Use of a Knowledge Patterns-Based Tool for Dealing With the "Narrative Meaning" of Complex Iconographic Cultural Heritage Items

Gian Piero Zarri ✉

STIH Laboratory, Sorbonne University, Paris 75005, France
zarri@noos.fr

Abstract. This paper suggests to use a powerful tool like NKRL, the "Narrative Knowledge Representation Language", to deal with the representation and management in digital form of those important Cultural Heritage entities corresponding to the "Iconographic Narratives". These denote the "stories" conveyed by paintings, drawings, frescoes, mosaics, sculptures, murals and similar but also by pictures, posters, comics, cartoons, movies etc. An example of use of NKRL to deal with the complex narrative situation represented by the central scene of Diego Velazquez's "The Surrender of Breda" is included in the paper.

Keywords: Iconographic narratives • NKRL • Knowledge patterns • Ontologies • Inference techniques.

CEUR-WS.org/Vol-2320/paper9.pdf

1 Introduction

The notion of "*pattern*" is very popular in the software engineering domain to denote general, reusable solutions to commonly occurring problems; examples are the architectural patterns, the design patterns, configuration patterns, memory management patterns, synchronization patterns and so on. A new class of software patterns has been proposed in the early 2000s, the "*knowledge patterns*" – this term appears in a paper presented by Clark and colleagues at KR 2000 [1]. A knowledge pattern denotes a first-order structure, independently created, that provides a *reusable solution* to a given knowledge representation (a modelling) problem. To use it, a "*morphism*" (transformation) must be fashioned for each intended application of this pattern into a target knowledge base; the morphism denotes a *consistent mapping* of the general terms of the pattern to specific terms in the base describing, then, how the pattern should be transformed.

These "knowledge patterns" show some evident similarities with the "*ontology design patterns*", ODPs, developed roughly in the same period in a Semantic Web (SW) context. The standard ODPs definition [2: 140] states, in fact, that "… an ontology design pattern is a set of ontological elements, structures or construction principles that intend to solve a specific engineering problem and that recurs, either exactly replicated or in an adapted form, within some set of ontologies …". As the knowledge patterns introduced above, the ODPs denote then reusable successful solutions to recurrent modeling problems that could be composed, specialized and reutilized. However, their modalities of creation are totally different. While the knowledge patterns are developed bottom-up according to rigorous shaping principles, ODPs are created top-down and simply denote small fragments of already existing ontologies in the DOLCE [3] and SW style. ODPs have then been *severely criticized* – interestingly enough, within the SW community itself – as being characterized by a high level of heterogeneity and by the lack of shared theoretical principles for their construction and use. We can see in this context the patterns collected in the ODP portal [4] that evoke clearly those idiosyncratic patterns that "lack compatibility with others" and lead "to decrease the semantic interoperability of ontologies", as mentioned in a well-known paper by Kozaki et al [5: 39]. Similarly, in their analysis of the (largely unsuccessful) attempts of the SW scholars to upgrade their standard binary structures to *n*-ary ones, Trame et al. [6: 211] note about patterns that "… [they are] frequently used in an arbitrary fashion, lacking any design rationale". Blomqvist and colleagues [7] list a series of important criticisms about the current ODPs situation. They note that, in an ODPs context, there is *a serious lack of standardization directives*, of (automatic) tools to develop and publish patterns and of tools for their evaluation. Additionally, there are evident conflicts about the proposed patterns and conceptual problems for extracting them from the original ontologies, a lack of tools to represent additional information to be added to a pattern after its extraction, an impossibility to understand how different patterns can relate to each other, a lack of maintenance and documentation means; etc.

A knowledge representation language and a computer science tool that conforms well to the knowledge pattern requirements is NKRL, the "Narrative Knowledge Representation Language" [8]. NKRL has been expressly created, thanks to several European projects, to formalize as accurately as possible and to manage then in the most efficient way those real world, dynamically characterized and particularly ubiquitous entities denoted as "*narratives*", see [9, 10]. In an NKRL context, a narrative is seen as a *coherent stream* (i.e., its components are logically and chronologically connected) of spatio-temporally constrained *elementary events* describing the activities, states, experiences, behaviors etc. of the characters involved in the narrative. The most important innovation of this language with respect to the usual ontological paradigms concerns the addition of an "ontology of elementary events" to the usual "ontology of concepts"; the two ontologies are formally and functionally different, but strictly integrated from an operation point of view. The nodes of the ontology of events are represented by well-formed knowledge patterns – called "*templates*" in an NKRL context – that denote general classes of elementary events like "be present in a place", "move a physical object", "have a specific attitude towards someone/something", "send/receive messages", etc. A precisely defined kind of *morphism* allow us to pass from the formal representation of a template to that of the specific elementary events belonging to the class defined by the template, like "Peter is now living in Paris".

NKRL has been successfully used in many different "narrative" domains see, e.g., [11, 12, 13, 14]. In this paper, we propose to test the possible utility of this language in a Cultural Heritage context, more precisely to make use of the NKRL's knowledge patterns approach to represent the "*stories*" denoted by those "*iconographic items*" – paintings, drawings, frescoes, mosaics, sculptures, murals etc. but also pictures, posters, advertising artworks, comics, cartoons, movies… – that represent a fundamental component of the Cultural Heritage domain. In the following, Section 2 will supply a short description of the main features of NKRL. Section 3 illustrates a concrete example showing how NKRL can be used to accurately represent and manage a complex iconographic situation; Section 4 will supply, eventually, a short "Conclusion".

2 A Short Revue of the Main Features of NKRL

2.1 The Two Ontologies

The NKRL ontology of concepts is called *HClass (hierarchy of classes)* and includes presently (January 2019) more than 7,500 standard concepts – "standard" meaning here that the properties or attributes used to define a given concept are simply expressed according to the usual binary relationships of the property/value type. From a formal point of view HClass – see [8: 43-55, 123-137] – is not fundamentally different, then, from the ontologies that can be built up by using the frame version of Protégé [15].

The ontology of elementary events is, by contrast, a new sort of hierarchical organization where, as already stated, the nodes correspond to *well-formed n-ary knowledge patterns* called "templates": their basic core is depicted by Eq. 1. This ontology is denoted in NKRL as HTemp (hierarchy of templates). In opposition to the "*static*" – i.e., that can be defined *a priori* – notions denoted by HClass concepts like "human being", "color", "artefact", "control room", "valve", "level of temperature" …, templates concern the representation in machine-understandable format of real world dynamically characterized entities as complex events, situations, scripts or narratives. More precisely, they must be conceived as the canonical, formal representation of general classes of elementary events like "be present in a place" etc., see the previous Section.

$$(\mathsf{L}_i\ (\mathsf{P}_j\ (\mathsf{R}_1\ \mathsf{a}_1)\ (\mathsf{R}_2\ \mathsf{a}_2)\ \ldots\ (\mathsf{R}_n\ \mathsf{a}_n)))\ . \qquad (1)$$

In Eq. 1, L_i is the symbolic label identifying (reifying) the *n*-ary structure corresponding to a specific template/knowledge pattern. P_j is a conceptual predicate. R_k is a generic functional role [16] used to specify the logico-semantic function of its filler a_k with respect to the predicate. a_k is then a predicate argument introduced by the role R_k.

2.2 Templates and Templates' Instantiation

Let us assume that a template following the general syntax of Eq. 1 and denoted by Move:TransferMaterialThingsToSomeone in NKRL (see the upper part of Table 1) must be instantiated (through a suitable morphism) to provide the formal representation of an elementary event like "Bill gives an art book to Mary as a present". The main result of the application of the morphism is that the predicate P_j (MOVE) will now introduce its four arguments a_k – JOHN_, MARY_, ART_BOOK_1 ("individuals", i.e., instances of HClass concepts) and as_a_gift (a specific HClass concept) – via, respectively, the four relationships (R_k roles) SUBJ(ect), BEN(e)F(iciary), OBJ(ect) and MODAL(ity). In the "external" format of the NKRL metalanguage, "individuals" are represented in uppercase and "concepts" in lowercase. The resulting *n*-ary structure (see the lower part of Table 1) will be then reified making use of a symbolic label L_i and managed as a coherent block. The templates' instances derived through the morphism are called *predicative occurrences* and supply the formal images of specific elementary events.

Table 1. Deriving a predicative occurrence from a template.

name: Move:TransferMaterialThingsToSomeone
father: Move:TransferToSomeone
position: 4.21
natural language description: "Transfer a Material Thing (e.g., a product, an object, a letter…) to Someone"

```
MOVE    SUBJ        var1: [var2]
        OBJ         var3
        [SOURCE     var4: [var5]]
        BENF        var6: [var7]
        [MODAL      var8]
        [TOPIC      var9]
        [CONTEXT    var10]
        { [modulators], ≠abs }

var1  =  human_being_or_social_body
var3  =  artefact_
var4  =  human_being_or_social_body
var6  =  human_being_or_social_body
var8  =  process_, sector_specific_activity, service_
var9  =  sortal_concept
var10 =  situation_, symbolic_label
var2, var5, var7 = location_

ex.c1)   MOVE   SUBJ      PIERRE_
                OBJ       ART_BOOK_1
                BENF      MARIE_
                MODAL     as_a_gift
                date-1:   2018-12-28
                date-2:
```

We can note that, to avoid the ambiguities of natural language and any possible combinatorial explosion problem, see [8: 56-61], both the conceptual predicate of Eq. 1 and the associated functional roles are "*primitives*". Predicates P_j belong to the set {BEHAVE, EXIST, EXPERIENCE, MOVE, OWN, PRODUCE, RECEIVE}, and the roles R_k to the set {SUBJ(ect), OBJ(ect), SOURCE, BEN(e)F(iciary), MODAL(ity), TOPIC, CONTEXT}. The HTemp hierarchy is structured, then, into *seven branches*, where each of them includes only the templates created – see Eq. 1 – around one of the seven allowed predicates P_j. HTemp includes presently (January 2019) more than 150 templates, very easy to specialize and customize, see in this context [8: 137-177].

As we can see from Table 1, in a template the arguments of the predicate (the a_k terms in Eq. 1) are actually represented by variables (var_i) with associated constraints. These last are expressed as concepts or combinations of concepts, i.e., using HClass terms – this confirms that the two NKRL's ontologies work in a strictly connected way. When creating a predicative occurrence as an instance of a given template, the constraints linked to the variables are used to specify the legal sets of HClass terms, concepts or individuals, that can be *substituted* for these variables within the occurrence. For example, in the situation of Table 1, we must verify that JOHN_ and MARY_ are true HClass instances of individual_person, a specific term of human_being_or_social_body, see the constraints on the SUBJ and BENF roles of the Table 1 template. The individual ART_BOOK_1 is an instance of the art_book concept, a specific term, through intermediate elements, of artefact_, see the constraint associated with OBJ in Table 1. as_a_gift is a specific term of the concept activity_related_property that is included in the qualifier_ sub-tree of HClass. Eventually, with respect to the instantiation morphism, we can note that this includes some *general procedures* – like the obligation to verify that all the constraints have been satisfied – and *a component proper to each specific template*, represented by the set of constraints associated with its variables var_i.

2.3 Second Order Structures

What described until now illustrates the NKRL solutions to the problem of representing single elementary events. In the context of larger dynamic situations like complex events, narratives, scripts, scenarios etc., several predicative occurrences corresponding to multiple elementary events must be associated through “connectivity phenomena” operators like causality, goal or indirect speech. In this case, we make use of *second order structures* created through the *reification* of the single occurrences. This is actually implemented utilizing their symbolic labels (the L_i terms of Eq. 1) according to two conceptual mechanisms. The first concerns the possibility of referring to an elementary (or complex) event *as an argument* of another (elementary) event – a “complex event” corresponds to a coherent set of elementary events. The (natural language) connectivity phenomenon involved here is the “indirect speech”. An example can be that of an elementary event *X* describing someone who speaks about *Y*, where *Y* is itself an elementary/complex event. In NKRL, this mechanism is called “*completive construction*” see [8: 87-91] and the occurrences breda.c5/breda.c6 in the example of Table 2 below.

The second (more general) process allows us to associate together, through several types of *connectivity operators*, elementary (or complex) events that, at the difference of the previous case, can still be regarded as fully independent entities. This relational mechanism is called “*binding occurrences*”, see [8: 91-98] and occurrences breda.c1 and breda.c2 in Table 2, and it is represented as labelled lists formed of a binding operator Bn_i and its L_i arguments. The general expression of a binding occurrence is then:

$$(Lb_k\ (Bn_i\ L_1\ L_2\ \ldots\ L_n))\ , \qquad (2)$$

where Lb_k is the symbolic label identifying the binding structure. The Bn_j operators are: ALTERN(ative), COORD(ination), ENUM(eration), CAUSE, REFER(ence), the weak causality operator, GOAL, MOTIV(ation), the weak intentionality operator, COND(ition). These structures are particularly important in an NKRL context. For example, as we will see in Section 3 (Table 2), the top-level knowledge patterns introducing the whole NKRL representation of any kind of structured dynamic knowledge entity (narrative, complex event, script, scenario…) necessarily have the general form of a *binding occurrence* (Eq. 2). They represent also an answer, in an NKRL context, to the remark raised by Blomqvist and colleagues [7], see Section 1, about the impossibility to understand how different ODPs patterns can relate to each other.

2.4 The Inference Procedures

"Reasoning", in NKRL, range from the direct questioning of knowledge bases (*KBs*) of NKRL entities using specific "*search patterns*" p_i to match/unify information in the base thanks to the use of a *Filtering Unification Module* (*FUM*) [8: 183-201], to complex inference operations based on the use of backward chaining *InferenceEngine*(s).

Besides offering the user the possibility of directly posing some questions to an NKRL knowledge base, the p_i search patterns can be also automatically generated by the *InferenceEngine*(s) during the high-level inference operations of NKRL, see sub-section 3.2 below. Formally, they represent a kind of knowledge patterns derived from particular (specialized and partially instantiated) templates of the HTemp hierarchy where, in particular, the "*explicit variables*" var_i that characterize the templates (see Table 1) have been replaced by concepts (or in some cases, individuals) *congruent* with the constraints linked to these variables in the original templates. Within a p_i, the concepts included in this pattern are used as "*implicit variables*". This means that, during the morphism-like operations executed by *FUM* to unify p_i with the predicative occurrences c_j in the *KB*, a p_i-included concept can match i) all the individuals present in the unified c_j that correspond to direct instances of this concept and, ii) all the concepts included in the c_j that, according to HClass, represent specializations of (are subsumed by) the p_i-included concept along with all their instances (individuals).

The high-level inference procedures of NKRL make use mainly of two kinds of rules, called "*transformations*" and "*hypotheses*", see [8: 201-234].

The transformation rules try to adapt, from a semantic point of view, a search pattern p_i that "*failed*" – i.e., that was unable to find a unification within the *KB* – to the actual contents of this base using a sort of *analogical reasoning*. This means that the transformations try to automatically convert p_i into one or more different patterns $p_1, p_2 \ldots p_n$ that are not strictly equivalent but only "*semantically close*" to the original one. A transformation rule is then formed of a left-hand side, the "*antecedent*" – i.e., the formulation, in search pattern format, of the question that failed – and of one or more right-hand sides, the "*consequent(s)*", providing one or more new search pattern(s) to be substituted for the original one – see, e.g., the example of Table 3 in sub-section 3.2. By denoting with A the antecedent and with Cs_i all the possible consequents, a transformation rule can be formalized as shown in Eq. 3; the $var_i \subseteq var_j$ restriction corresponds to a "*safety condition*" requiring that all the variables declared in the antecedent A are also included in the Cs_i consequent accompanied, in case, by additional variables.

$$A(var_i) \Rightarrow Cs_i(var_j), \quad var_i \subseteq var_j . \tag{3}$$

The hypothesis rules allow us to automatically build up a kind of *"causal" explication* for an event (a predicative occurrence) retrieved within the *KB*. These rules are expressed formally as "*biconditionals*" like:

$$X \text{ iff } Y_1 \text{ and } Y_2 \ldots \text{ and } Y_n \ , \tag{4}$$

where the "*head*" of the rule corresponds to the predicative occurrence c_j to be "*explained*" and the different reasoning steps Y_i – called "*condition schemata*" in a hypothesis context – must all give rise to a *positive result*. This means that, for each of them, *InferenceEngine* must be able to automatically produce at least a successful search pattern capable then, using *FUM*, of *effectively unifying* some information of the *KB*. In this case, the set of predicative occurrences $c_1, c_2 \ldots c_n$ retrieved by the condition schemata Y_i thanks to their conversion into p_i can be interpreted as *causal explications* – or, at least, as interpretations of the general context – of the original occurrence c_j.

To mention a well-known NKRL example [8: 205-2012], let us suppose we have directly retrieved the information: "Pharmacopeia, a USA biotechnology company, has received 64,000,000 dollars from the German company Schering in the context of its R&D activities"; this information corresponds then to pc_i (X). Using a "hypothesis" rule, we can construct a *causal explanation* for this event by retrieving in the *KB* information like: i) "Pharmacopeia and Schering have signed an agreement concerning the production by Pharmacopeia of a new compound", pc_1 (Y_1) and ii) "in the framework of this agreement, Pharmacopeia has actually produced the new compound", pc_2 (Y_2).

3 NKRL and the Representation of Iconographic Narratives

We will use, as an example, the NKRL representation (see Table 2) of the central scene of Diego Velazquez's picture concerning "The Surrender of Breda". This represents Ambrosio Spinola, Commander in Chief of the Spanish Army during the Eighty Years' War, receiving, on June 5, 1625, the keys to the city by Justinus van Nassau, governor of Breda. The main interest of this scene resides in the *benevolent attitude* of the winner, Spinola, towards the loser, van Nassau, a not so common behavior at that time.

3.1 NKRL Representation of "The Surrender of Breda"

As stated in sub-section 2.3, any NKRL representation of narratives necessarily begins with the creation of a binding occurrence (having the general format of Eq. 2) that specifies the tree structure of the representation. In our case, the top binding occurrence breda.c1 (Lb_k in Eq. 2) includes *three blocs logically equivalent*, see the use of the binding operator COORD(ination). The first bloc, breda.c2, includes in turn *four coordinated predicative occurrences*, where #breda.c6 is introduced by breda.c5 as *filler* of its CONTEXT role following the "completive construction" modalities, see again 2.3. Accordingly, breda.c5 and breda.c6 represent together a coherent entity that supplies the formalization of the most important narrative element of the scene: while he receives the keys to the city (breda.c6), Ambrosio Spinola *prevents* (PRODUCE activity_blockage) a Justinus' attempt to genuflect in front of him (breda.c5). activity_blockage is a specific term of activity_ in HClass, genuflecting_ a specific term of negative_relationship and then of relationship_. Note, see breda.c5, that the genuflecting_ concept has been reified through the transformation into a GENUFLECTING_1 individual to allow us to *reference unambiguously this term within several occurrences*. It appears then in breda.c7, where it is specified that the genuflecting gesture is both (COORD1) sketched_ (specific term of qualifier_ via general_characterising_property) and in_front_of (specific term of binary_relational_property) Ambrosio Spinola. Note that the "OWN OBJ property_ TOPIC…" knowledge patterns in a breda.c7 style are regularly used to describe the properties of specific *inanimate entities* that represent the fillers of the SUBJ role; the corresponding animate entities properties are normally declared making use of BEHAVE templates see, e.g., breda.c3 and breda.c4.

The formal rendering of Table 2 highlights the importance of the use in NKRL of the so-called "*complex arguments*" or "*expansions*", built up as lists introduced by operators like SPECIF(ication) and used as fillers, instead of simple HClass concepts/individuals, of functional roles in the predicative occurrences. They are created using the four "*AECS sub-language*" operators, see [8: 68-70]. In addition to SPECIF(ication), the attributive operator = S, AECS includes the disjunctive operator ALTERN(ative) = A, the distributive operator ENUM(eration) = E and the collective operator COORD(ination) = C – within predicative occurrences, this last is denoted as COORD1, see breda.c7, to differentiate it from the analogous COORD operator used in a binding occurrence context. The interweaving of these operators is controlled by a "*priority rule*" that forbids, e.g., the use of COORD1 lists within the scope of lists SPECIF – the inverse is perfectly legal see, e.g., breda.c7. "*Modulators*" – like the modulator obs(erve) in breda.c3/breda.c4 – represent an important category of *determiners* [8: 70-86] that apply to a well-formed template or predicative occurrence to particularize its meaning. obs(erve) means, in particular, that at the date associated with date-1, the information represented by the corresponding template/occurrence is certainly true.

We can note that the logical arrangement of a generic narrative (like, e.g., that of Table 2) can always be represented as some sort of *complex tree structure*, see Fig. 1. This remark can be considered as valid in general independently from the formalization adopted see, e.g., the "story trees" of Mani and Pustejovsky in [17].

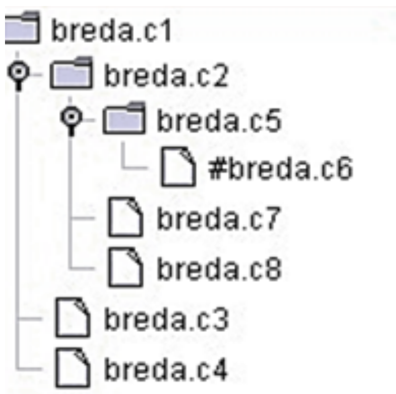

Figure 1. Tree-shaped representation of the Table 2 formalism.

3.2 Use of High-Level Inference Rules in an Iconographic Narratives Context

Setting up the formal representation of a complex narrative would not make much sense without the possibility of using this representation in the context of concrete applications. Storing the occurrences of Table 2 within an NKRL KB, we could then create search patterns able to recover factual data about, e.g., the status of Spinola at that time, "(BEHAVE (SUBJ AMBROSIO_SPINOLA) (MODAL army_role))", or about van Nassau's functions "(MODAL professional_role)", see the occurrences breda.c3 and breda.c4 of Table 2. Terms like army_role et professional_role act here as "implicit variables", see 2.4, able to match all their specific HClass terms along with their instances.

Table 2. Modeling of the central scene of “The Surrender of Breda”.

```
breda.c1)          (COORD  breda.c2  breda.c3  breda.c4)
```

The formalization of this iconographic narrative is formed of three main blocks.

```
breda.c2)          (COORD  breda.c5  #breda.c6  breda.c7  breda.c8)
```

The first block includes four predicative occurrences (# = completive construction).

```
breda.c5)  PRODUCE SUBJ     AMBROSIO_SPINOLA: (BREDA_)
                   OBJ      activity_blockage
                   MODAL    hand_gesture
                   TOPIC    (SPECIF GENUFLECTING_1 JUSTINUS_VAN_NASSAU)
                   CONTEXT  #breda.c6
                   date-1:  05/06/1625
                   date-2:
Produce:CreateCondition/Result (6.4)
```

(*Within the* breda.c6 *framework*), *Spinola stops van Nassau who is starting to genuflect.*

```
breda.c6)  RECEIVE SUBJ     AMBROSIO_SPINOLA: (BREDA_)
                   OBJ      (SPECIF key_to_the_city BREDA_)
                   SOURCE   JUSTINUS_VAN_NASSAU
                   CONTEXT  CELEBRATION_1
                   date-1:  05/06/1625
                   date-2:
Receive:TangibleThing (7.1)
```

Spinola receives the keys to the city from van Nassau in the context of particular celebrations.

```
breda.c7)  OWN  SUBJ   (SPECIF GENUFLECTING_1 JUSTINUS_VAN_NASSAU)
                OBJ    property_
                TOPIC  (COORD1 sketched_ (SPECIF in_front_of AMBROSIO_SPINOLA))
                date-1: 05/06/1625
                date-2:
Own:CompoundProperty (5.42)
```

Van Nassau’s genuflecting in front of Spinola is only scketched.

```
breda.c8)  OWN  SUBJ   CELEBRATION_1: (BREDA_)
                OBJ    property_
                TOPIC  (SPECIF surrender_ BREDA_)
                date-1: 05/06/1625
                date-2:
Own:CompoundProperty (5.42)
```

The celebrations are about the surrender of Breda.

```
breda.c3)  BEHAVE  SUBJ     AMBROSIO_SPINOLA
                   MODAL    commander_in_chief
                   TOPIC    SPANISH_ARMY
                   CONTEXT  EIGHTY_YEARS_WAR
                   { obs }
                   date-1:  05/06/1625
                   date-2:
Behave:Role (1.11)
```

On the 5th of June 1625, Ambrosio Spinola is the Commander in Chief of the Spanish Army.

```
breda.c4)  BEHAVE  SUBJ     JUSTINUS_VAN_NASSAU
                   MODAL    (SPECIF governor_ dutch_)
                   TOPIC    BREDA_
                   CONTEXT  EIGHTY_YEARS_WAR
                   { obs }
                   date-1:   05/06/1625
                   date-2:
Behave:Role (1.11)
```

By the same date, Justinus van Nassau is the Dutch governor of Breda.

More interesting results could be obtained using the NKRL's high-level inference procedures – provided, of course, to have at our disposal a "*real*" *KB*, broader than that the very reduced one represented by the six predicative occurrences of Table 2. To give, however, at least an idea of how these inference procedures could be used in a concrete context, we will make use of the few information available to try to infer some additional indications about the attitude of Ambrosio Spinola versus his opponent. To have a realistic chance to find some matches, we will ask whether Spinola's attitude is a "*positive*" one. We will use then the search pattern p_i of Table 3, derived from a partial instantiation of the template Behave:ConcreteVersusHumanAttitude.

This search pattern in itself is unable to find direct unifications with the Table 2 data. We can however imagine to find, within the *transformation rules repository* associated with a hypothetical iconographic narratives NKRL system, a (sufficiently general) rule stating that, "should a given person stop a submissiveness expression towards herself/himself from another person, this implies a positive attitude of the first person against the second". A formulation of this rule is given in Table 4.

Table 3. A search pattern about data of Table 2.

```
BEHAVE
SUBJ : AMBROSIO_SPINOLA :
OBJ :   JUSTINUS_VAN_NASSAU :
MODAL : positive_attitude
{}
date1 : 05/06/1625
date2 :
```

To activate the rule, we must check whether the p_j pattern to transform will be able to *unify* the left-hand side of the rule: in this case, the rule will be triggered and the *antecedent variables* will be bound to the terms associated with the corresponding roles of the pattern, *var1* = AMBROSIO_SPINOLA, *var2* = JUSTINUS_VAN_NASSAU. These values will be transferred to the first consequent schema (*conseq1*) in the right-side of the rule; this consequent schema, transformed into a new search pattern p_j and characterized by the presence of a *new variable*, *var3*, will try in turn to find unifications within the *KB*, producing then *new values* for the new variable. All these values will be transmitted to the second consequent, and so on; as already stated, the transformation will be validated iff all the consequents can find at least a valid unification within the base. In our example, the search pattern derived from *conseq1* will unify breda.c5 of Table 2; the value GENUFLECTING_1 – instance of genuflecting_, specific term of the negative_relationship constraint – will be linked to *var3* and transferred to the pattern derived from *conseq2*. This last will unify breda.c7. The two occurrences, breda.c5 and breda.c7, will be supplied then to the user as an "*indirect answer*" to the original question. Note that transformation *t41* of Table 4 conforms to the "*safety condition*" (see 2.4) since we can find in the right-side of the rule all the variables of the antecedent accompanied by two additional variables, *var3* et *var4*.

By considerably enlarging the embryonic *KB* of Table 2 we could use the hypothesis rules of NKRL to evaluate some of the "*possible reasons*" introduced to explain the behavior of Spinola versus his enemy and for advancing, in case, new ones. Among those already proposed, we can mention i) an astute propaganda operation to the benefit of the Spanish royal household, ii) the fact that the Spanish Army had really admired the bravery of the Dutch soldiers, iii) Velazquez's wish to promote a "Christian way" of conducting warfare, iv) Velazquez's friendship for Spinola, etc. Other interesting investigations paths could concern exploring the *possible influences* on Velazques exerted by well-known masterpieces dealing with similar topics, e.g., Rubens' "Meeting of King Ferdinand of Hungary and the Cardinal-Infante Ferdinand of Spain at Nördlingen" or "The reconciliation between Jacob and Esau".

Table 4. A transformation rule example.

t41: *"recovering from a submissive condition" transformation*

antecedent:

```
BEHAVE  SUBJ    var1
        OBJ     var2
        MODAL   positive_attitude

var1  =  individual_person
var2  =  individual_person
var1  ≠  var2
```

first consequent schema (*conseq1*):

```
PRODUCE  SUBJ    var1
         OBJ     activity_achievement
         TOPIC   (SPECIF var3 var2)

var3  =  negative_relationship
```

second consequent schema (*conseq2*):

```
OWN  SUBJ    var3
     OBJ     property_
     TOPIC   (SPECIF var4 var1)

var4  =  binary_relational_property
```

If a given person stops a submissiveness expression towards herself/himself from another person, this could imply a positive attitude of the first person against the second.

5 Conclusion

This paper suggests that, to deal in a suitable computerized way with the difficult knowledge representation and management problems proper to an important Cultural Heritage sub-field, the Iconographic Narratives domain – which concerns the "stories" related by paintings, drawings, frescoes, mosaics, sculptures, murals and so on, but also by pictures, posters, advertising artworks, comics, cartoons, movies etc. – simple tools based on quite generic notions of "pattern" are not sufficient. In this context, more powerful and specialized tools like NKRL, the "Narrative Knowledge Representation Language" – which makes use the very precise notion of "*knowledge patterns*" derived from the Software Engineering domain – must then be used. A concrete example of utilization of NKRL to supply a detailed formal description and some propositions of advanced exploitation of the iconographic narrative represented by central scene of "The Surrender of Breda" picture by Diego Velazquez is included in the paper.

References

1. Clark, P., Thompson, J., Porter, B.: Knowledge Patterns. In: Cohn, A.G. et al. (eds.) KR 2000, Proc. of the Seventh Int. Conference on Principles of Knowledge Representation and Reasoning, pp. 591-600. Morgan Kaufmann, Burlington MA (2000).
2. Blomqvist, E.: Semi-automatic Ontology Construction Based on Patterns (PhD thesis). School of Engineering of the Linköping University, Linköping (2009).
3. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: WonderWeb Deliverable D18 (IST Project 2001-33052). Laboratory for Applied Ontology, Trento (2003), http://www.loa.istc.cnr.it/old/Papers/D18.pdf, last accessed 2 January 2019.
4. ODP Portal Homepage, http://www.ontologydesignpatterns.org, last accessed 2 January 2019.
5. Kozaki, K., Sunagawa, E., Kitamura, Y., Mizoguchi, R.: Role Representation Model Using OWL and SWRL. In: Proc. of the 2nd Workshop on Roles in Object Oriented Programming and Ontologies, co-located with ECOOP '07 (2007), http://www.ei.sanken.osaka-u.ac.jp/pub/kozaki/Role07kozaki.pdf, last accessed 2 January 2019.
6. Trame, J., Kessler, C., Kuhn, W.: Linked Data and Time – Modeling Researcher Life Lines by Events. In: Tenbrink et al. (eds.) Proc. of the 11th Int. Conference on Spatial Information Theory, COSIT 2013, LNCS, vol. 8116, pp. 205–223. Springer, Heidelberg (2013).
7. Blomqvist, E., Hitzler, P., Janowicz, K., Krisnadhi, A., Narock, T., Solanki, M.: Considerations Regarding Ontology Design Patterns. Semantic Web 7(1), 1–7 (2016).
8. Zarri, G.P.: Representation and Management of Narrative Information, Theoretical Principles and Implementation. Springer, London (2009).
9. Bal, M.: Narratology: Introduction to the Theory of Narrative. University Press, Toronto (1997).
10. Jahn, M.: Narratology: A Guide to the Theory of Narrative (version 2.0). English Department of the Cologne University, Cologne (2017), http://www.uni-koeln.de/~ame02/pppn.htm, last accessed 2 January 2019.
11. Zarri, G.P.: A Conceptual Methodology for Dealing with Terrorism "Narratives". International Journal of Digital Crime and Forensics 2(2), 47–63 (2010).
12. Zarri, G.P.: Knowledge Representation and Inference Techniques to Improve the Management of Gas and Oil Facilities. Knowledge-Based Systems 24(7), 989–1003 (2011).
13. Zarri, G.P.: Sentiments Analysis at Conceptual Level Making Use of the Narrative Knowledge Representation Language. Neural Networks (Special Issue on "Affective Neural Networks and Cognitive Learning Systems for Big Data Analysis") 58, 82–97 (2014).
14. Amarilli, F., Amigoni, F., Fugini, M.G., Zarri, G.P.: A Semantic-Rich Approach to IoT Using the Generalized World Entities Paradigm. In: Sheng et al. (eds.) Managing the Web of Things, pp. 105-147. Morgan Kaufmann Elsevier, Cambridge MA (2017).

15. Noy, N., Fergerson, R.W., Musen, M.A.: The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility. In: Dieng, R., Corby, O. (eds.) Proc. of EKAW 2000, LNCS, vol. 1937, pp. 17–32. Springer, Heidelberg (2000).
16. Zarri, G.P.: Functional and Semantic Roles in a High-Level Knowledge Representation Language. Artificial Intelligence Review (AIRE), doi: 10.1007/s10462-017-9571-5 (2019).
17. Mani, I. Pustejovsky, J.: Temporal Discourse Models for Narrative Structure. In: Webber, B., Byron, D. (eds.) Proc. of the ACL Workshop on Discourse Annotation, pp. 57– 64. ACL, Stroudsburg PA (2004).