Welcome from the Conference Chairs

It was our pleasure to welcome you to the 22nd International Conference on Conceptual Structures, which was held at the University of Savoie in Annecy, France. The conference location offered ample opportunities for sightseeing and outdoor activities with the Mont-Blanc summit nearby.

The idea of a colocation of ICCS and the International Conference on Formal Ontology in Information Systems (FOIS) in the French Alps germinated in a few brains in Mumbai in January 2013, during ICCS 2013 whose audience was quite small. We wanted to revitalize ICCS by renewing the program committee around a preserved historic core, in order to build relationships with new communities. ICCS 2014 in Iasi in Romania was an important first step in that direction. For ICCS 2016 we have assembled an excellent technical program, and the colocation with the 9th International Conference on Formal Ontology in Information Systems offers even more opportunities to attend high-quality research presentations and additional workshops. That co-location which took place in Annecy, was unanimously supported by the ICCS Steering Committee, is a beautiful concretization of our will.

Two highlights of this year's program were the keynote addresses delivered by two renowned researchers. Dr Mateja Jamnik, from the Cambridge University Computer Laboratory, has made significant research contributions to the Diagrams and Automated Reasoning communities. Her research aims to investigate and mechanise human mathematical reasoning, such as the use of diagrams in proofs. Dr Jamnik delivered a keynote talk entitled *Automating human-like reasoning with diagrams on machines*. The topic of her talk reflected our drive to broaden the community from which ICCS aims to attract both paper submissions and delegates attending the conference. Dr Fabien Gandon is research Director in Informatics and Computer Science at Inria Sophia Antipolis méditerranée and the Inria representative at the World-Wide Web Consortium (W3C). His professional interests include: Web, Semantic Web, Social Web, Ontologies, Knowledge Engineering and Modelling, Mobility, Privacy, Context-Awareness, Semantic Social Network / Semantic Analysis of Social Network, Intraweb, Distributed Artificial Intelligence. Dr Gandon delivered a keynote entitled *On the many graphs of the Web and the interest of adding their missing links*.

ICCS 2016 received 40 full paper submissions from authors spanning 23 countries. These papers were independently and anonymously reviewed, typically by four members of the Program Committee. After the reviews and an initial round of discussions amongst the reviewers and Program Chairs, the authors were invited to submit rebuttals. To ensure a high quality program, the rebuttals were discussed and final versions of the reviews were produced. At the end of this process, the Program Chairs selected 14 full papers, giving a competitive acceptance rate of 35%. A further five short papers were accepted for presentation and publication alongside six posters.

As with any organizational task of this scale, very many people have contributed to the success of ICCS 2016. We would especially like to thank the members of the Program Committee for their expert reviews and for their excellent engagement with the two discussion phases. The Program Committee also invited other experts to provide reviews for some papers, and we are grateful for their important contributions. We would also like to thank the following individuals: Richard Dapoigny and Patrick Barlatier who made all of the local arrangements; Juliette Dibie, Liliana Ibanescu, Stéphane Dervaux for liaising with Springer and compiling the proceedings; Cassia Trojahn dos Santos for taking care of the conference finances and sponsorship; Sven Linker for publicizing the conference; Nicolas Seydoux and Rémi Cavallo for being the web master and web site designer respectively. We are also grateful to the Steering Committee for their support and for entrusting us to organize ICCS 2016: Madalina Croitoru, Frithjof Dau, Ollivier Haemmerlé, Uta Priss, and Sebastian Rudolph. Finally, we thank our sponsors: Université Savoie Mont Blanc and Inria for financially supporting the conference organisation, and AFIA, the French Society on AI, for sponsoring the Best Paper Award.

We hope that you found the conference and its co-located events exciting and stimulating.

Ollivier Haemmerlé
General Chair

Catherine Faron Zucker and **Gem Stapleton**
Program Co-Chairs

Organization

ICCS Executive

General Chair:	Ollivier Haemmerlé (Université Toulouse Jean Jaurès, France)
Co-program Chairs:	Gem Stapleton (University of Brighton, UK) and Catherine Faron-Zucker (Université Nice Sophia Antipolis, France)
Local chairs:	Richard Dapoigny (Université de Savoie, France) and Patrick Barlatier (Université de Savoie, France)
Steering Committee:	Madalina Croitoru (Université Montpellier 2, France) Frithjof Dau (SAP Research Dresden, Germany) Ollivier Haemmerlé (Université Toulouse Jean Jaurès, France) Uta Priss (Ostfalia University of Applied Sciences, Wolfenbttel, Germany) Sebastian Rudolph (Technical University Dresden, Germany)

ICCS Program Committee

Bernd Amann (France)
Simon Andrews (UK)
Galia Angelova (Bulgaria)
Peggy Cellier (France)
Dan Corbett (USA)
Olivier Corby (France)
Madalina Croitoru (France)
Cornelius Croitoru (Romania)
Licong Cui (USA)
Frithjof Dau (Germany)
Aldo De More (The Netherlands)
Harry Delugach (USA)
Juliette Dibie (France)
Pavlin Dobrev (Bulgaria)
Florent Domenach (Cyprus)
Ged Ellis (Australia)
Jérôme Euzenat (France)
Catherine Faron Zucker (France)
Jérôme Fortin (France)
Cynthia Vera Glodeanu (Germany)
Tarik Hadzic (Ireland)
Ollivier Haemmerlé (France)
Jan Hladik (Germany)
Rinke Hoekstra (The Netherlands)
John Howse (UK)
Dmitry Ignatov (Russia)
Mateja Jamnik (UK)
Adil Kabbaj (Morocco)
Mary Keeler (USA)
Hamamache Kheddouci (France)
Leonard Kwuida (Switzerland)
Jérôme Lang (France)
Steffen Lohmann (Germany)
Natalia Loukachevitch (Russia)
Pierre Marquis (France)
Philippe Martin (France)
Franck Michel (France)
Bernard Moulin (Canada)
Sergei Obiedkov (Russia)
Peter Øhrstrøm (Denmark)
Yoshiaki Okubo (Japan)
Nathalie Pernelle (France)

Heather D. Pfeiffer (USA)
Simon Polovina (UK)
Uta Priss (Germany)
Marie-Christine Rousset (France)
Sebastian Rudolph (Germany)
Fatiha Saïs (France)
Eric Salvat (France)
Atsushi Shimojima (Japan)
Iain Stalker (UK)
Gem Stapleton (UK)
Gerd Stumme (Germany)
Nouredine Tamani (France)
Annette Ten Teije (The Netherlands)
Michaël Thomazo (Germany)
Serena Villata (France)
Martin Watmough (UK)

Additional Referees

Wouter Beek
Jim Burton
Peter Chapman
Sara Maglicane
Martin Möhrmann
Antonio Penta
Brigitte Safar
Stephan Schulz
Nicolas Schwind

ICCS Administrative

Proceedings	Juliette Dibie, Liliana Ibanescu, Stéphane Dervaux
Finance, sponsors	Cassia Trojahn dos Santos
Publicity	Sven Linker
Website	Nicolas Seydoux
	Rémi Cavallo

Partners

AFIA (French Artificial Intelligence Association)
INRIA (French Institute for Research in Computer Science and Automation)
Université Savoie Mont Blanc

Table of Contents

Posters

Concept Lattices over Power Context Families

Jens Kötters

Abstract. Conjunctive queries can be represented by labeled graphs and are used in this paper as concept descriptions for concepts over relational data. Intension graphs are introduced as the basis for a formalization of conjunctive queries. Intension graphs are similar to concept graphs, but different in that they are purely intensional (i.e. without a realization component). Power context families are consequently used to model relational data, and realizations are used in this scenario to model pattern matches (i.e. solutions to a query).

Keywords: Conjunctive Queries, Pattern Concepts, Concept Graphs, Power Context Families

1 Base Patterns

Conjunctive queries correspond to logical formulas built from atoms using only conjunction ($\wedge$) and existence quantification ($\exists$). This section features relational structures, power context families and intension graphs as "base patterns" which can model relational data and (in combination with one or more designated elements, see Sect. 3) conjunctive queries alike.

1.1 Relational Structures

A *relational signature* is a set S of *relation symbols*, each having assigned an *arity* $k \geq 1$. By S_k we denote the subset of all k-ary relation symbols.

A *relational structure* over S (or simply an S-*structure*) is a pair $(A, \mathfrak{r})$ consisting of a *carrier set* A and a map $\mathfrak{r}$ which maps each symbol $R \in S_n$ to an n-ary relation $\mathfrak{r}(R) \in A^n$ ($n \geq 1$). We set $|(A, \mathfrak{r})| := A$ and $R^{(A,\mathfrak{r})} := \mathfrak{r}(R)$ for $R \in S$. A *homomorphism* $f : \mathfrak{A} \to \mathfrak{B}$ of relational structures is a map $f : |\mathfrak{A}| \to |\mathfrak{B}|$ such that $(x_0, \dots, x_{n-1}) \in R^{\mathfrak{A}}$ implies $(f(x_0), \dots, f(x_{n-1})) \in R^{\mathfrak{B}}$ for all $R \in S_n$, $n \geq 1$.

The class of all S-structures which satisfy a set Φ of axioms (first-order sentences over the signature S), together with structure homomorphisms, forms the category $\mathbf{Str}_{(S,\Phi)}$.

1.2 Power Context Families

A *power context family* is a sequence $((G_i, M_i, I_i))_{i\in\mathbb{N}}$ of formal contexts where $G_i \subseteq (G_0)^i$ for all $i \geq 1$.

A *system of attribute implications* is a sequence $((M_i, \to_i))_{i\in\mathbb{N}}$, where each M_i is an attribute set, and $\to_i$ is a set of attribute implications over M_i.

A power context family over a system $((M_i, \to_i))_{i\in\mathbb{N}}$ of attribute implications is a power context family $(\mathbb{K}_i)_{i\in\mathbb{N}}$ such that M_i is the attribute set of $\mathbb{K}_i$ and $\to_i$ holds in $\mathbb{K}_i$ for all $i \in \mathbb{N}$.

A *homomorphism* $\varphi : ((G_i, M_i, I_i))_{i\in\mathbb{N}} \to ((H_i, M_i, J_i))_{i\in\mathbb{N}}$ of power context families over $\mathcal{M}$ is a map $\varphi : G_0 \to H_0$, extended to tuples by setting

$$\varphi((g_1, \ldots, g_k)) := (\varphi(g_1), \ldots, \varphi(g_k)) \tag{1}$$

for $(g_1, \ldots, g_k) \in (G_0)^k$, $k \geq 1$, which satisfies

$$gI_km \Rightarrow \varphi(g)J_km \tag{2}$$

for all $k \in \mathbb{N}$, $g \in G_k$ and $m \in M_k$. We call φ a *row homomorphism*, if

$$\varphi(G_k) \subseteq H_k \tag{3}$$

holds for all $k \geq 1$.

For a given system $\mathcal{M}$ of attribute implications, let $\mathbf{PCF}_{\mathcal{M}}$ denote the category of all power context families respecting $\mathcal{M}$, with homomorphisms taking the role of morphisms. By $\mathbf{PCF}^+_{\mathcal{M}}$ we denote the category with the same objects as $\mathbf{PCF}_{\mathcal{M}}$, but with row homomorphisms as its morphisms.

1.3 Intension Graphs

A *multi-system of intents* is a sequence $((M_i, \mathcal{H}_i))_{i\in\mathbb{N}}$, where M_i is an attribute set and $\mathcal{H}_i \subseteq \mathcal{P}(M_i)$ is a closure system over M_i ($i \in \mathbb{N}$).

A *simple relational graph* is a triple (V, E) consisting of a set V of vertices and a set $E \subseteq \bigcup_{k\geq 1} V^k$ of edges. An edge $e \in V^k$ is said to have *arity* k. We set $E^{(0)} := V$ and $E^{(k)} := E \cap V^k$, $k \geq 1$.

An *intension graph* over a multi-system $((M_i, \mathcal{H}_i))_{i\in\mathbb{N}}$ of intents is a triple (V, E, κ) where (V, E) is a simple relational graph and $\kappa : V \cup E \to \bigcup_{k\in\mathbb{N}} \mathcal{H}_k$ is a map with $\kappa(E^{(k)}) \subseteq \mathcal{H}_k$, $k \in \mathbb{N}$.

A *homomorphism* $\varphi : (V_1, E_1, \kappa_1) \to (V_2, E_2, \kappa_2)$ of intension graphs over a given multi-system $\mathcal{H}$ of intents is a map $\varphi : V_1 \to V_2$ which satisfies

$$\varphi(e) \in E_2, \tag{4}$$

$$\kappa_1(u) \subseteq \kappa_2(\varphi(u)) \tag{5}$$

for all $e \in E_1$ and $u \in V_1 \cup E_1$, where φ is extended to tuples as in (1). We denote by $\mathbf{IG}_{\mathcal{H}}$ the category of intension graphs over $\mathcal{H}$.

2 Equivalence of Formalisms

This section defines functors between the formalisms. The functors from power context families to relational structures and intension graphs organize data column-wise and row-wise, respectively. In this sense, relational structures and intension graphs can be considered dual to each other, as reflected in the equations below. Homomorphisms (as defined in the natural way) do not correspond exactly, however.

2.1 Power Context Families and Relational Structures

Let $\mathcal{M}$ be a system of attribute implications. We denote by $S(\mathcal{M})$ the relational signature with $S_1 := M_0 \cup M_1$ and $S_i := M_i$ for $i > 1$. Moreover, let $\Phi(\mathcal{M}) \subseteq L^{S(\mathcal{M})}$ contain for each $m_1, \ldots, m_k \to m \in M_n$, $n \geq 1$, the formula

$$\varphi_{m_1,\ldots,m_k \to m} \quad := \quad \forall x_1 \ldots \forall x_n \bigwedge_{1 \leq i \leq n} m_i(x_1, \ldots, x_n) \quad \to \quad m(x_1, \ldots, x_n), \tag{6}$$

and nothing else. A functor $\mathrm{rstr}_{\mathcal{M}} : \mathbf{PCF}_{\mathcal{M}} \to \mathbf{Str}_{(S(\mathcal{M}),\Phi(\mathcal{M}))}$ and a right inverse functor $\mathrm{pcf}_{\mathcal{M}} : \mathbf{Str}_{(S(\mathcal{M}),\Phi(\mathcal{M}))} \to \mathbf{PCF}_{\mathcal{M}}$ are given by

$$\mathrm{rstr}_{\mathcal{M}}(\vec{\mathbb{K}}) := (G_0, \{m \mapsto m' \mid m \in \bigcup_{i \in \mathbb{N}} M_i\}), \tag{7}$$

$$\begin{aligned} \mathrm{pcf}_{\mathcal{M}}(\mathfrak{A}) := ((|\mathfrak{A}|, \emptyset, \emptyset), (|\mathfrak{A}|^1, S_1, \in_{\mathfrak{A}}^{(1)}), (|\mathfrak{A}|^2, S_2, \in_{\mathfrak{A}}^{(2)}), \ldots), \\ \text{where } u \in_{\mathfrak{A}}^{(k)} m :\Leftrightarrow u \in m^{\mathfrak{A}} \end{aligned} \tag{8}$$

for all $(u, m) \in |\mathfrak{A}|^k \times S_k$ and $k \in \mathbb{N}$.

2.2 Power Context Families and Intension Graphs

Let $\mathcal{M} =: ((M_i, \mathcal{M}_i))_{i \in I}$ be a system of attribute implications. Every implication set $\mathcal{M}_i$ induces a closure system $\mathcal{M}_i^*$ of intents on M_i, and by extension $\mathcal{M}$ induces a multi-system

$$\mathcal{H}(\mathcal{M}) := ((M_i, \mathcal{M}_i^*))_{i \in I} \tag{9}$$

of intents.

Define $\mathrm{ig}_{\mathcal{M}} : \mathbf{PCF}^+{}_{\mathcal{M}} \to \mathbf{IG}_{\mathcal{H}(\mathcal{M})}$ and $\mathrm{pcf}_{\mathcal{M}} : \mathbf{IG}_{\mathcal{H}(\mathcal{M})} \to \mathbf{PCF}^+{}_{\mathcal{M}}$

$$\mathrm{ig}_{\mathcal{M}}(\vec{\mathbb{K}}) := (G_0, \bigcup_{i \geq 1} G_k, \{u \mapsto u' \mid u \in \bigcup_{i \in \mathbb{N}} G_i\}), \tag{10}$$

$$\begin{aligned} \mathrm{pcf}_{\mathcal{M}}(\mathfrak{G}) := ((E^{(k)}, M_k, \ni_{\mathfrak{G}}^{(k)}))_{k \in \mathbb{N}}, \\ \text{where } u \ni_{\mathfrak{G}}^{(k)} m :\Leftrightarrow m \in \kappa(u) \end{aligned} \tag{11}$$

3 Conjunctive Queries

We state a formalization of conjunctive queries using intension graphs, and define a Galois connection which relies on the categorical product. As shown by the functors in Sect. 2, the following definitions and results carry over to power context families, and almost carry over to relational structures.

3.1 Windowed Intension Graphs

A *windowed intension graph* is a pair $(x, \mathfrak{G})$ consisting of an intension graph $\mathfrak{G}$ and a node $x \in V_{\mathfrak{G}}$. A *homomorphism* $f : (x_1, \mathfrak{G}_1) \to (x_2, \mathfrak{G}_2)$ is a homomorphism $f : \mathfrak{G}_1 \to \mathfrak{G}_2$ with $f(x_1) = x_2$.

3.2 Products

The *product* of a family $((V_i, E_i, \kappa_i))_{i\in I}$ of intension graphs is the intension graph $\bigtimes_{i\in I}(V_i, E_i, \kappa_i) := (V, E, \kappa)$, where

$$V := \bigtimes_{i\in I} V_i, \tag{12}$$

$$(x_1, \dots, x_k) \in E^{(k)} :\Leftrightarrow \forall i \in I : (x_1(i), \dots, x_k(i)) \in \bigtimes_{i\in I} E_i^{(k)}, \tag{13}$$

$$\kappa(u) := \bigcap_{i\in I} \kappa_i(u) \tag{14}$$

for $k \geq 1$, $x_1, \dots, x_k \in V$ and $u \in V \cup E$. The *product* of a family $((x_i, \mathfrak{G}_i))_{i\in I}$ of windowed intension graphs is then given by

$$\bigtimes_{i\in I}(x_i, \mathfrak{G}_i) := ((x_i)_{i\in I}, \bigtimes_{i\in I} \mathfrak{G}_i). \tag{15}$$

3.3 Concept Lattice

A *realization* $\rho : \mathfrak{G} \to \vec{\mathbb{K}}$ is a map $\rho : V \to G_0$ with $\rho(u) \in \kappa(u)'$ for all $u \in V \cup E$, where $\rho((x_1, \dots, x_k)) := (\rho(x_1), \dots, \rho(x_k))$ for $x_1, \dots, x_k \in V$, $k \geq 1$.

The concept lattice is defined as a lattice of pattern concepts by the following Galois connection (cp. [1]):

$$A^\diamond := \bigtimes_{g\in A}(g, \mathrm{ig}_{\mathcal{M}}(\vec{\mathbb{K}})), \tag{16}$$

$$(x, \mathfrak{G})^\diamond := \{\rho(x) \mid \rho : \mathfrak{G} \to \vec{\mathbb{K}}\} \tag{17}$$

for all $A \subseteq G_0$, $\mathfrak{G} \in \mathbf{IG}_{\mathcal{H}(\mathcal{M})}$ and $x \in V_{\mathfrak{G}}$. It is sensible to identify a pattern intent (as defined in (16)) with the connected component which contains the single designated node. A similar Galois connection has been presented in [2].

A Python module (which can display intension graphs when used with the IPython notebook) is hosted at `https://github.com/koetters`. In the implementation, intension graphs are required to have non-empty edge labels (i.e. $\kappa(e) \neq \emptyset$ for $e \in E_{\mathfrak{G}}$), which leads to a slightly different product in Sect. 3.2 and full correspondence to relational structures.

References

1. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Delugach, H.S., Stumme, G. (eds.) Proceedings of ICCS 2001. LNCS, vol. 2120, pp. 129–142. Springer (2001)
2. Kötters, J.: Concept lattices of a relational structure. In: Pfeiffer, H.D., Ignatov, D.I., Poelmans, J., Gadiraju, N. (eds.) Proceedings of ICCS 2013. LNCS, vol. 7735, pp. 301–310. Springer (2013)

Discovering the Gaps in Enterprise Systems via Conceptual Graphs & Formal Concept Analysis

Simon Polovina[1], Hans-Jürgen Scheruhn[2],
Stefan Weidner[3], and Mark von Rosing[4]

[1] Conceptual Structures Research Group, Sheffield Hallam University, UK
[2] Business Informatics, Hochschule Harz, Wernigerode, Germany
[3] SAP UCC & School of Computer Science, University Magdeburg, Germany
[4] LEADing Practice ApS | Global Univerisity Alliance, Denmark
s.polovina@shu.ac.uk, hscheruhn@hs-harz.de,
weidner@ucc.uni-magdeburg.de, mvr@leadingpractice.com

Abstract. Enterprise systems such as SAP are software applications that are intended to bring the productivity of computers to bear on the human endeavour of enterprise. An industrial-strength SAP enterprise information model was rendered as meta-object→relation→meta-object in Conceptual Graphs (CGs). Then Formal Concept Analysis (FCA)'s *CGtoFCA* algorithm was used to generate the meta-object⌒relation→meta-object binaries, revealing gaps in some of model's key performance indicators that human decision-makers need to realise the enterprise's vision.

1 Introduction

Enterprise systems are software applications that are intended to bring the productivity of computers to bear on the human endeavour of enterprise [3]. The SAP University Alliances program (http://uac.sap.com) includes a set of industrial-strength case studies to demonstrate SAP including an enterprise information model [1, 5]. Conceptual Graphs (CGs) and Formal Concept Analysis (FCA) were used to abstract the conceptual structures from the business and systems layers in this model [2]. Our purpose was to illustrate how gaps in the alignment between the associated human-oriented business concepts and their counterpart computer-oriented structures could be discovered.

2 Outcomes

By expressing the model's underlying meta-object→relation→meta-object triples as Conceptual Graphs (CGs) we brought together into one simple, common conceptual structure the model's myriad business and systems modelling notations (e.g. Value chain diagrams, Organisation Charts, UML, BPMN, Data models). The simplification extended to rendering the model's i) Value, Competency, Service, Process, Application, and Data layers across ii) the four levels of detail for each of these layers. The rationale for the layers and levels are detailed

Fig. 1. Application, CGs

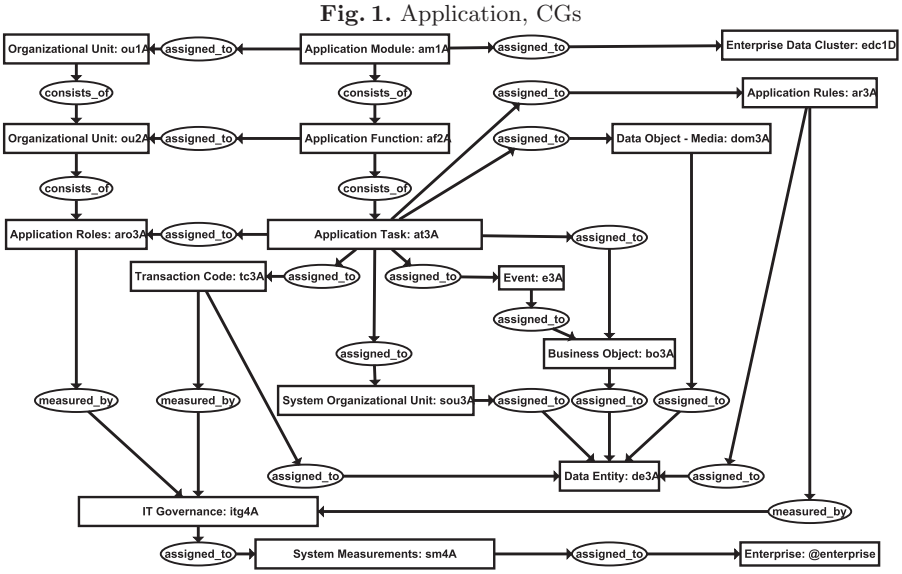


Fig. 2. Application, FCA

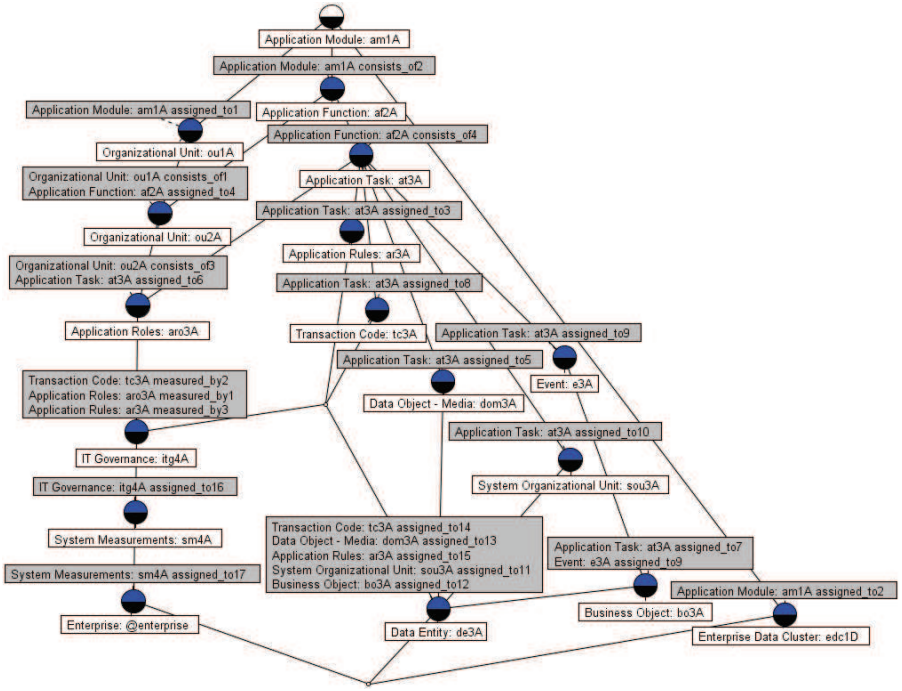


Fig. 3. Data, CGs

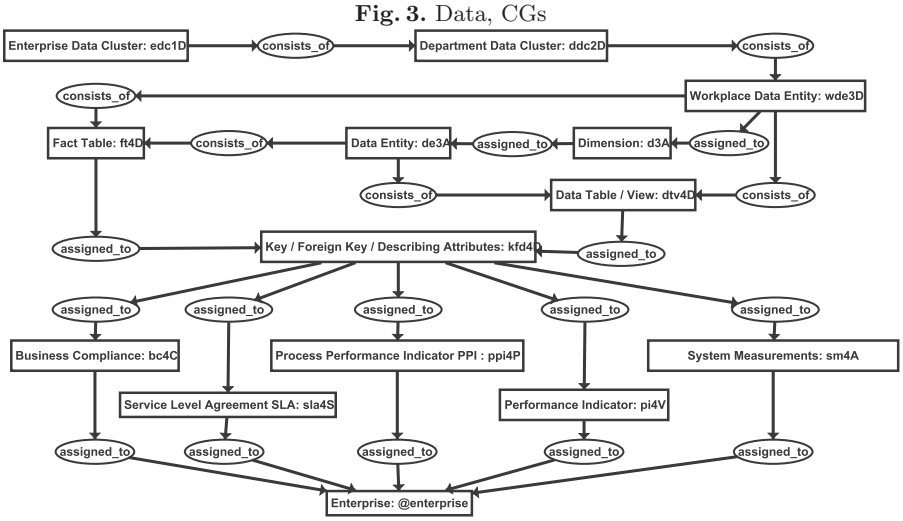
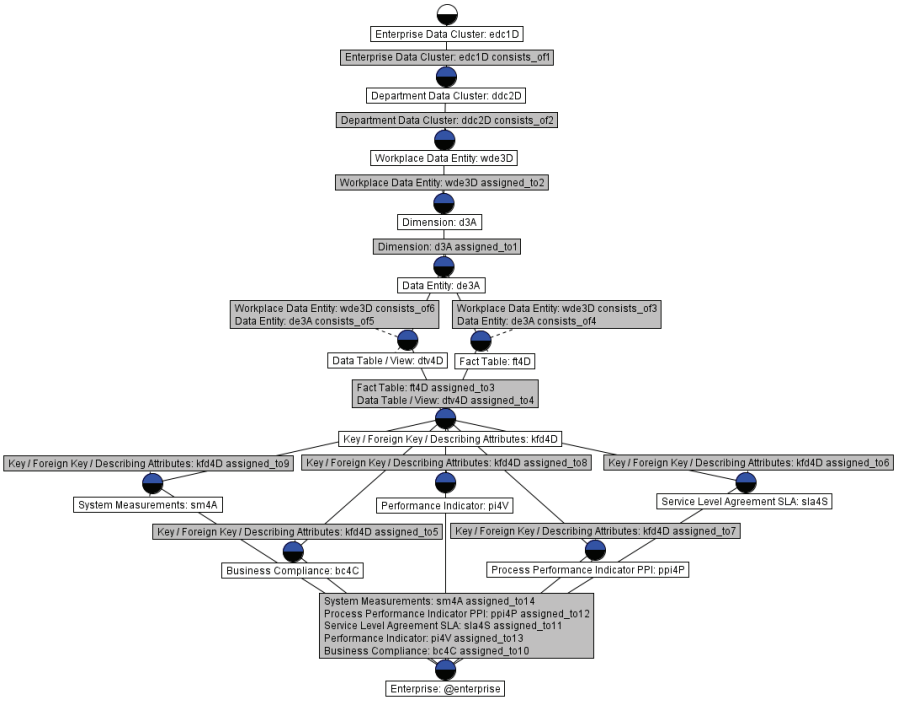


Fig. 4. Data, FCA



elsewhere [4]. Put simply, their intention is to align the human concepts and computer structures that realise an enterprise's vision i.e. its purpose for existing. Formal Concept Analysis (FCA)'s *CGtoFCA* algorithm then generated the meta-object⌢relation→meta-object binaries [2]. Figure 1 depicts the model's Application layer CG (Conceptual Graph). It portrays each meta-object as a CG concept made up of a CG type label of the meta-object name (e.g. Organisational Unit) and a unique identifier in the referent field e.g. `ou1A` denotes a meta-object that is Organisational Unit (ou), level 1 (1), and A (Application layer) from the model. The `[Enterprise: @enterprise]` concept has a CGs measure referent (`@enterprise`). For the model to be aligned as described earlier, each meta-object⌢relation (an attribute in FCA generated by *CGtoFCA*) should be in the intent of the bottommost formal concept at `[Enterprise: @enterprise]` [2].

On inspecting this CG the `[Application Module: am1A]` → `(assigned_to)` → `[Enterprise Data Cluster: edc1D]` CG triple therein does not point directly or indirectly to `[Enterprise: @enterprise]`. Likewise for the three meta-objects (`sou3a`, `bo3a`, and `dom3a`) that are `(assigned_to)` → `[Data Entity: de3A]`. Using index numbers to illustrate that the relations are distinct from each other (e.g. `assigned_to`N where $N \geq 1 \leq 17$), the *CGtoFCA* Formal Concept Lattice (FCL) figure 2 evidences that `[Enterprise: @enterprise]` is not bottommost. That is because of the meta-object⌢relation attributes that are outside the intent of the level 4 key performance indicator (KPI) meta-object `[System Measurements: sm4A]`, which evaluates the Application layer. These gaps however do not occur in the Data layer - see figures 3 and 4 respectively. Here the extent of all the attributes is `[Enterprise: @enterprise]` including from all the relevant KPIs (level 4 meta-objects) including `[System Measurements: sm4A]`. Gaps also occurred in other layers (Competency, Service, Process). Given that human decision-makers rely on these KPIs to realise the enterprise's vision, the gaps demonstrate how their misalignment with an enterprise system can be discovered, thus focussing any further investigation and the gaps resolved.

References

1. Simha R Magal and Jeffrey Word. *Integrated business processes with ERP systems*. Wiley Publishing, 2011.
2. Simon Polovina and Simon Andrews. Cgs to fca including peirce's cuts. *International Journal of Conceptual Structures and Smart Applications (IJCSSA)*, 1(1):90–103, 2013.
3. August-Wilhelm Scheer and Alexander Hars. Extending data modeling to cover the whole enterprise. *Communications of the ACM*, 35(9):166–ff, 1992.
4. Hans-Jürgen Scheruhn, Richard Fallon, and Mark Rosing. *Information Modelling and Process Modelling*, volume 1 of *The Complete Business Process Handbook. Body of Knowledge from Process Modelling to BPM*, pages pp 511–550. Elsevier, 2015.
5. Hans-Jürgen Scheruhn, Sandro Sicorello, and Stefan Weidner. *Business Information Systems Workshops: BIS 2012 International Workshops and Future Internet Symposium, Vilnius, Lithuania, May 21-23, 2012 Revised Papers*, chapter Repository-Based ERP Case Studies: A Study about Chances and Benefits of Agile Case Study Development, pages 186–197. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

Deriving Binary Relation Types From Concept Types

Philippe A. Martin [1] and Jérémy Bénard [2]

[1] EA2525 LIM, ESIROI I.T., University of La Réunion, F-97490 Sainte Clotilde, France
+ adjunct researcher of the School of I.C.T. at Griffith University, Australia
Philippe.Martin@univ-reunion.fr
[2] GTH, Logicells, 55 rue Labourdonnais, 97400 Saint-Denis, France
Jeremy.Benard@logicells.com

Abstract. This article describes an ontology design pattern. It helps normalizing knowledge, reducing the introduction of new relation types and keeping all the types organized. Thus, it leads knowledge providers to represent knowledge in more normalized, precise and inter-related ways, hence in ways that help the matching and exploitation of knowledge from different sources. This pattern is also a knowledge sharing best practice that is domain and language independent. It can be used as a criteria for measuring the quality of an ontology.

Keywords: Knowledge sharing, best practices, relation type generation

1 Introduction

Ontology Design Patterns (ODPs) are "modeling solutions to solve a recurrent ontology design problem" [1]. Many ODPs have been found, e.g., about 160 are currently registered in the "ODP catalog at http://ontologydesignpatterns.org". However, the thousands of ontologies (UML schemas included) that exist on the Web are poorly inter-connected and heterogeneous in their design. It is then difficult for automated agents to *compare or match* such independently created knowledge representations (KRs, e.g., types or statements) to know if some KRs are equivalent to others or specializations of others. Thus, it is difficult for people and automated agents to *align and aggregate* – and, thus relate, infer from, search or exploit – KRs or ontologies.

In other words, there is a need for ODPs specifically aimed for knowledge sharing and, more precisely, for solving *the problem of leading knowledge providers to create more matchable and re-usable KRs*. As later detailed, this implies leading them to create more precise, normalized, well related and easy-to-understand KRs. In order to be adopted, these ODPs should also be easy to follow and easy to use as criteria for automatically measuring the quality of an ontology, to help developing an ontology or selecting ontologies to re-use. Finally, the ODPs – or, at least the knowledge sharing ODPs – should be well inter-related by semantic relations to help people i) know about them and their advantages, and ii) select those they want to commit to. Then, tools can check or enforce these commitments.

This article proposes such a knowledge sharing focused ODP which is also a best practice (BP). This BP, which in this article will now be referred to as ABP, is: *"using binary relation types directly derived from concept types"*. No ODP catalog appears to include similar ODPs. Like most BPs, it is domain and language independent and it can be used for any dataset. The interest of using binary relation types only for KR is well understood, if only because many KRLs (KR languages) only support binary relations. Hence, this article does not explain this interest.

2 Deriving Relation Types from Concept Types

In this article, types that are not relation types (RTs) are refered to as "concept types" (CTs; "classes" in RDF and OWL). A relation is not a type. Since ABP is language independent, this article uses a general terminology. It is compatible with those for Conceptual Graphs, RIF-FLD [2] and the W3C Framework for Logic Dialects of the Rule Interchange Format.

When buildings RCs, it is better to use RTs with definitions than without. It makes some information explicit and ensures that every distinction in the (subtype) hierarchy of RTs is also included in the CT hierarchy. This last point is important for two reasons. First, it avoids that some knowledge providers develop distinctions only in the RT hierarchy while others develop distinctions only in the CT hierarchy, thus leading to *(automatically) undetected redundancies* within a shared knowledge base or in different ontologies. Second, it ensures that any distinction can be used – *without loosing in knowledge representation and matching possibilities* – with both its CT form and its RT form. More possibilities come from the CT form since i) unlike RTs, CTs can be quantified in many different ways (e.g., "3 landings", "all landings" or "8% of landings" can only be described via the CT "Landing", not the RT "r__landing"), ii) it is easier to organize CTs (by subtype relations) than RTs, and iii) the number of used or re-usable existing CTs is much greater than the number of used or re-usable RTs.

These advantages of using *defined RTs* come for free when the RTs are automatically derived from CTs and hence defined with respect to them. *Furthermore*, such derivations permits a system to display fewer types in the RT hierarchy (which is then easier to read and grasp). Indeed, the derived RTs may be left hidden or may not have to be created all. This last option was used in the knowledge server Ontoseek [3] and is used in the knowledge base server WebKB (www.webkb.org; [4]). In Ontoseek, any type derived from the noun-related part of the lexical ontology Sensus could be re-used as a CT or a RT. WebKB also re-uses a lexical ontology derived from WordNet – the "Multi-Source Ontology" (MSO [5]). However, unlike Ontoseek, WebKB only allows the subtypes of certain types to be re-used as RTs. This is defined by specifications that users can adapt. More precisely, this is defined by relation signatures which are directly associated to certain top-level CTs. The MSO includes more than 75,000 categories (mainly types) and relates them by more than 100,000 relations.

Table 1. RIF-FLD PS rules for automatically deriving a binary RT from a CT (and, if needed, doing so for all its subtypes) based on a kind of signature associated to this CT. In these examples, the types created by the authors of this article have no prefix to indicate their namespace. RIF-FLD PS, the Presentation Syntax of RIF-FLD, is used because it is both expressive and rather intuitive. RT names begin by "r__" and function names begin by "f__". Logical rules are used since RIF-FLD is used but here logical equivalences could also be used. ":-" means "<=".

The derived RT does not have to be explicitly defined. Its signature is directly associated to the CT via a relation of type r__signature_for_derived_binary_relation or a function of type f__derived_binary_relation. Thanks to their definitions, the derived RT is automatically created (see the next paragraph in bold characters). A CT may have different RT signatures associated to it, as long as the signatures are "un-comparable" (i.e., as long as none specializes another).

```
r__signature_for_derived_binary_relation ( Father    List ( Animal  Male ) )
    //-> derives the RT  r__father  that has for domain an Animal and range a Male

Forall  ?t  (  r__signature_for_derived_binary_relation ( ?t   List ( Thing  ?t ) )
              :-   ?t  ##  thing_usable_for_deriving_a_binary_relation_with_it_as_destination
   //"?st  ## ?t" <=> subtypeOf (?st ?t); this rule derives the expected RT for each subtype of
   //      Thing_usable_for_deriving_a_binary_relation_with_it_as_destination

Forall  ?t   Exists ?r
  And (  ?r = f__ derived_binary_relation ( ?t   List ( Agent  Object ) )
         Forall  ?agent  ?object    And ( r__agent (?t  ?agent)   r__ object (?process  ?object)
                                          ) :-  ?r (?agent  ?object)
       ) :-   ?t  ##  Process      //-> derives the expected RT for each subtype of Process
```

Furthermore, the derived RTs have the same subtype relations as the CTs they derive from. However, to keep things simple, it is here assumed that no RT with the same name as the derived RT has previously been manually created. The RT name is created by taking the CT name, lowering its initial and prefixing it with "r__". The functions f__denotation_of_type_name, f__type_name, f__cons, f__cdr, f__lowercase used below are identical to their counterparts (without the prefix "f__") in KIF.

```
Forall  ?t   ?r__t    ?t_domain   ?t_range
        ?t_supertype    ?r__t_supertype    ?t_sup_domain   ?t_sup_range  (
    And (  rdfs:domain (?r__t   ?t_domain )       rdfs:range (?r__t   ?t_range )
           ?r__t  =  f__denotation_of_type_name
                     ( f__cons ( f__lowercase ( f__car ( f__type_name ( ?t ) ) )
                       f__cdr ( f__name ( ?t ) )    )
           ?r__t    #   ?r__t_supertype
              :-  And (  ?t   #  ?t_supertype
                         ?r__t_supertype  =  f__ derived_binary_relation
                                             ( ?t_supertype
                                               List ( ?t_sup_domain  ?t_sup_range ) )    )  )
    :-   ?r__t   =   f__ derived_binary_relation ( ?t   List ( ?t_domain  ?t_range ) )   )

Forall  ?t   ?t_domain   ?t_range (
    Exists  ?r__t  ( ?r__t   =   f__ derived_binary_relation ( ?t    List ( ?t_domain  ?t_range ) ) )
     :-   f__signature_for_derived_binary_relation ( ?t   List ( ?t_domain  ?t_range ) )  )
```

Table 1 illustrates the approach and then gives rules that would actually generate the derived RTs. These rules permit to formalize the framework. They rely on the functions *f__type_name* and *f__denotation_of_type_name* which are identical to the KIF functions *name* and *denotation* formalized in the documentation of KIF [6]. In WebKB, no such rules are executed (a more efficient and ad-hoc process is used): during the analysis of RCs, when a CT is used in places where RTs are expected, WebKB simply checks that one of the signatures associated to the CT is respected and acts as if the relevant derived RT was actually used. Thus, in WebKB, there is no need to use the actual names of the virtually derived RTs: the CT names can be used directly. As described by Table 1, signatures are inherited along subtype relations between CTs and an error is generated if a CT is associated to two signatures that are "comparable". This approach and ODP seem original.

3 Acknowledgment

We thank one of our reviewers for its very positive review on our initially submitted article.

4 References

1. Presutti, V., Gangemi, V.: Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies. In: ER 2008, Spaccapietra S. et al. (eds.)
2. Boley, H., Kifer, M. (eds.): RIF Framework for Logic Dialects (2nd edition). W3C Recommendation, http://www.w3.org/TR/2013/REC-rif-fld-20130205/ (2013)
3. Guarino, N., Masolo, C., Vetere, G.: Ontoseek: Content-based Access to the Web. IEEE Intelligent Systems, vol. 14, No. 3, pp. 70–80 (1999)
4. Martin, Ph.: Collaborative knowledge sharing and editing, IJCSIS, vol. 6, Issue 1, pp. 14–29 (2011)
5. Martin, Ph.: Correction and Extension of WordNet 1.7. In: LNAI 2746, pp. 160–173. See also http://www.webkb.org/doc/MSO.html
6. Genesereth, M., Fikes R.: Knowledge Interchange Format, Version 3.0, Reference Manual. Technical Report, Logic-92-1, Stanford Uni., http://www.cs.umbc.edu/kse/ (1992)

Explanation Dialogues for Erroneous SameAs via Argumentation Theory

Abdallah Arioua[1], Madalina Croitoru[2], Laura Papaleo[4,5], Nathalie Pernelle[3], Swan Rocher[2]
[1] GraphIK, INRA, Montpellier, France
[2] GraphIK, Univ. of Montpellier, France
[3] LaHDAK, LRI, Univ. Paris Sud, France
[4] ICT Department, Metropolitan City of Genoa, Italy
[5] Tetherless World Constellation, Rensselaer Polytechnic Institute, USA

Abstract. Due to the impressive growing of the LOD graph in the last years, assuring the quality of its content is becoming a very important issue. We focus on identity links (*sameAs*) and present a novel idea which applies argumentation semantics to detect inconsistencies in sameAs statements.

1 Introduction, first results and discussion

Today, we are experiencing an unprecedented production of resources, published as Linked Open Data, and we are moving to the creation of a *global data space* with billions of available assertions [4]. RDF [5] provides formal ways to build these assertions. Most of the RDF links, connecting resources coming from different data sources, are *identity links*, also called *sameAs statements*. Here, we show how inconsistency-tolerant semantics (e.g. [3, 6]) for assessing the quality of sameAs statements can represent a first step in the direction of the design of new interactive paradigms, which could support the experts in improving the overall quality of complex knowledge bases.

We present a novel interactive framework, called an *explanation dialogue* that, by considering a knowledge base and a set of sameAs with inconsistencies, uses *argument-based explanation* to provide elucidations to the user.

Thanks to our explanation framework, the domain expert can interact with the reasoner regarding a *problematic sameAs* discovering, for example, that the sameAs is not erroneous but - instead - the data in the knowledge base contain typos errors and needs cleaning; or she can understand that some rules considered are wrong, e.g. some properties have been defined as *functional*, but they are not. If the sameAs is conceptually wrong, the explanation framework can support her in the comprehension of the problems encountered and, on the basis of the dialogues performed, she could decide that wrong decisions have been made during the linkage and thus, she can propose a re-linking phase.

We have implemented a first prototype of the explanation dialogue that communicates with a Datalog$_{\pm}$ rule-based reasoner called *Graal* [2]. For the knowledge base, we considered facts from the CORA dataset [7] and sameAs computed using the SILK framework [1]. Due to space limitations, here we present a single example of our approach.

Facts (portion of $\mathscr{F}$)
$sameAs(r_3,r_2)$, $sameAs(r_2,r_4)$, $sameAs(a_1,a_2)$, $sameAs(r_1,r_2)$
$confName(r_1$, 'proceedings aaai-98')
$confName(r_3$, 'in proceedings aaai-98')
$confName(r_2$, 'in proceedings of aaai')
$confName(r_4$, 'in proc. aaai')
$isconfNameDiffLevenshtein$('proceedings aaai-98', 'in proceedings of aaai', 0.73)
$isconfNameDiffLevenshtein$('proceedings aaai-98', 'in proceedings aaai-98', 0.73)
$isconfNameDiffLevenshtein$('in proceedings aaai-98', 'in proc. aaai', 0.73)
$isconfNameDiffLevenshtein$('proceedings aaai-98', 'in proc. aaai', 0.41)
$ispageFromDiffJaccard(30,15,0)$
$published(a_1,r_1)$, $published(a_2,r_3)$
$pageFrom(a_2,15)$, $pageFrom(a_1,30)$
Rules (portion of $\mathscr{R}$)
$sameAs(x,y) \wedge published(x,w_1) \wedge published(y,w_2) \rightarrow sameAs(w_1,w_2)$
$sameAs(x,y) \wedge pageFrom(x,w_1) \wedge pageFrom(y,w_2) \rightarrow isEquiv(w_1,w_2)$
$sameAs(x,y) \wedge confName(x,w_1) \wedge confName(x,w_1) \rightarrow isEquiv(w_1,w_2)$
$sameAs(x,y) \wedge sameAs(y,z) \rightarrow sameAs(x,z)$
$sameAs(y,x) \rightarrow sameAs(x,y)$
$isDiff(y,x) \rightarrow isDiff(x,y)$
$isconfNameDiffLevenshtein(x,y,\sigma) \rightarrow isDiff(x,y)$
$ispageFromDiffJaccard(x,y,\sigma) \rightarrow isDiff(x,y)$
Negative Constraints (portion of $\mathscr{N}$ and implicitly derivable)
$isEquiv(x,y) \wedge isDiff(x,y) \rightarrow \bot$
[derivable negative constraints]
$sameAs(x,y) \wedge pageFrom(x,w_1) \wedge pageFrom(x,w_1) \wedge isDiff(w_1,w_2) \rightarrow \bot$
$sameAs(x,y) \wedge confName(x,w_1) \wedge confName(x,w_1) \wedge isDiff(w_1,w_2) \rightarrow \bot$

Table 1: A portion of the facts $\mathscr{F}$, rules $\mathscr{R}$ and negative constraints $\mathscr{N}$ used to build our knowledge base $\mathscr{K} = (\mathscr{F},\mathscr{R},\mathscr{N})$.

Example. Let us consider a query Q as $sameAs(r_1,r_2)$, where r_1,r_2 are URIs describing two resources in CORA which describe two 'conferences' with title ($confName$) 'proceedings aaai-98' and 'in proceedings of aaai', respectively. Using inconsistent tolerant semantics, the query Q is *not entailed* by the $\mathscr{K} = (\mathscr{F},\mathscr{R},\mathscr{N})$ partially described in Table 1: the two conferences are *not the same*. In Table 2 and Table 3 we present our explanation dialogue providing details on the reasons *why* Q is not entailed as showing the dialogue between the reasoner and the expert.

Future works. Different interesting research directions can be exploited in the near future. For example, it could be interesting to study suitable graphical representations and UI (interactive system) to be used in the explanation of the inconsistencies and the properties involved. Or to study how to model a contextual weight associated to each property used. Such weights could depend on different factors: the reliability of each property in the initial dataset, suggestions coming from the experts, and so on.

Acknowledgments. The authors acknowledge the support of ANR grants ASPIQ (ANR-12- BS02-0003), QUALINCA (ANR-12-0012) and DURDUR (ANR-13-ALID-0002). The work of the second author has been carried out in a part of the research delegation at INRA MISTEA Montpellier and INRA IATE CEPIA Axe 5 Montpellier.

References

1. *Silk - The Linked Data Integration Framework*. Available at http://silk-framework.com/.
2. J.-F. Baget, M. Leclere, M.-L. Mugnier, S. Rocher, and C. Sipieter. Graal: A toolkit for query answering with existential rules. In *To Appear in Proc of RuleML*. 2015.

1. `User`**: Why** r_1 **and** r_2 **are not the same?** • *formalism*: $Q = sameAs(r_1, r_2)$? • *explanation*: This is an **explanation request** made by the `User` (EXPLAIN).
2. `Reasoner`**: Because** r_1 **is the same as** r_4**.** • *formalism*: $\langle\{sameAs(r_1, r_2), sameAs(r_2, r_4)\}, sameAs(r_1, r_4)\rangle$. • *explanation*: `Reasoner` provides an **attempt** with an argument **against** Q (ATTEMPT).
3. `User`**: clarify?** • *explanation*: The `User` asks for a **clarification**, meaning the *rule* relying the support and the conclusion of the argument before. (CLARIFY)
4. `Reasoner`**:** r_1 **is the same of** r_2**,** r_2 **is the same of** r_4 ***and by transitivity*** r_1 **is the same of** r_4 • *formalism*: $\langle\{sameAs(r_1, r_2), sameAs(r_2, r_4)\}, \{sameAs(x, y) \wedge sameAs(y, z) \rightarrow sameAs(x, z)\}, \{sameAs(r_1, r_4)\}\rangle$. • *explanation*: The `Reasoner` provides the **clarification**. In this case, the rule of transitivity for *sameAs* (CLARIFICATION)
5. `User`**: Why is this a problem?** • *explanation*: the `User` wants to know more about the logic behind the attack (DEEPEN)
6. `Reasoner`**:** $confName_{r_1}$ **is 'proceedings aaai-98' (**a**) and** $confName_{r_4}$ **is 'in proc. aaai' (**b**).** a, b **have Levenshtein similarity of** 0.41**. It is impossible that** r_1 **and** r_4 **are the same thing and they have dissimilar conference names (**$confName$ **is *functional*).** • *formalism*: - $sameAs(r_1, r_4), confName(r_4, b), confName(r_1, a)$ - $isconfNameDiffLevenshtein(a, b, 0.41) \rightarrow isDiff(a, b)$ - $sameAs(x, y) \wedge confName(x, w_1) \wedge confName(x, w_1) \wedge isDiff(w_1, w_2) \rightarrow \bot$. • *explanation*: The `Reasoner` provides the facts, the rules and the negative constraints that triggers the attack. (DEEPENING)
7. `User`**: Understood.** • *explanation*: the `User` ends the dialogue (POSITIVE)

Table 2: A query failure explanation dialogue for a sameAs query involving the resources r_1 and r_2. For each dialogue we outline the formalism and the utterances involved.

7. `User`**:** $sameAs(a_1, a_2)$**,** a_1 **is published in** r_1**,** a_2 **is published in** r_3**,** $sameAs(r_3, r_2)$**, thus** r_1 **is the same as** r_2**. (***published* **is functional.)** • *formalism*: - $sameAs(a_1, a_2), published(a_1, r_1), published(a_2, r_3), sameAs(r_3, r_2)$ - $sameAs(a_1, a_2) \wedge published(a_1, r_1) \wedge published(a_2, r_3) \rightarrow sameAs(r_1, r_3)$ - $sameAs(r_1, r_3) \wedge sameAs(r_3, r_2) \rightarrow sameAs(r_1, r_2)$. • *explanation*: The `User` gives disacknowledges and presents an argument that supports her query Q (NEGATIVE).
8. `Reasoner`**: But** a_1 **has** $pageFrom$ **value** 30**,** a_2 **has** $pagefrom$ **value** 15 **and** 30 **and** 15 **are different** • *formalism*: - $sameAs(a_1, a_2), pageFrom(a_1, 30), pageFrom(a_2, 15)$ - $ispageFromDiffJaccard(30, 15, 0) \rightarrow isDiff(30, 15)$ - $sameAs(x, y) \wedge pageFrom(x, w_1) \wedge pageFrom(x, w_1) \wedge isDiff(w_1, w_2) \rightarrow \bot$. • *explanation*: `Reasoner` provides an **attempt** with an argument **against** the previous argument (ATTEMPT).
9. `User`**: Understood.** • *explanation*: the `User` ends the dialogue (POSITIVE)

Table 3: A new portion of the failure explanation dialogue for an invalid sameAs involving the resources r_1 and r_2. In this case, the user asks for further explanations by providing an argument against the reasoner conclusion (instead of step 7. in the previous table).

3. M. Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc of AAAI*, 2012.
4. C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Intern Journal Semantic Web Information Systems*, 5(3), 2009.
5. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan& Claypool, 1st edition, 2011.
6. D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. Savo. Inconsistency-tolerant semantics for description logics. In *Proc. Intern. Conf. on Web Reasoning Rule Systems*, 2010.
7. A. McCallum, editor. *Cora Research Paper Dataset*. Available at http://people.cs.umass.edu/ mccallum/data.html.

Towards Natural Language from Formal Concepts in Formal Concept Analysis

Tim Wray, Peter Eklund

IT University of Copenhagen
Denmark, petw@itu.dk

Abstract. In many applications with formal concept analysis it is desirable to have more fluid and meaningful natural language descriptions of formal concepts. This paper outlines the data structures and processes used to generate natural language statements from formal concepts.

Our framework CollectionWeb uses `Predicates` to add an additional semantic layer to attributes and `Parsers` to express these semantics in NL. A `Predicate` can be thought of as a qualifier that characterises the relationship between the `Attribute` and the `Object`. For example, a work `byArtist` *Consuelo Kanaga* which is `typeOf` *photograph* that `depicts:location` *New York* is represented as follows.

```
fromTimePeriod:20th century
artistNationality:American
depicts:person-subject:child
fromTimePeriod:1920s
depicts:person-subject:mother and child
depicts:emotional-subject:melancholy
depicts:person-subject:children
isTypeOf:photograph
depicts:location:New York
depicts:person-subject:mother
byArtist:Consuelo Kanaga
depicts:emotional-subject:poverty
```

This is similar to the `Subject-Predicate-Object` relationship found in RDF. `Predicates` exist independently of `Attributes`: an `Attribute` may be assigned to one `Predicate` but a `Predicate` may be assigned to multiple `Attributes`. For example, the following attributes:

```
fromTimePeriod:20th century
fromTimePeriod:1920s
fromTimePeriod:1930s
fromTimePeriod:1940s
```

If assigned as `Attributes` to a `FormalConcept`, the following results from the `this.getNaturalLanguageExpression()` method: “works from the 1920s to the 1940s”.

The `Predicate` describes the semantic relationship between an object and an attribute and contains simple NL processing rules and templates, whereas its `Parser` contains *procedural* information on how a certain class of attributes should be mined from its free text field and subsequently expressed in NL.

The parsers use the Illinois Part of Speech Tagger[1] [2] to recognise nouns and noun phrases, and the Stanford Named-Entity Recogniser[2] [1] to identify people, places and organisations. The other `Predicates` built into the *CollectionWeb* framework are as follows:

(1) The `isTypeOf` predicate uses the `ObjectNounParser` for extracting and displaying textual data that describes actual object types. For example, attribute values that represent category types and classes such as `Contemporary works` or `paintings` would use the `isTypeOf` predicate. The `ObjectNounParser` ensures that object nouns are represented in their correct form when expressed in singular or plural forms. When parsed from textual data for mining and database storage, it ensures that they are represented as collective nouns (e.g. `warfare objects` instead of just `warfare`) and that they are both expressed correctly in singular and plural form (e.g., an object with the text snippet "painting" will be expressed as `painting` if the formal concept contains one object or `paintings` if the formal concept contains more than one object.) As an example of how the `isTypeOf` predicate expresses attributes, the following attributes: `isTypeOf:photograph isTypeOf:drawing isTypeOf:painting`. If assigned as `Attributes` to a `FormalConcept`, will produce the following result for `this.getNaturalLanguageExpression()` method: "paintings, drawings and photographs"

(2) The `is` predicate uses the `ObjectDescriptionParser` to parse descriptive elements and non-iconographical concepts from textual data that describe a literal depiction of an object by extracting noun-phrases. It also implements a custom method for expressing those entities in a meaningful way. The `ObjectDescriptionParser` works particularly well if the artwork's title is used as a data source, although very short snippets describing the description of the work can be used as well. The parser itself does not distinguish the level of iconography expressed within the source text, and it assumes that all text fed into the parser describes the literal description of the work, rather than its iconography. As an example of how the `is` predicate expresses attributes, the following attributes: `is:cover is:box`. If assigned as `Attributes` to a `FormalConcept`, will produce the following result for `this.getNaturalLanguageExpression()` method: "objects that consist of boxes and covers"

(3) The `depicts` predicate uses the `ImageSubjectParser` to parse entities and iconographical concepts from short snippets of textual data that succinctly describe an image-based artwork by extracting noun-phrases and named-entities. The parser recognises and distinguishes the difference between iconographical concepts that represent objects, people or places in a work of art, and implements custom methods for expressing those entities in a meaningful way. The

[1] `http://cogcomp.cs.illinois.edu/page/download_view/POS`
[2] `http://nlp.stanford.edu/software/CRF-NER.shtml`

`ImageSubjectParser` works particularly well if the artwork's title is used as a data source, although very short snippets describing the iconography of the work can be used as well. The parser does not distinguish the level of iconography expressed within the source data, and it assumes that all text fed into the parser describes the main themes associated with the artwork, rather than a literal description of the object.

The *predicate map* can be used to highlight concepts that are more general or more specific to one another, where attributes that describe more specific concepts (such as the name of an artist) may be displayed or expressed in place of attributes that describe more broader concepts (such as the nationality of an artist). As an illustration of this example, a formal concept with the following attributes:

```
fromTimePeriod:20th century
artistNationality:American
fromTimePeriod:1920s
isTypeOf:photograph
byArtist:Consuelo Kanaga
```

Will generate the following natural language statement: "photographs by Consuelo Kanaga from the 1920s"

Note that the the attribute value of `artistNationality:American` is not displayed in the statement due to it being 'overridden' by the `byArtist:Consuelo Kanaga` attribute as shown in the predicate map in the code on the first page, and the `fromTimePeriod: 20th century` attribute is not shown due to the way the `TimePeriodParser` omits century attributes if more specific date attributes are present. In this way, the predicate map can be used to imply a hierarchy of attribute values that have different predicates but share a common semantic class. For example, given that `byArtist` attributes are more specific than `artistNationality` attributes, and that `designedIn` attributes are more specific than `associatedWithLocation` attributes, only the more specific `byArtist` and `designedIn` attributes would be displayed – even with the presence of `artistNationality` and `associatedWithLocation` attributes.

If, following the same rules and predicate map as above, the attributes `fromTimePeriod:1920s` and `byArtist:Consuelo Kanaga` were to be 'removed' from the formal concept – for example by means of navigating to a formal concept's upper neighbour – then the generated natural language statement would appear as follows: "photographs by American artists from the 20th century".

The expression of the broader terms – `American artists` and `20th century`, implies that the concept is broader. Hence, to a user, navigating between the two concepts would appear as follows:

photographs by Consuelo Kanaga from the 1920s ⇒ *more photographs by American artists from the 20th century*

CollectionWeb prepends the word `more` to the concept that the user is navigating to if an attribute with an `is` or `isTypeOf` predicate is present, or `other`

if those attributes are not present, as shown in the following example where the user would then navigate to another upper neighbour that does not have the `isTypeOf:photograph` attribute present:

photographs by Consuelo Kanaga from the 1920s ⇒ more photographs by American artists from the 20th century ⇒ other works by American artists from the 20th century

The new attributes introduced by the second formal concept `depicts: emotional-subject:melancholy` and `depicts:person:mother and child` are the only ones displayed when the user navigates to it, as shown in the example below:

photographs by Consuelo Kanaga that depict poverty ⇒ similar photographs that depict mothers in melancholy

Note that attributes with `is` or `isTypeOf` predicates are always displayed. In continuing with the running example, the user would then navigate to an upper neighbour of the latter formal concept, removing the attribute `depicts: emotional-subject:melancholy` so that its upper neighbour is displayed.

photographs by Consuelo Kanaga that depict poverty ⇒ similar photographs that depict mothers in melancholy ⇒ more works by Consuelo Kanaga that also depict mothers

1 Conclusion

Our case studies in museums gave rise to the requirement that we generate meaningful narrative descriptions of formal concepts from lists of attribute values. In this paper we have shown how an effective and simple ontology of predicates can be organised into a predicate map with accompanying parsers to generate NL descriptions of formal concepts. While enhancing usability of our case studies, the design goal of a more narrative like expression of formal concepts using natural language aided the key tasks of *identification* and *recognition* in Wille's key tasks of Conceptual Knowledge Processing [3]. In so doing, we adhere to the principles of Conceptual Knowledge Processing.

References

1. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363 —- 370, 2005.
2. D. Roth and D. Zelenko. Part of Speech Tagging Using a Network of Linear Separators. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 1136 —- 1142, 1998.
3. Rudolph Wille. Conceptual landscapes of knowledge: A pragmatic paradigm of knowledge processing. *Classification in the Information Age*, pages 344–356, 1999.

Author Index