On Using Semantically-Aware Rules for Efficient Online Communication

Zaenal Akbar, José María García, Ioan Toma, Dieter Fensel

Semantic Technology Institute, University of Innsbruck, Austria
firstname.lastname@sti2.at

Abstract. The ever growing number of communication channels not only enables a broader outreach for organizations, but also makes it more difficult for them to manage a very large number of channels and adapted content efficiently. Thus, finding the right channels to disseminate some content and adapting this content to specific channel requirements are real challenges for sharing information both efficiently and effectively. In this work, we present a rule-based system that addresses these challenges by decoupling the information to be shared from the actual channels where it is published. We propose semantic models to characterize and integrate various information sources and channels. A set of independent rules then interrelates these models, specifying the concrete publication workflow and content adaptation required. Furthermore, we evaluate our rule-based system using two different use cases, discussing the added value that the defined rules provide to this scenario and how they contribute to overcoming the identified challenges effectively.

Keywords: online communication, rule-based systems, knowledge modelling, social media

1 Introduction

In order to be able to disseminate the information about their products or services, each organization needs to reach the widest possible audience. During the era of Internet, the number and kind of dissemination channels have been increasing: websites, e-mails, and social media have become mainstream means of communication.

For the organizations, being present on several channels is not enough, since they also have to make sure that their content is suitable for each channel. In this case, information dissemination is not only about finding suitable channels, but also fitting the content to the available channels dynamically. These are the main challenges for effective and efficient information dissemination, and for online communication in general.

Our solution to overcoming these challenges is to decouple information from channels, defining separate models for each of them, and then interlinking them with an intermediary component [1]. Semantic technologies play four important roles in the solution [2]: semantic analysis, semantic channels as sharing data with reusable vocabularies, semantic content modeling, and semantic matchmaking.

In this paper, we discuss in detail the intermediary component to interlink content and channel, whose main objective is to align both components. Although this interlinking process is comprised of several elements, in this work we focus on the processing rules called *publication rules*. We first show the formalization of our solution, followed by rules construction accompanied by motivating examples. Furthermore, we discuss current implementation of this solution and two different use cases that validate our proposal.

The remainder of this paper is organized as follows: Section 2 describes our conceptual solution to overcoming the identified challenges of online communication. Section 3 introduces the publication rules as the main element of our online communication platform. Section 4 describes the technologies to implement our publication rules. Section 5 shows the application of our proposal to two different use cases. Finally, we discuss some related works in Section 6 and our conclusions and future work in Section 7.

2 Conceptual Approach

In this section, we describe our proposed conceptual solution to enable effective and efficient online communication. The proposed solution separates the content and channel to enable various dimensions of reuse in transactional communication [1]. This solution requires the development of information models, channel models and an intermediary component to align both models.

Our conceptual solution is shown in Fig. 1 where various information models and channel models were devised in order to represent the available information and targeted channels respectively. A component called *Weaver* corresponds to the intermediary component. Each component is described as follows:

Information Model An information model is an ontology that describes the information items that are used in typical acts of communication in a certain domain. As a formal, explicit specification of a shared conceptualisation [3], an ontology represents the concepts, the relations between concepts, and their constraints. In the information models, the relevant concepts for information dissemination are determined and shared among the content sources (i.e. documents, databases) which might have different data formats/representations.

Channel Model In online communication, a channel can be described as a means of exchanging information through the online space, which can be referred to (but not necessarily) with an Uniform Resource Identifier (URI) [4]. More than just as a place to spread or access information, a channel is also considered as a way to express or refer to the information. Each channel may have its own particularities including which types of information items can be read from or written to and access methods, among other particularities.

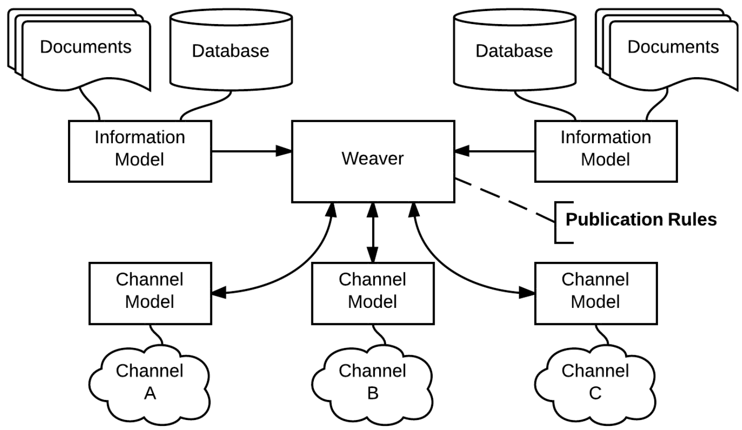

Fig. 1. Our conceptual solution for effective and efficient Online Communication

Weaver The Weaver is the component responsible for aligning the information and channel models. Formally, it has nine elements [1]: *a*) an information item, *b*) an editor, *c*) an editor interaction protocol, *d*) an information type, *e*) a processing rule, *f*) a channel, *g*) an executor, *h*) an executor interaction protocol. This paper is focused on the definition of processing rules called publication rules that govern how the information and channel models fit together.

Rules A rule is a form of representing knowledge specifying a certain conclusion whenever a certain premise is satisfied, represented as `IF Premise THEN Conclusion`. Generally, rules can be divided into three categories: deduction (derivation) rules, normative (integrity) rules, and reactive (active) rules [5]. Reactive rules are usually further divided into the form of Event-Condition-Action (ECA) rules and Condition-Action (CA) rules also known as production rules. We use production rules (in the form `IF Conditions DO Actions`) as the foundations to define our publication rules, where the `Actions` part will be executed whenever a change makes `Conditions` true.

3 Publication Rules

In this section, we describe the publication rules in detail, starting with the essential definitions, followed by the rule constructors, and finishing with a few examples of rule usage for complex online communication scenarios.

3.1 Definitions

Definition 1 (Information Item). *An information item I is the basic element of information in the domain of interest. Each element is identified by a name and the expected type of its value.*

The basic elements of information differ from one domain to the next. An example of these basic elements are `name` with the expected type *Text*, `date` with the expected type *Date*, `url` with the expected type *URL*, and so on. An element `name` might be divided further into `firstName` and `lastName`, depending on the modeled domain.

Definition 2 (Content). *A content C is described as a tuple of information items $I = (i_1, \cdots, i_n)$ where $|I| > 0$ is the number of items covered by C.*

The cardinality of the contents shows the richness of the information items represented, and may vary for each implementation. For example, information items (`title`, `description`, `location`) are used to describe a content `Event`.

Definition 3 (Content Transformator). *A content transformator T is an operator which transforms an input content C to produce a transformed content C^T.*

A content transformator operates on the information items of an input content, for example selecting a subset of the available items, shortening the value of an item, and so on.

Definition 4 (Channel). *A channel H is a place to publish contents C where each channel supports at least one content transformator T.*

Definition 5 (Transformation Specification). *A transformation specification S is a tuple of (H, T) where H is a channel and T is a content transformator that specifies that T is supported by H.*

Typically, an expert who is familiar with the channel specificities determines whether a content transformator is supported by a channel.

Definition 6 (Mapping). *A mapping M is a tuple of (C, H) where C is the content to be published and H is the targeted channel.*

The mapping is determined by experts who understand which content will be published to which channel including which content transformation is required. A content could be mapped to one or more channels and a channel could be mapped with one or more contents.

Definition 7 (Publication). *A publication P is a tuple of (C^T, H) where C^T is the transformed content of C and H is the selected publication channel.*

Based on the previously explained definitions, we define the publication rules as follows:

Definition 8 (Publication Rules). *A publication rule R for a content C to a channel H is a mapping of C and H and a content transformator T supported by the channel H to produce a publication P. Given a transformation specification $S(H, T)$, a publication rule can be represented as $\{M(C, H) \wedge T(C) \rightarrow P(C^T, H)\}$*

With this definition, a publication rule is interlinking the information model (content) and channel model trough a mapping and a content transformation. The interlinking intention is to fit a content to a particular channel or to find the proper channels for a content. Therefore, a publication rule serves as: *a*) a mapping between a content and a channel, and *b*) a transformation of the content according to the mapped channels' specificities.

The publication rules are also controllable through a workflow or scheduling specification. For this case, we introduce the following definitions:

Definition 9 (Publication Workflow). *A publication workflow is a coordinated publication where a publication P_1 will be performed only after a publication P_0 has been successfully executed. Given a transformation specification $S(H,T)$, a publication rule with a workflow can be represented as $\{M(C,H) \wedge T(C) \wedge P_0 \rightarrow P_1(C^T, H)\}$.*

A workflow is useful to specify the publication order for a certain channels, one after another. For example, when we need a reference to a specific channel, the workflow can be used to ensure that the content will be published in the right order.

3.2 Rule Construction

Based on the definitions previously discussed, we define three different types of actions that could be fired by the rules as follows:

1. `Mapping` – an action to align a content to a channel
2. `Transform` – an action to transform a content using a content transformation operator associated with a channel
3. `Publish` – an action to publish a content to a channel

Each action will assert a new fact and together with predefined facts they form a collection of facts to be used to identify if a specific condition is fulfilled. We define four different types of facts as follows:

1. `hasTransformation` – a predefined fact to specify if a channel has a content transformation operator
2. `hasMapping` – a fact, will be inserted by the action `Mapping` to specify if a content is mapped to a channel
3. `isTransformedBy` – a fact, will be inserted by the action `Transform` to specify if a content has been transformed using a specific content transformation operator
4. `hasPublished` – a fact, will be inserted by the action `Publish` to specify if a content has been published to a channel

Based on those defined actions and facts, we constitute the publication rules with one basic fact and three basic rules as follows:

1. `Mapping(C,H)`
   This is a fact to mapping a content and a channel.

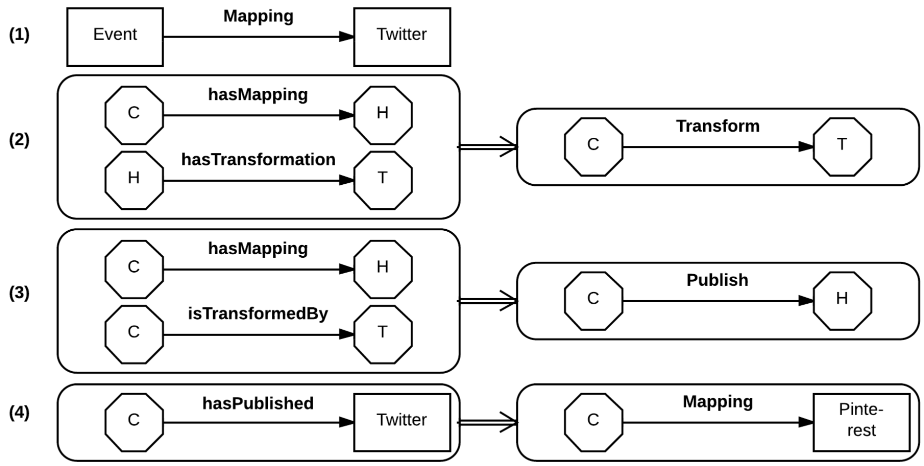

Fig. 2. An example of fact and rules construction

2. IF `hasMapping(C,H)` THEN `Transform(C)`
   This rule performs a content transformation operation using a transformation operator associated to a channel.
3. IF `hasMapping(C,H)` $\wedge$ `isTransformedBy(C,T)` THEN `Publish(C,H)`
   This rule performs a publishing action on a content to a channel whenever a mapping and a transformation have applied successfully.
4. IF `hasPublished(C,H1)` THEN `Mapping(C,H2)`
   This rule is to define a workflow, where a content will be published to the second channel whenever it has been published to the first channel.

Fig. 2 shows an example of the fact and publication rules construction using Grailog representation [6]. The octagon shapes represent variables (C for contents, H for channels, T for transformation operators), the rectangle shapes represent instances of content or channel or transformation operator, the single-line arrows represent the relationships, the double-lines arrows imply the rule consequences. The rounded boxes with solid-lines imply conjunctions. Given two transformation operators $t_1, t_2 \in T$, an information model `Event` $\in C$ contains information items `(name, description, location, url, startDate, endDate)` $\in I$, two channel models `Twitter, Pinterest` $\in H$ to represent Twitter (`https://twitter.com`) and Pinterest (`https://pinterest.com`) respectively. Two predefined content transformation specification are `(Twitter,` t_1`), (Pinterest,` t_2`)` $\in S$.

Fact and rules in Fig. 2 are constructed to publish information about an event to social media channels Twitter and Pinterest. Fact `(1)` is mapping an event to Twitter,rule `(2)` defines a content transformation action whenever a content is mapping to a channel using a transformation operator associated with the channel, rule `(3)` defines a publishing action whenever the content transformation has successfully applied, rule `(4)` defines a mapping between some content to

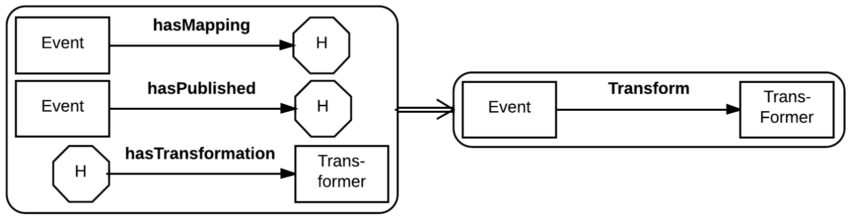

Fig. 3. A rule to define an implicit content transformation

channel Pinterest whenever the content has been already published to channel Twitter successfully. In this example, each instance of Event will be mapped to Twitter, then transformed by a transformation operator associated with Twitter and finally will be published to Twitter. After being successfully published, a second mapping is triggered to channel Pinterest, consequently repeating the content transformation and publish actions with this second mapping.

3.3 Rule Usage on Complex Online Communication

In addition to the typical publication rules (`Mapping - Transform - Publish`) as shown in previous section, our publication rules are also capable of handling various complex scenarios of online communication.

Implicit Content Transformation A typical publication rule contains a mapping and a content transformation. However, a publication rule can also be constructed by defining the content transformation implicitly. As shown in Fig. 3, a more flexible rule can be devised by defining a rule to match a transformation operator from previous publication activities. In this example, the rule does not specify the content transformation explicitly. The rule will be matched to a transformation (`Transformer`) from previously published similar content (`Event`) to a similar channel (variable H).

Diverse Content Transformation A channel may have more than one content transformation operator. Applying a different transformation to the same content will produce a different output. In the rule shown in Fig. 4, to publish a content C which has been published before (in this case to channel `Twitter`), the rule will be matched to a different transformation operator from that previously used. The box with dashed-lines indicates a negation, such that only the `Transformer`, which is not used by the previous publication, will be satisfied.

Rule Overwriting A rule can also be used to overwrite another rule permanently or temporarily (depending on certain conditions). As shown in Fig. 5, we can overwrite a mapping fact by using rules. Fact `(1)` defines a mapping between the content Event to channel Twitter, rule `(2)` will be to overwrite the

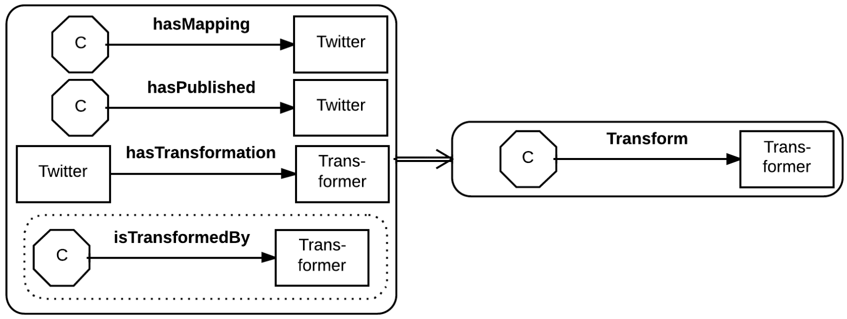

Fig. 4. A rule to use a different transformation implicitly

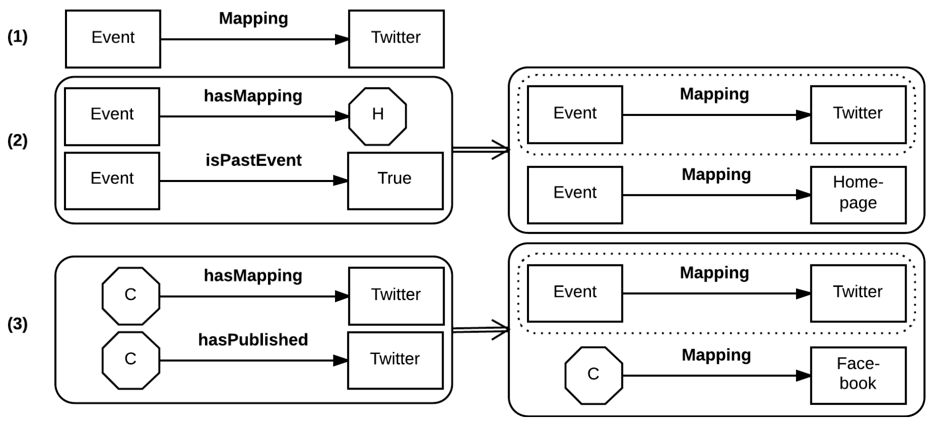

Fig. 5. A fact and two rules to overwrite the mapping

mapping whenever the event is identified as a past event, rule `(3)` will overwrite the mapping if the content has been published on Twitter. Each rule will retract the old mapping fact and insert a new one.

4 Implementation

In this section we explain the implementation of the publication rules in our online communication platform. Fig. 6 shows the platform which consists of various components that are grouped based on their functionality in the conceptual solution explained in Section 2.

Documents The online data sources hold the related information to be disseminated. There are two types of data sources currently supported:

1. Annotated sources. An additional information (metadata) can be attached to the existing piece of data with the intention to describe the data. More

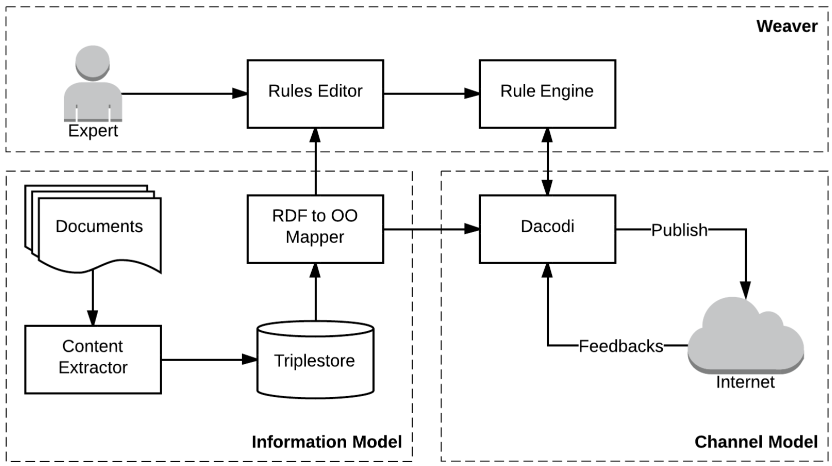

Fig. 6. The Online Communication platform architecture

annotated sources are available on the internet with a specific vocabulary to enable machines to interpret and use the data.

2. Un-annotated sources. The information is available without a common format/representation, existing in various database systems.

Content Extractor This component is responsible for obtaining the content from data sources and representing them into the common vocabularies. The Linked Open Vocabularies (`http://lov.okfn.org/dataset/lov/`) are used primarily to achieve a reusable and interoperable information model: *a*) Dublin Core, a specification of all metadata terms to support of resource description (`http://dublincore.org/`), *b*) Friend of a Friend, a vocabulary to describe people, the links between them, the things they create and do (`http://www.foaf-project.org/`), *c*) Good Relations, a vocabulary to describe e-commerce products and services (`http://purl.org/goodrelations/`), *d*) Schema.org, a collection of tags to markup a page in ways recognized by major search engines (`http://schema.org/`). We use Apache Any23 (`https://any23.apache.org/`) to extract content (in the format of triples/RDF) from the annotated sources, and for un-annotated sources a manual mapping is required to relate the database items into the desired vocabularies.

Triplestore A triplestore is a database repository to store triples/RDF statements extracted from data sources. We use OWLIM (`http://www.ontotext.com/owlim`) for a persistent storage and Apache Jena's (`https://jena.apache.org/`) In-Memory model for a non-persistent storage.

RDF to OO Mapper This component maps the RDF models (instances of classes including their properties) from the triplestore into object-oriented models to be used by other components. In our current implementation we are using RDFBeans (`http://rdfbeans.sourceforge.net/`).

Rule Engine and Editor To matching the facts against the defined rules, we use Drools (`http://drools.jboss.org/`) as rule engine, which implements and extends the Rete Algorithm [7] as its matching algorithm. To enable domain experts to maintain the rules, we use Drools Guvnor (`http://guvnor.jboss.org/`) as rule editor, it has rich web-based interface as well as a controllable access to the rules repository.

Dacodi Dacodi is the component responsible for distributing the content to the selected communication channels, as well as for collecting and analyzing feedback from those channels [8]. In our current implementation, Dacodi offers various functionalities such as: role management, publication, feedback collection, and front-end.

After the information models are defined and targeted dissemination channels are selected, the publication rules can be specified and constructed by an expert through the rules editor. Then, the stored rules will be consumed by the rule engine to construct a knowledge base to be employed by the Dacodi to make a publication decision. For each new content successfully extracted by the content extractor, a new fact is inserted into the knowledge base by the Dacodi, triggering the rule engine to match the fact against the existing rules. Whenever a match is found, the associated publication actions will be executed, followed by a scheduling monitoring to all feedbacks for each published content on each channel.

5 Use Cases

In order to validate our proposal, in this section, we show two use cases where the publication rules have been applied. First we discuss how the rules are implemented in PlanetData (`http://www.planet-data.eu`) and Tourismusverband Innsbruck (`http://www.innsbruck.info`), followed by a discussion on the contributions of the publication rules to both use cases.

PlanetData PlanetData is an European project funded by the EU Seventh Framework Programme between 2007 - 2013. For its dissemination activities, various information models have been defined, such as Project, Activity, Partners, WorkPackage, Event, Deliverable, FactSheet, Presentation, and others, as well as numerous dissemination channels: such as News, HomepageNews, RSS, PlanetData Mailing List, PlanetData Wiki, FacebookWall, and Semantics [9].

A subset of the publication rules for the PlanetData use case is shown in Fig. 7. The rules are intended to publish information about events to relevant channels. An object `Event` is used to represent all information items of event,

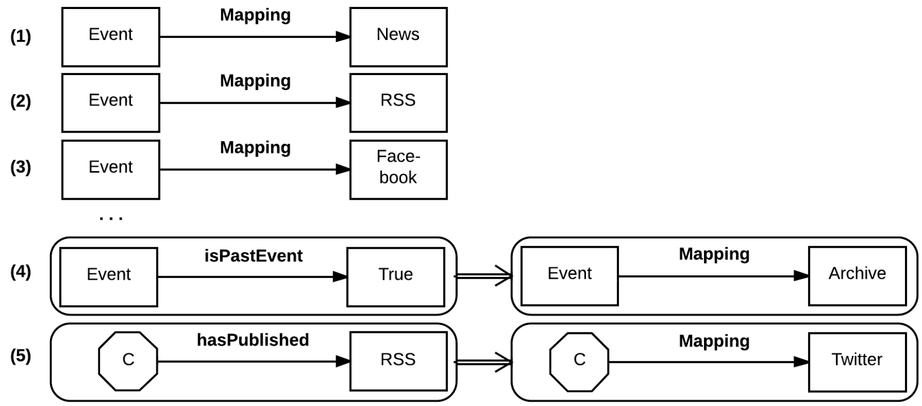

Fig. 7. A subset of the facts and publication rules for PlanetData

including an item `pastEvent` as the representation of its dueness. The objects `News`, `RSS`, `Facebook`, `Archive` and `Twitter` are used to represent channels: the news section of the project website, the website RSS output, the project Facebook account, the archive section of the website and the project Twitter account, respectively.

As shown in Fig. 7, there are three methods to perform a mapping between a content to its selected dissemination channels. First, by using explicit mapping through facts `(1-3)`; second by using a conditional mapping where a mapping will hold only if a certain condition is satisfied as shown in the rule `(4)` (the mapping between event to a channel Archive will hold only if the event has happened some time ago); third by using a publication workflow as shown in rule `(5)` (the mapping to the channel Twitter will hold only after the event has been published to the channel RSS). The rules for content transformation and publication are identical to the rules shown in Fig. 2.

Tourismusverband Innsbruck The relevant information for dissemination found in Tourismusverband Innsbruck (TVb)'s website have been categorized as Hotels, Food and Drink Establishment, Events, Trips, Place of Interest and News. Those categories are then modeled as concepts in TVb's ontology where the main concepts are Place, Event, Organization, Trip (Action), Creative Work, and Person [10].

As dissemination channels, TVb is using Facebook, Twitter, YouTube, and a Blog, and has been planning to use other channels such as Google+, Pinterest, tumblr [11]. A subset of the current publication rules for TVb is shown in Fig. 8. The objects `Hotel`, `Trip`, `Event`, `PoI`, `News` are used to represent the information concepts for hotel, trip, event, place of interest and news respectively. On the other hand, the objects `Facebook`, `YouTube`, and `Twitter` are used to repre-

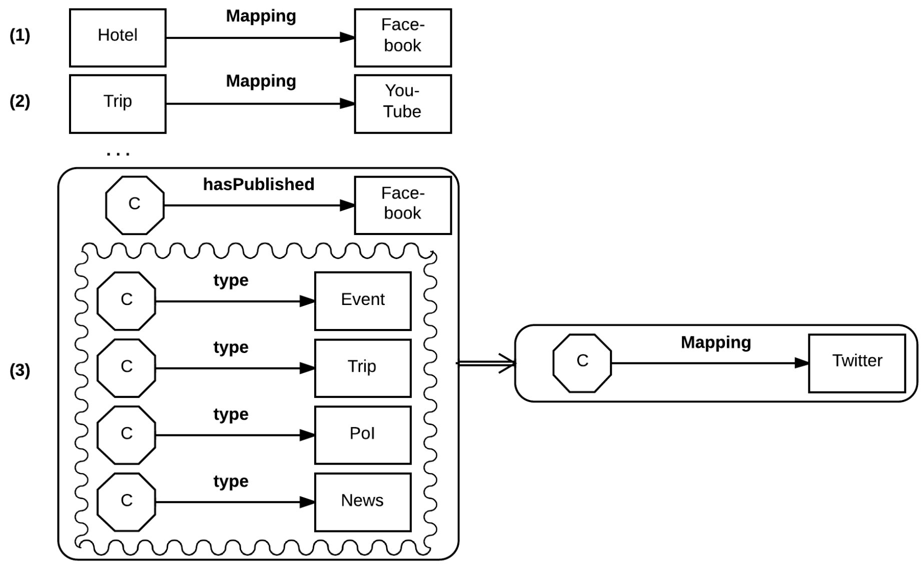

Fig. 8. A subset of the facts and publication rules for Tourismusverband Innsbruck

sent the TVb's Facebook account, TVb's YouTube account and TVb's Twitter account respectively.

Fig. 8 shows a subset of facts and rules implementation where a box with a solid-wavy-line implies a disjunction. There are two methods to do a mapping; first by using explicit mapping through facts `(1-2)`; second through a workflow definition as shown in rule `(3)` where a mapping to the channel Twitter for a content (event or trip or place of interest or news) will hold only after the content has been published to the channel Facebook. The rules for content transformation and publication are identical to the rules shown in Fig. 2.

Discussion

From both use cases, once the content types and targeted channels have been selected, the publication rules can be easily constructed. Given predefined content transformation operators for each channel, the mapping between content and channel can be defined explicitly and implicitly through rules. A publication workflow can be specified by adding referenced publication to the condition part of the relevant rules. In PlanetData, a content which needs to be published to Twitter must have a reference link to RSS, therefore the publication to RSS is included as criteria in condition of the Twitter's publication rule. TVb introduced a blog as a new input source where its contents are annotated with the Schema.org vocabulary. Since this vocabulary is supported by our information model, there is no need to modify the rule. A change is required only in the

Dacodi to add the blog's URL as an input source to be included in the next content extraction cycle.

In both use cases, the publication rules have shown numerous ways of interlinking the information models and channel models in the context of content dissemination. While the interlinking is determined by the experts, their representations might be specified explicitly through facts or implicitly through rules. These rich representations validate our solution to have various dimensions of reusing the content and channel in transactional communication.

6 Related Work

As the web is becoming more dynamic, reactive capability becomes more important in a variety of web applications [12, 13]. Reactive rules as a form to represent knowledge can be used to realize this reactivity. The reaction rules have been standarized to include reaction rules and rule-based event processing in Reaction RuleML [14, 15]. The publication rules presented in this paper are reactive rules derived from production (Condition-Action) rules which are capable of reacting to any changes in the input models. As our main usage is for online communication applications, the input models are ontologies (information models) which are domain specific. Our work is related to at least three relevant topics, described in the following paragraphs.

The first relevant topic is the channel to channel interlinking of online publishing. In this case, there are two prominent existing services: IFTTT (`http://ifttt.com`) and Zapier (`http://zapier.com`). Both services offer a solution to connect a channel to other channels including publishing content between channels through an automatic tool which is represented in a simple form `IF Trigger THEN Action`. Compared to these services, the main difference to our approach is in the creation of `Trigger`. In our approach, a knowledge-model is built, independent of any input channels. Instead of specifying a channel (i.e Facebook, Twitter) directly in Trigger, we use an information concept (i.e `Event`) instead, where the source of this concept could be a Facebook, a Twitter, or any other channels. We argue that using an information model as Trigger is highly suitable for integrating various input channels and offers high scalability. Whenever a new input channel is introduced where its content type is already included in the information model, then there is no need to create a new rule ("recipe" in IFTTT or "Zap" in Zapier).

The second relevant topic is to schema/ontology matching. The aim of the publication rules is to match two semantically represented models (information and channel). In this sense, the rules can be seen a matching mechanism. But in contrast to schema/ontology matching (such as [16, 17]), the matching is determined by the experts where each implementation has a different matching mechanism, such as different information models and/or targeted channels. Moreover, our rule framework enables the selection of content transformation operator dynamically by using rules (as shown at Subsection 3.3) as a contrary to a static selection by defining the operator explicitly in a fact. A publication workflow can be defined by adding a reference to a publication as criteria to the condition of the rule as shown in rule `(5)` in Fig. 7 and rule `(3)` in Fig. 8.

The third relevant topic refers to workflow control. Controlling a workflow with a rule-based system has been investigated in a few works such as in [18, 19]. From the three most commonly used workflow frameworks (control-flow graph, triggers, and temporal constraints), our workflow representation is an implementation of the triggers framework, where a workflow defines which publication activity needs to be executed first before the other activity.

7 Conclusions and Future Work

In this work we presented our rule-based solution for providing efficient and effective online communication, based on the separation of information and channel models. The publication rules introduced in this paper are reactive rules that are constructed to match the semantically represented domain specific information models to the channel models. This matching determines which information has to be disseminated to which channel, which content transformation and which publication workflow (if any) is required.

We have applied this type of rule to an online communication platform, which has been validated with two different use cases. In addition to reactiveness to changes in the information models, our rule-based solution also introduces new capabilities to dynamically adapt the content transformations and publication workflows if necessary. In conclusion, in order to achieve an effective and efficient online communication, publication rules devote a significant role to enable various dimensions of interlinking the information and channel models.

As future work, considering that in online communication the information is becoming more specific and targeted to a specific audience, we plan to extend the publication rules and associated channel models to reflect those specificities, such as enabling the definition of specific transformations for a certain channel. Furthermore we are going to incorporate more contextual dimensions into the publication rules, such as publication time, and location of target audience, among others.

Acknowledgements

We would like to thank all the members of the Online Communication working group (`http://oc.sti2.at`) for their valuable feedback and suggestions. This work was partly funded by the European Union's Seventh Framework Programme (FP7) under grant agreements no. 284860 (MSEE) and 257641 (PlanetData).

References

1. Fensel, D., Leiter, B., Thaler, S., Thalhammer, A., Toma, I.: Effective and efficient on-line communication. In: Database and Expert Systems Applications (DEXA), 23rd International Workshop on. (2012) 294–298

2. Toma, I., Fensel, D., Gagiu, A.E., Stavrakantonakis, I., Fensel, A., Leiter, B., Thalhammer, A., Larizgoitia, I., Garcia, J.M.: Enabling scalable multi-channel communication through semantic technologies. In: Web Intelligence, International Conference on. Volume 1. (2013) 591–596
3. Studer, R., Benjamins, V., Fensel, D.: Knowledge engineering: Principles and methods. Data & Knowledge Engineering **25**(1-2) (1998) 161 – 197
4. Fensel, A., Fensel, D., Leiter, B., Thalhammer, A.: Effective and efficient online communication: The channel model. In: International Conference on Data Technologies and Applications. (2012) 209–215
5. Boley, H., Kifer, M., Ptrnjan, P.L., Polleres, A.: Rule interchange on the web. In: Reasoning Web. Volume 4636 of LNCS., Springer (2007) 269–309
6. Boley, H.: Grailog 1.0: Graph-Logic Visualization of Ontologies and Rules. In: Proc. 7th International Web Rule Symposium: Research Based and Industry Focused (RuleML 2013). Volume 8035 of LNCS., Springer (July 2013) 52–67
7. Forgy, C.L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial Intelligence **19**(1) (1982) 17 – 37
8. Toma, I., Fensel, D., Oberhauser, A., Fuchs, C., Stanciu, C., Larizgoitia, I.: Sesa: A scalable multi-channel communication and booking solution for e-commerce in the tourism domain. In: e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on. (Sept 2013) 288–293
9. Fensel, D., Leiter, B., Brenner, C.: Planetdata on-line communication handbook. White paper, Semantic Technology Institute, University of Innsbruck (April 2012) `http://oc.sti2.at/results/white-papers/planetdata-line-communication-handbook`.
10. Akbar, Z., Lasierra, N., Tymaniuk, S.: Tourismusverband ontology. Technical report, Semantic Technology Institute, University of Innsbruck (January 2014) `http://oc.sti2.at/results/white-papers/tourismusverband-ontology`.
11. Akbar, Z., Garcia, J.M.: Tourismusverband publication rules. Technical report, Semantic Technology Institute, University of Innsbruck (March 2014) `http://oc.sti2.at/results/white-papers/tourismusverband-publication-rules`.
12. Bry, F., Eckert, M.: Twelve theses on reactive rules for the web. In: Current Trends in Database Technology EDBT 2006. Volume 4254 of LNCS., Springer (2006) 842–854
13. Berstel, B., Bonnard, P., Bry, F., Eckert, M., Ptrnjan, P.L.: Reactive rules on the web. In: Reasoning Web. Volume 4636 of LNCS., Springer (2007) 183–239
14. Paschke, A., Kozlenkov, A.: Rule-based event processing and reaction rules. In: Rule Interchange and Applications. Volume 5858 of LNCS., Springer (2009) 53–66
15. Paschke, A., Boley, H., Zhao, Z., Teymourian, K., Athan, T.: Reaction ruleml 1.0: Standardized semantic reaction rules. In: Rules on the Web: Research and Applications. Volume 7438 of LNCS., Springer (2012) 100–119
16. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal **10**(4) (December 2001) 334–350
17. Saake, G., Sattler, K.U., Conrad, S.: Rule-based schema matching for ontology-based mediators. Journal of Applied Logic **3**(2) (2005) 253 – 270
18. Davulcu, H., Kifer, M., Ramakrishnan, C.R., Ramakrishnan, I.V.: Logic based modeling and analysis of workflows. In: Proceedings of the 7th Symposium on Principles of Database Systems. (1998) 25–33
19. Mukherjee, S., Davulcu, H., Kifer, M., Senkul, P., Yang, G.: Logic-based approaches to workflow modeling and verification. In Chomicki, J., Meyden, R., Saake, G., eds.: Logics for Emerging Applications of Databases. Springer (2004) 167–202