schema for the meta section.

Listing 1.10. RNG schema for the Meta section

```
<element name="meta">
  <element name="title">
    <attribute name="text">
      <text/>
    </attribute>
  </element>
  <element name="definition">
    <attribute name="text">
      <text/>
    </attribute>
    <attribute name="theme">
      <text/>
    </attribute>
    <ref name="refhints"/>
  </element>
  <element name="version">
    <attribute name="id">
      <text/>
    </attribute>
  </element>
  <element name="validation">
    <attribute name="status">
      <choice>
        <value>research</value>
        <value>implementing</value>
        <value>testing</value>
        <value>running</value>
        <value>expired</value>
      </choice>
    </attribute>
  </element>
  <optional>
    <element name="institution">
      <attribute name="name">
        <text/>
      </attribute>
    </element>
  </optional>
  <optional>
    <element name="author">
      <attribute name="name">
        <text/>
      </attribute>
    </element>
  </optional>
  <optional>
    <element name="validator">
      <attribute name="name">
        <text/>
      </attribute>
    </element>
  </optional>
  <optional>
    <element name="implementer">
```

```

        <attribute name="name">
          <text/>
        </attribute>
      </element>
    </optional>
    <element name="date">
      <attribute name="text">
        <data type="date"/>
      </attribute>
    </element>
    <element name="bibliography">
      <oneOrMore>
        <element name="item">
          <attribute name="id">
            <text/>
          </attribute>
          <attribute name="text">
            <text/>
          </attribute>
        </element>
      </oneOrMore>
    </element>
  </element>

```

Listing 1.11 implements the meta section for FATIGUE exemplarily. This also includes a minimalistic version of the bibliography, which contains two elements in this example.

Listing 1.11. The META section for FATIGUE

```

<meta>
  <title text="fatigue"/>
  <definition text="An overwhelming, sustained sense of exhaustion and decreased
    capacity for physical and mental work at usual level"/>
  <version id="1.0"/>
  <validation status="implementing"/>
  <institution name="University of Vienna"/>
  <author name="Georg Kaes"/>
  <validator name="Stefanie Rinderle-Ma"/>
  <implementer name="Juergen Mangler"/>
  <date text="2014-04-01"/>
  <bibliography>
    <item id="1" text="M. Doenges, M. F. Moorhouse, and A. Geissler-Murr,
      Pflegediagnosen und Massnahmen, 3rd ed. Verlag Hans Huber, 2002."/>
    <item id="2" text="Marlies Ehmman and Ingrid Volkel, Pflegediagnosen in
      der Altenpflege, 2nd ed. URBAN & FISCHER, 2004"/>
  </bibliography>
</meta>

```

4.3 Custom

The *custom* section contains institution specific preferences regarding <tasks> elements (see the next section for more information about tasks).

- <**recommended**> is used to express not binding priorities for various treatments of a specific diagnosis. The value of its element (ranging from 1 to 10) influences the order in which arbitrary treatments to a specific diagnoses are listed.
- <**mandatory**> is used to enable care manager to enforce a specific treatment for a specific diagnoses.

Listing 1.12. RNG scheme of the Custom section

```
<element name="custom">
  <zeroOrMore>
    <element name="recommended">
      <attribute name="id">
        <text/>
      </attribute>
      <attribute name="score">
        <data type="integer">
          <param name="minInclusive">1</param>
          <param name="maxInclusive">10</param>
        </data>
      </attribute>
    </element>
  </zeroOrMore>
  <zeroOrMore>
    <element name="mandatory">
      <attribute name="id">
        <text/>
      </attribute>
    </element>
  </zeroOrMore>
</element>
```

4.4 Guideline

The *Guideline* section represents the actual care-specific knowledge and consists of the following sections:

- **<factors>** contain ④ and ⑤ for each diagnose where they apply. Semantically speaking, it expresses reasons that may cause symptoms or define the specific risks. Again, multiple **<factor>** elements (with at least the attributes *text* (semantic description), *type* (either risk (④) or related (⑤))) are nested below one **<factors>** element. The optional attribute *category* is used for further refinement e.g. psychological, physiological or environmental. Further do **<factor>** elements support additional elements including **<hints>** and/or **<examples>** and **<causes>** elements (described above).
- **<symptoms>** represent the building block ③. They contain multiple **<symptom>** elements which itself can nest elements of the type **<causes>**, **<hints>**, and **<examples>**.
- **<outcomes>** defining the evaluation criteria for treatments, which is to be done everytime a treatment has ended. This element nests multiple elements of the type **<outcome>**. **<outcome>** elements themselves define three mandatory attributes, namely *text* (description), *source* (⑥), and an *id* (unique for the scope of the guideline). Similar to the two above, an **<outcome>** element nests multiple elements of the type **<hints>**, **<examples>**, and **<inputs>** (which connects it to ⑩ for evaluation purposes).
- **<tasks>** represent sequences of activities and are therefore used to specify ⑦, ⑧ and ⑩. During the specification of a therapy for a patient, it can be defined to execute the distinct **<task>** elements, which contain the actual activities, either sequentially (nest them into a **<sequential-task>** element) or in parallel (nest them into a **<parallel-task>** element). It should be noted that at this point arbitrary levels of nesting are supported, allowing to define complex tasks too. We explicitly

avoided supporting *cycles* within guidelines as this is expressed at the (higher) level of the care plan (which is out of the scope of this technical report). Further, both `<sequential-tasks>` and `<parallel-tasks>` elements support the optional attributes *name* and *text*. Each `<task>` element contains the attributes *id* and *text*.

Additionally, each of these tags can contain a *source* attribute, thus documenting the scientific source of the formalized knowledge. The source references the ID of the respective element in the bibliography section of the diagnosis.

Listing 1.22 shows an excerpt of the `<tasks>` section of FATIGUE. The implemented guideline steps have been taken from various sources, including [8] and the german literature [7] and [5].

Listing 1.13. RNG schema for the definition of tasks

```
<element name="tasks">
  <element name="labels">
    <oneOrMore>
      <element name="label">
        <attribute name="text">
          <text/>
        </attribute>
      </element>
    </oneOrMore>
  </element>
  <oneOrMore>
    <ref name="reftask"/>
  </oneOrMore>
</element>
```

Listing 1.14 shows the RNG schema for the Emphasis of Nursing Documentation. The listing references 1.15, which implements the specific documentary items.

Listing 1.14. RNG schema for the definition of Emphasis of Nursing Documentation

```
<element name="documentation">
  <oneOrMore>
    <ref name="refdocu"/>
  </oneOrMore>
</element>
```

Listing 1.15. RNG definition for refdocu

```
<define name="refdocu">
  <element name="item">
    <attribute name="id">
      <text/>
    </attribute>
    <attribute name="phase">
      <text/>
    </attribute>
    <attribute name="text">
      <text/>
    </attribute>
    <ref name="refhints"/>
    <ref name="refexamples"/>
    <ref name="refinputs"/>
  </element>
</define>
```

Listing 1.16 concludes the documentary elements by giving an example of the nursing diagnosis FATIGUE.

Listing 1.16. Exemplary elements from the Emphasis of nursing documentation of the diagnosis FATIGUE

```
<documentation>
  <item text="Aeusserungsformen der Erschoepfung und andere Ergebnisse der
    Einschaetzung" id="30" phase="Pflegeassessment oder Neueinschaetzung">
  <item text="Grad der Beeintraechtigung, Auswirkungen auf die gewohnte
    Lebensweise" id="31" phase="Pflegeassessment oder Neueinschaetzung">
  <item text="Erwartungen des Patienten / seiner Angehoerigen bezueglich
    Faehigkeiten / individueller Situation" id="32" phase="Pflegeassessment
    oder Neueinschaetzung">
  <item text="Pflegeplan/-interventionen und beteiligte Personen" id="33" phase="
    Planung">
  <item text="Plan fuer die Patientenanleitung, -schulung und -beratung" id="34"
    phase="Planung">
  <item text="Reaktionen auf Interventionen / Anleitung und ausgefuehrte
    Pflgetaetigkeiten" id="35" phase="Durchfuehrung/Evaluation">
  <item text="Veraenderungen des Plans" id="36" phase="Durchfuehrung/Evaluation">
  <item text="Zielerreichung / Fortschritte in Richtung Zielerreichung" id="37"
    phase="Durchfuehrung/Evaluation">
  <item text="Beduerfnisse hinsichtlich Spitalaustritt / Entlassungs- oder
    Austrittsplanung: Was muss getan werden, wer ist verantwortlich?" id="38"
    phase="Entlassungs- oder Austrittsplanung">
  <item text="Spezifisches Vermitteln an andere Gesundheitsberufe" id="39" phase="
    Entlassungs- oder Austrittsplanung">
</documentation>
```

For the definition of the control flow of tasks during the execution of therapies, tasks can be embedded into arbitrary numbers of control flow structures supporting parallel and sequential execution, which enable the creation of complex workflows and dependencies. Additionally to the tasks, the elements from the Emphasis of Nursing Documentation section can be added to a patients therapy plan, thus emphasizing a comprehensive documentation of the patients state. This section also includes the *Emphases of Nursing Documentation*, as described in [7].

Listing 1.17 shows the RNG schema for the plans, which can be generated using our NNN formalization. It implements three definitions, which are illustrated in listings 1.18, 1.19, 1.20 and 1.21.

Listing 1.17. RNG schema for the workflow definition of tasks

```
<element name="tasks">
  <choice>
    <ref name="refsequential"/>
    <ref name="refparallel"/>
  </choice>
</element>
</element>
```

Tasks or other group of tasks within the *sequential-tasks* are carried out one after another.

Listing 1.18. RNG definition for refsequential

```
<define name="refsequential">
  <element name="sequential-tasks">
    <ref name="tasksbody"/>
  </element>
</define>
```

In contrast to this, tasks or group of tasks within the *parallel-tasks* section can be executed in parallel.

Listing 1.19. RNG definition for refparallel

```
<define name="refparallel">
  <element name="parallel-tasks">
    <ref name="tasksbody"/>
  </element>
</define>
```

Listing 1.20 defines, what exactly can be included in a `<parallel-tasks>` or a `<sequential-tasks>` element. It can either be another group, or an atomic tasks. Additionally, some attributes can be used to describe the current group of tasks in detail.

Listing 1.20. RNG definition for tasksbody

```
<define name="tasksbody">
  <optional>
    <attribute name="name">
      <text/>
    </attribute>
  </optional>
  <optional>
    <attribute name="text">
      <text/>
    </attribute>
  </optional>

  <!-- include subordinate plans or atomic tasks -->
  <oneOrMore>
    <choice>
      <ref name="refsequential"/>
      <ref name="refparallel"/>
      <ref name="reftask"/>
    </choice>
  </oneOrMore>
</define>
```

Listing 1.21 defines the exact structure of an atomic task item.

Listing 1.21. RNG definition for reftask

```
<define name="reftask">
  <element name="task">
    <attribute name="text">
      <text/>
    </attribute>
    <attribute name="id">
      <text/>
    </attribute>
    <optional>
      <attribute name="predictedeffort">
        <data type="integer"/>
      </attribute>
    </optional>
    <optional>
      <attribute name="score">
        <data type="integer"/>
      </attribute>
    </optional>
    <ref name="refhints"/>
    <ref name="refexamples"/>
    <ref name="refinputs"/>
  </element>
</define>
```

Listing 1.22 gives an exemplary overview of the implementation of the `<tasks>` tag in the NNN formalization.

Listing 1.22. Tasks of the diagnosis FATIGUE in the NNN formalization

```
<tasks>
  <labels>
    <label name="Energy Management"/>
  </labels>
  <task text="Evaluate medication" id="0">
    <!-- examples can support the understanding of the related component -->
    <examples>
      <example text="Fatigue can be a byeffect of beta blockers and chemo therapy
      ."/>
    </examples>
    <!-- Inputs can be used to define the documentation of a task -->
    <inputs>
      <input label="Short summary" xmlns="http://relaxng.org/ns/structure/1.0">
        <element name="summary">
          <data type="string">
            <param name="maxLength">50</param>
          </data>
        </element>
      </input>
      <input label="Detailed Medication" xmlns="http://relaxng.org/ns/structure
      /1.0">
        <element name="eingabe2">
          <text/>
        </element>
      </input>
    </inputs>
  </task>
  <task text="assess physical or psychical medical conditions" id="1">
    <examples>
      <example text="MS"/>
      <example text="Lupus"/>
      <example text="chronical pain"/>
      <example text="Hepatitis"/>
      <example text="AIDS"/>
      <example text="fear"/>
    </examples>
  </task>
  <task text="evaluate adequacy of nutrition and sleep" id="2">
    <!-- Hints can be used to give other additional information than examples -->
    <hints>
      <hint>Sometimes clients with chronic fatigue symptom can sleep excessively
      and need support to limit sleeping.</hint>
    </hints>
  </task>
  <task text="assess how the fatigue develops during the day" id="3">
    <inputs>
      <input label="Detailed Analysis" xmlns="http://relaxng.org/ns/structure
      /1.0">
        <element name="eingabe2">
          <text/>
        </element>
      </input>
    </inputs>
  </task>
  [...]
</tasks>
```

Listing 1.23 defines the body which is used in the factors and outcomes sections.

Listing 1.23. RNG schema for the recurring standard body

```
<define name="stdbody">
  <optional>
    <source>
      <oneOrMore>
        <element name="item">
          <attribute name="id">
```

```

        <text/>
      </attribute>
    </element>
  </oneOrMore>
</source>
</optional>
<optional>
  <attribute name="category">
    <text/>
  </attribute>
  <optional>
    <attribute name="subcategory">
      <text/>
    </attribute>
  </optional>
</optional>
<attribute name="text">
  <text/>
</attribute>
<ref name="refhints"/>
<ref name="refexamples"/>
</define>

```

As stated before, multiple elements necessary for implementing a comprehensive representation of the NNN guidelines cannot be represented using CPG approaches. Especially factors related to the diagnosis have no matching elements in the evaluated CPGs. Listing 1.25 shows how we model these building blocks.

Listing 1.24. RNG schema for the factors section

```

<element name="factors">
  <oneOrMore>
    <element name="factor">
      <attribute name="category">
        <choice>
          <value>risk</value>
          <value>related</value>
        </choice>
      </attribute>
      <ref name="stdbody"/>
    </element>
  </oneOrMore>
</element>

```

Listing 1.25. Factors of the diagnosis FATIGUE in the NNN formalization

```

<factors>
  <factor category="psychological" text="Boring lifestyle"/>
  <factor category="psychological" text="Stress"/>
  <factor category="psychological" text="Anxiety"/>
  <factor category="psychological" text="Depression"/>
  <factor category="environmental" text="Humidity"/>
  <factor category="environmental" text="Humidity"/>
  <factor category="environmental" text="Lights"/>
  <factor category="environmental" text="Noise"/>
  <factor category="environmental" text="Temperature"/>
  <factor category="physiologisch" text="changed chemical processes in the patient
's body">
    <factors>
      <factor text="medicines"/>
      <factor text="drug withdrawal"/>
      <factor text="chemotherapy"/>
    </factors>
  </factor>
</factors>
<hints>
  <hint text="e.g. because of medicines, drug withdrawal or other reasons"/>
</hints>

```

```

    </factor>
    [...]
</factors>

```

The defining characteristics are possible symptoms a patient can show when he suffers the given diagnosis. Listing 1.27 shows how symptoms can be modeled in our formalization.

Listing 1.26. RNG schema for the symptoms section

```

<element name="symptoms">
  <oneOrMore>
    <element name="symptom">
      <ref name="stdbody"/>
    </element>
  </oneOrMore>
</element>

```

Listing 1.27. Symptoms of the diagnosis FATIGUE in the NNN formalization

```

<symptoms>
  <symptom category="subjective" text="inability to restore energy even after
    sleep"/>
  <symptom category="subjective" text="lack of energy or inability to maintain
    usual level of physical activity"/>
  <symptom category="subjective" text="increase in the rest requirements"/>
  <symptom category="subjective" text="tired"/>
  <symptom category="subjective" text="lethargic"/>
  <symptom category="subjective" text="listless"/>
  <symptom category="subjective" text="perceived need for additional energy to
    accomplish routine tasks"/>
  <symptom category="subjective" text="introspection"/>
  <symptom category="subjective" text="compromised libido"/>
  <symptom category="subjective" text="feeling of guilt for not keeping up with
    responsibilities"/>
  <symptom category="subjective/objective" text="inability to maintain usual
    routines"/>
  <symptom category="subjective/objective" text="compromised concentration"/>
  <symptom category="subjective/objective" text="disinterest in surroundings"/>
  <symptom category="subjective/objective" text="drowsy"/>
  <symptom category="objective" text="decreased performance"/>
  <symptom category="objective" text="increase in physical complaints"/>
  <symptom category="objective" text="verbalization of an unremitting and
    overwhelming lack of energy"/>
</symptoms>

```

Outcomes, as defined in the Nursing Outcomes Classification, are seamlessly integrated into our formalization. Listing 1.29 shows three exemplary outcomes and the respective NOC labels. While the first two outcomes implement documentary <input> items for assessing whether a patient has reached the goal or not, the last outcome includes the possibility of documenting the way a patient describes his plan of conserving energy.

Listing 1.28. RNG schema for the outcome section

```

<element name="outcomes">
  <element name="labels">
    <oneOrMore>
      <element name="label">
        <attribute name="text">
          <text/>
        </attribute>
      </element>
    </oneOrMore>
  </element>
</element>

```

```

</element>
<oneOrMore>
  <element name="outcome">
    <attribute name="id">
      <text/>
    </attribute>
    <attribute name="goal">
      <choice>
        <value>achieve</value>
        <value>maintain</value>
        <value>prevent</value>
      </choice>
    </attribute>
    <attribute name="text">
      <text/>
    </attribute>
    <ref name="refhints"/>
    <ref name="refexamples"/>
    <ref name="refinputs"/>
  </element>
</oneOrMore>
</element>

```

Listing 1.29. Exemplary outcomes for FATIGUE

```

<outcomes>
  <labels>
    <label text="Endurance"/>
    <label text="Concentration"/>
    <label text="Energy Conservation"/>
    <label text="Nutrition Status: Energy"/>
  </labels>
  <outcome id="1" goal="achieve" text="The patient verbalizes increased energy.">
    <inputs>
      <input label="Goal reached" xmlns="http://relaxng.org/ns/structure/1.0">
        <element name="select">
          <element name="yes">Yes</element>
          <element name="no">No</element>
        </element>
      </input>
    </inputs>
  </outcome>
  <outcome id="2" goal="achieve" text="The patient verbalizes improved well-being
  .">
    <inputs>
      <input label="Goal reached" xmlns="http://relaxng.org/ns/structure/1.0">
        <element name="select">
          <element name="yes">Yes</element>
          <element name="no">No</element>
        </element>
      </input>
    </inputs>
  </outcome>
  <outcome id="3" goal="achieve" text="The patient explains energy conservation
  plan to offset fatigue."/>
    <inputs>
      <input label="Patients explanation" xmlns="http://relaxng.org/ns/structure
      /1.0">
        <element name="input-pat">
          <text/>
        </element>
      </input>
    </inputs>
  </outcome>
  [...]
</outcomes>

```

5 Discussion

The NNN formalization presented in this paper constitutes an initial step into the field of 'computer-aided' nursing in the care domain as envisioned by the ACaPlan project by providing the basic building blocks required for comprehensive and formalized care planning and execution.

During development of the NNN formalization, the discrepancies between the medical and the care domain as described in chapter 1 based on the differences between respective guidelines became obvious very quickly, although these fields of research are related on many levels. The fact, that the revised CIGs do not support any means for implementing risks and related factors resulted in the need for developing an approach to formalize knowledge specific for NNN guidelines. The implementation of NNN guidelines in this technical report covers all relevant parts of a diagnosis, so a patient's state can be assessed from various points of view: On the one hand, based on data about his history, risks and related factors can be used to diagnose potential threats very early; on the other hand it is possible to use symptoms a patient shows to determine his diagnoses. These different perspectives support nursing personnel when creating therapy plans in many situations. Additionally, by introducing <input> tags, different scales and documentary forms can be added to support the documentation.

Evaluations with domain experts show that the formalization presented in this technical report addresses all relevant parts of NANDA, NIC and NOC. Thus, guidelines formalized using this approach can be applied for supporting nurses in real life scenarios.

Acknowledgment: We thank Adelheid Beyerl and Martin Zigler from the care centers St. Pölten Pottenbrunn and Vitacon for the discussions and insights into the care domain as well as the great cooperation within the ACaPlan project.

References

1. NANDA International, "NANDA-I, NIC, NOC for safe patient care," 2012. [Online]. Available: <http://www.nanda.org/NNN.aspx>
2. A. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q. Zeng, D. Wang, V. Patel, R. Greenes, E. Shortliffe *et al.*, "Glif3: a representation format for sharable computer-interpretable clinical practice guidelines," *Journal of biomedical informatics*, vol. 37, no. 3, pp. 147–161, 2004.
3. R. Kosara, S. Miksch, Y. Shahar, P. Johnson *et al.*, "Asbruvview: capturing complex, time-oriented plans-beyond flow-charts," *Thinking with diagrams*, vol. 98, pp. 22–23, 1998.
4. T. Pryor and G. Hripcsak, "The arden syntax for medical logic modules," *International journal of clinical monitoring and computing*, vol. 10, pp. 215–224, 1993.
5. Marlies Ehmann and Ingrid Völkel, *Pflegediagnosen in der Altenpflege*, 2nd ed. URBAN & FISCHER, 2004.
6. NANDA International, "Nursing diagnosis frequently asked questions," 2012. [Online]. Available: <http://www.nanda.org/NursingDiagnosisFAQ.aspx>
7. M. Doenges, M. F. Moorhouse, and A. Geissler-Murr, *Pflegediagnosen und Massnahmen*, 3rd ed. Verlag Hans Huber, 2002.
8. NANDA International, "Nursing diagnoses - fatigue — nursing diagnosis for nurses and bsn students," <http://nandanursingdiagnosis.org/nursing-diagnosis-fatigue/>, 2013. [Online]. Available: <http://nandanursingdiagnosis.org/nursing-diagnosis-fatigue/>

9. "Fatigue - mynurse," <http://mynurse.weebly.com/fatigue.html>, 2013. [Online]. Available: <http://mynurse.weebly.com/fatigue.html>
10. "Nanda glossary of terms," <http://www.nanda.org/nanda-international-glossary-of-terms.html>, 2012. [Online]. Available: <http://www.nanda.org/nanda-international-glossary-of-terms.html>
11. N. Mulyar, W. van der Aalst, and M. Peleg, "A pattern-based analysis of clinical computer-interpretable guideline modeling languages," *Journal of the American Medical Informatics Association*, vol. 14, no. 6, pp. 781 – 787, 2007.
12. P. De Clercq, K. Kaiser, and A. Hasman, "Computer-interpretable guideline formalisms," *Studies in health technology and informatics*, vol. 139, p. 22, 2008.
13. M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, R. Hall, P. D. Johnson, N. Jones, A. Kumar *et al.*, "Comparing computer-interpretable guideline models: a case-study approach," *Journal of the American Medical Informatics Association*, vol. 10, no. 1, pp. 52–68, 2003.
14. Evicare, "Evidence-based care processes: Integrating knowledge in clinical information systems," 2009.
15. Y. Shahar, O. Young, E. Shalom, A. Mayaffit, R. Moskovitch, A. Hessing, and M. Galperin, "Degel: A hybrid, multiple-ontology framework for specification and retrieval of clinical guidelines," in *Artificial Intelligence in Medicine*, ser. Lecture Notes in Computer Science, M. Dojat, E. Keravnou, and P. Barahona, Eds. Springer Berlin Heidelberg, 2003, vol. 2780, pp. 122–131. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-39907-0_18
16. G. Hripcsak, P. D. Clayton, T. A. Pryor, P. Haug, O. B. Wigertz, and J. van der lei, "The arden syntax for medical logic modules," *Symposium on Computer Applications in medical care*, pp. 200–204, 1990.
17. P. A. de Clercq, K. Kaiser, and A. Hasman, "Computer-interpretable guideline formalisms," *Studies in Health Technology and Informatics*, vol. 139, pp. 22–43, 2008.
18. Y. Shahar, S. Miksch, and P. Johnson, "The asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines," *Artificial Intelligence in Medicine*, vol. 14, no. 1/962, pp. 29 – 51, 1998.
19. M. Peleg, A. A. Boxwala, S. Tu, D. Wang, O. Ogunyemi, and Q. Zeng, "Guideline interchange format 3.5 technical specification," 2004.
20. A. K. Aplikasi, "Nanda: Nursing diagnosis: Fatigue application of nanda, noc, nic," 2012. [Online]. Available: <http://asuhankeperawatanonline.blogspot.co.at/2012/03/nursing-diagnosis-fatigue-application.html>