Data Structuring for Launching Web Services Triggered by Media Content

Makoto Urakawa(✉) and Hiroshi Fujisawa

NHK (Japan Broadcasting Corporation), 1-10-11, Kinuta, Setagaya-ku, Tokyo, Japan
{urakawa.m-gi,fujisawa.h-ja}@nhk.or.jp

Abstract. There are some efforts to inspire the viewers to buy something or visit somewhere when they are viewing such objects or places on the TV programs or web sites. It is required to link some specific services, which can be run on smartphones or tablets, to content on TV or a web site. However, simple combination of content and possible services still requires some operations taken by the viewers such as typing search words in the applications or the pages of the services. Such required actions may decline their motivations to use the services. Therefore, the authors propose the data model that allows to seek matching of an entity within content with various services, and to generate service launch information dynamically changed by combination of an entity and a service. The data model is based on combination of ontology class structure and reasoning. In this paper, effectiveness of the data model is shown by the prototype applications which call various web services to inspire viewers for subsequent actions in accordance with media content.

Keywords: Ontology · Resource description framework · Knowledge base · Linked data · Semantic web rule language · Inference · Ifttt

1 Introduction

Broadcasters are working toward using the Internet to connect with viewers action from broadcasts—one example is displaying related keywords during broadcasts to urge viewers to search for information using the Internet. However, connecting TV programs with user action remains difficult, as television viewers must use their smartphones to type in search keywords. In terms of the web, although users who browse websites using their smartphones can use their web browser's share button to share the browsed information to social media networks (or they can send a link over an email), actions are limited to transmitting information. They are not able to purchase or do some actions directly when viewing web sites. Approaches are therefore in place to connect content with user action for TV programs and websites. However, some issues remain.

An increasing number of services based on the concept of If-This-Then-That (IFTTT)—such as ifttt[1] and zapier[2]—function as frameworks to link a wide range of

[1] https://ifttt.com/discover.
[2] https://zapier.com/.

Z. Wang et al. (Eds.): JIST 2017, LNCS 10675, pp. 19–34, 2017.
https://doi.org/10.1007/978-3-319-70682-5_2

web services and Internet of Things (IoT) devices. For example, ifttt recipes have been released that make it possible to send an email or perform some other action, such as a New York Times article serving as a trigger.[3] However, most of the available actions are related to just transmitting information. 224,590 recipes published at ifttt as of September 2015 is ranked higher by Twitter, e-mail, SMS, Facebook, Evernote as an action [1]. Most functions and most actions performed via IFTTT make it possible to transmit information (such as sending an email). It is likely because it is easy to specify services and applications for transmitting information. To connect content with user action that goes beyond transmitting information, it is necessary to gain an understanding of the types of services available and generate queries for launching these services. However, gaining an understanding of the services being developed every day is a difficult task.

In this paper, a data model based on semantic technology is proposed, which use classes that define entities (such as "Itsukushima Shrine") within broadcast program/web site and services (such as Google Maps) including native applications/web applications/web sites, to perform matching these and generate information to launch matched services inherited from property information of entities within content.

In addition to just starting a map application (as the "go" application) when (for example) "Itsukushima Shrine" is introduced at a TV program, the proposed data model can make it possible to seamlessly obtain and configure information required to search for "Itsukushima Shrine" in the map application from the TV program. Other user actions connecting content are also possible. One example is a user who is viewing a cooking recipe that includes "asparagus" on a website. The software can check the user's and place an order on an EC site if there is no asparagus in his/her refrigerator. The idea proposed in this paper will make it possible to create new services that go beyond merely transmitting information.

This paper consists of the following items. Section 2 presents some important issues. Section 3 introduces related research. and Sect. 4 presents a data model for resolving the issues presented in Sect. 2. Section 5 discusses expansion of services with prototype applications. Section 6 provides a summary of this paper. Note that smartphone applications and web services are both referred to as "web services" in this study.

2 Issues

The objective of this research is to propose the way of starting web services, which are operated by different industries or companies, with information of entities in media content (such as television programs or websites). This paper premises that content holders (such as broadcasters or website operators) are generally not able to comprehend what kinds of web services can be linked to their content, and service providers are not able to understand what kinds of content they can use for their services as well. In order to launch web services with information of matched entity, 3 data processing is needed. They are extracting entity information from content, matching entity with relevant web services, and generating service launch information inherited from entity.

[3] https://ifttt.com/nytimes.

It is possible for users to identify entities within content by selecting items on a device with an interactive screen. However, even if they find the entity, it would be difficult for them to search for a web service that could be launched. Even if entities and services could be matched automatically or if users could select a service that could be launched, the user would need to enter some parameters (such as search keywords) for launching the web service. Such a process does not enable a smooth connection with the user's action.

From the perspective of content holders, it is possible to extract entities from content based on its subtitle information (for television programs) or text information (for websites). However, configuring links to web services is difficult, as a content holder will need to know each provider's web services. Even if a content holder did have information on specific web services, it is not realistic for the content holder to configure information for launching web services, owing to the large amount of work involved in the process.

For example, the NHK World website about Nagasaki[4] provides information on "chanpon noodles" (a type of food) and "Dejima" (a sightseeing area). In this case, it would be necessary to insert links to search results for "chanpon noodles" on an EC site such as Amazon[5] to support the actions of users who might want to purchase this food item. It would likewise be necessary to configure links to search results for "chanpon noodles" on recipe sites[6] to support the actions of users who might want to try cooking this dish themselves. Another link would need to be configured for the "Dejima" entity for launching a service to enable users to find information about visiting the location.

As described in this section, users and content holders can identify entities, however the difficulty is in configuring information to match and launch services provided by other providers.

3 Related Work

IFTTT is said to be able to automate actions such as task automation by integrating with Internet service after securing usability [2]. On top of that, there are several researches to introduce Semantic Web technology into IFTTT, which links various Web services and IoT devices in consideration of situation or context. For example, based on relationships and dependencies that can be defined by ontology, a research that cooperates objects such as sensors in space [3] and A tool that allows the user to specify behaviors in the space has been carried out [4]. Evented web ontology (EWE) [5] has been developed so that vocabulary and rules can be shared for task automation such as IFTTT. In addition, research is also being conducted to achieve compatibility of processing performance and expression power on inference [6]. We also see the importance of the

[4] https://www3.nhk.or.jp/nhkworld/en/culture/mydestination/20170720.html.
[5] https://www.amazon.co.jp/s/ref=nb_sb_noss?__mk_ja_JP=%E3%82%AB%E3%82%BF%E3%82%AB%E3%83%8A&url=search-alias%3Daps&field-keywords=%E3%81%A1%E3%82%83%E3%82%93%E3%81%BD%E3%82%93&rh=i%3Aaps%2Ck%3A%E3%81%A1%E3%82%83%E3%82%93%E3%81%BD%E3%82%93.
[6] https://www3.nhk.or.jp/nhkworld/en/food/search/?qt=chanpon.

efficiency to describe the linking of services [7]. In this way, by incorporating elements of the semantic Web into IFTTT, more effective service cooperation is possible, and examples of application in concrete scenes such as office space are also shown. However, since the main objective of these researches are to connect various services, these are not considered of inheriting the information of a connected source service. As a research aimed at improving user service, there is research [8] to optimize the existing IFTTT service to the user's situation by utilizing ontology and reasoning. This research is a proposal on optimization considering user's location information and so on, and service collaboration by inheritance of contents information targeted by this paper is not done. In addition, there is research [9] linking information and channels to be offered on a rule basis, to enable content holders to efficiently provide information on-line. In this research, proposal is being made to extend the rule according to the type of content information. However, it is limited to static filtering of rules, and dynamic cooperation is not considered. In terms of adopting OWL into an automatic process, OWL-S is proposed as a framework focusing on describing service and process profile, not dynamic matching services and generating launch information [10].

Therefore, in this paper, we propose a method to match different web services and dynamically generate service launch information that is originated from entity information matched with them.

4 Data Model

It has been proposed that data collaboration can be realized by classes instead of entities [11, 12]. This paper focuses on the class structures to which entities belong (rather than the entities themselves) as well as these researches, reflecting the procedure used by humans to launch the web services based on given information. For example, when one wants to buy carrots, he/she would select services where sell foods because carrots are one of food. Likewise, one would select a map application for "Dejima" because it is a "location." Although "Itsukushima Shrine" could be defined as a "shrine," a "shrine" is a subordinate concept of a "location"; therefore, one would select a map application in this case. It should thus be possible to match content entities with web services by abstracting information as entity classes (rather than entities themselves), and then making use of their superordinate/subordinate concepts. In other words, dynamic matching of entity and service can be solved by implementing semantic technology such as ontology. The launch information varies by the type of web service. For example, users would search by the entity name (such as "chanpon noodles") on an EC site, on the other hand, they could use an address or latitude/longitude information (rather than the entity name) on a map website. This dynamic launching services requires data processing of defining what information is required for launching a specific web service, inheriting the required information from the entity, and then automatically generating queries. Especially inheritance of entity information solves dynamic generation of service launching information.

It is assumed that multiple service providers (such as content holders, EC site operators, and map application providers) can be involved to connect user action with TV program or website content. Data are therefore separated into three layers (entity layer, class layer, and matched data layer), and an additional data processing layer is placed between the class layer and matching layer. This allows multiple providers to work together. The following section describes the proposed data model after analysis of actual web services and smartphone applications.

4.1 Analysis of Actual Service Categories and Genres

We think that it is possible to use categories and genres already provided by each service as information for class defining in data model. Therefore, we did a survey of 60 services and applications, such as purchasing, map, social network services, e-mail and dial application, identified about 3,000 categories (as of July 2017). For example, Amazon (the major purchasing service) had 20 categories and 255 subcategories[7], one of major Japanese EC sites Rakuten had 35 categories and 463 subcategories[8], and one of major supermarkets Tokyu Store had 19 categories and 106 subcategories[9]. Comparing food-related categories for these services showed that Amazon's AmazonFresh category contained subcategories such as vegetables, fruit, and processed meat/meat processed products, whereas Rakuten's Food category contained subcategories such as vegetables/mushrooms, fruit, and meat/meat processed products. Tokyu Store had a much more segmented category composition, for instance, its vegetables category containing subcategories such as leafy vegetables, beans/stew, and vegetables/root crops, and its meat category containing subcategories such as processed meat and ham/sausage/grilled pork. Category information expressing services depend on service providers. In such a case, a large amount of rule description is necessary to combine category information. Therefore, dividing individual classes for each service provider and common external classes to connect individual classes, which leads the effectiveness of description for linking services and the availability of sharing knowledge for understanding other services.

We also investigated the information necessary for starting the above-mentioned 60 services. There are services (such as Google Maps) that offer two launch methods (keyword [http://maps.google.co.jp/maps?q=%keyword%] and latitude/longitude [http://maps.google.co.jp/maps?q=%latitude%,%longitude%]), most services are launched using one keyword. About 80% of 60 services and applications that were surveyed are able to be launched by inputting entity name within URL, which comes from two notable examples being Amazon (https://www.amazon.co.jp/s?field-keywords=%keyword%) and Rakuten (http://search.rakuten.co.jp/search/mall/%keyword%/).

4.2 Overview About Data Model

Figure 1 illustrates the data model for cooperation between content holders (of broadcast and website content) and service providers (of map application and EC website) based on the analysis of actual services. The data model consists of 4 data layers, entity layer,

7 https://www.amazon.co.jp/gp/site-directory/ref=nav_shopall_btn.
8 https://www.rakuten.co.jp/category/?l-id=top_normal_gmenu_d_list.
9 http://shop.tokyu-bell.jp/tokyu-store/app/common/index.

class layer, data process layer, matched data layer, to connect content and services flexibly by making it possible for entity, class and inference to be exchanged respectively.

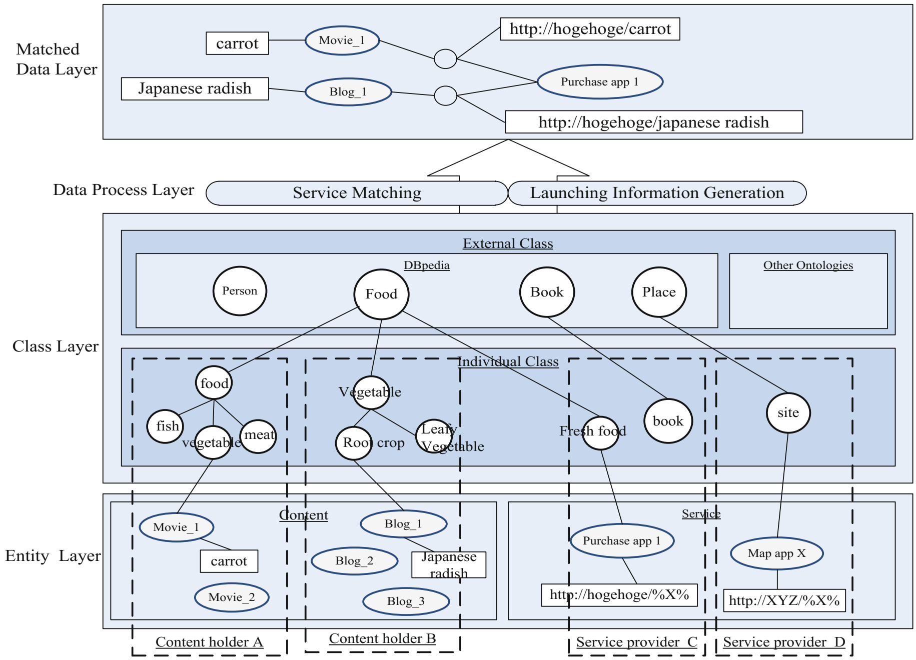

Fig. 1. Layered data model

Knowledge about offerings of content holders and service providers is stored at the entity layer. Entities are related to individual classes which can be defined by each service provider at the class layer. Individual classes also can be belonged to external classes. Knowledge about connection of classes are described at the class layer. Inference based on the class structure can be written at the data process layer, and knowledge about combination between content and services can be accessed via the matched data layer.

Figure 1 represents an example where a content holder A distributes a cooking video of "carrots," while a content holder B maintains a website on "Japanese radishes." A web service provider C offers a purchasing service, while a service provider D offers a map service. Each provider defines its own entity data and individual class structure. The scope defined by each company is its own entity data and its own class structure, and it is separated from the external class based on the idea from the actual service analysis mentioned in Sect. 4.1. The data model enables to match independently defined class structures to ones in an external ontology. In the example presented in Fig. 1, individual classes of content holder A, B, and service provider C, are related to the "food" class in the DBpedia ontology as an upper concept. Other ontologies such as GoodRelations[10] ontology can also be adopted as the external one.

The data processing layer is responsible for inferring connections between an entity and a service, and for generating service launch information based on the established connections. Section 4.3 will describe how referencing to external classes can optimize inference expressions. The matched data layer stores results generated through inference.

In Fig. 1, a cooking video about "carrots" relates to a purchasing application, and launching information is generated. Similarly, a purchasing application is linked with a website on "Japanese radishes," and service launch information is stored within the data model. Content holder A and B, which distribute the video and maintain the website, can put the links to launch these services within their internal services by retrieving a result from the matched data layer. This makes it possible to induce user action from content. It is also possible to add action-linked functionality to the existing website browsing functionality, by displaying matched layer results in the web browser's share button.

The next sections detail the data model that enables this architecture, and provide information on inference-based data processing.

4.3 Structuring Data

In this research, we associate entities within content with services in order for users to act on services when viewing content. That means content is a trigger to launch services. For this reason, we define separate structures for content and services. With the aim of dynamic and efficient data processing by using inference language such as SWRL(Semantic Web Rule Language), we adopt a data structure using an ontology that describes the conceptual structure of information, and then describe media content and web service information using Resource Description Framework (RDF) and Web Ontology Language (OWL).

4.3.1 Data Structure for Content

Figure 2 shows an ontology diagram for media content data and Fig. 3 shows an example of instances for broadcast content.

As shown in Fig. 2, media content, which is an instance of cn:Program or cn:webPage, is connected structurally with entities through the cn:hasEntity property relationship. Classes (for which it is assumed that content holders will independently create entities, such as cn:SightseeingSpot and cn:Book) are defined to serve as ranges of values for cn:hasEntity, and these classes are allocated in the individual class layer of the data model. The external classes in the class layer of the data model are associated with individual classes by rdfs:subClassOf relation.

Class definitions for entities are an important part of this data model. In Figs. 2 and 3, entities are associated with TV program scenes for launching services that are relevant

[10] https://www.w3.org/wiki/GoodRelations.

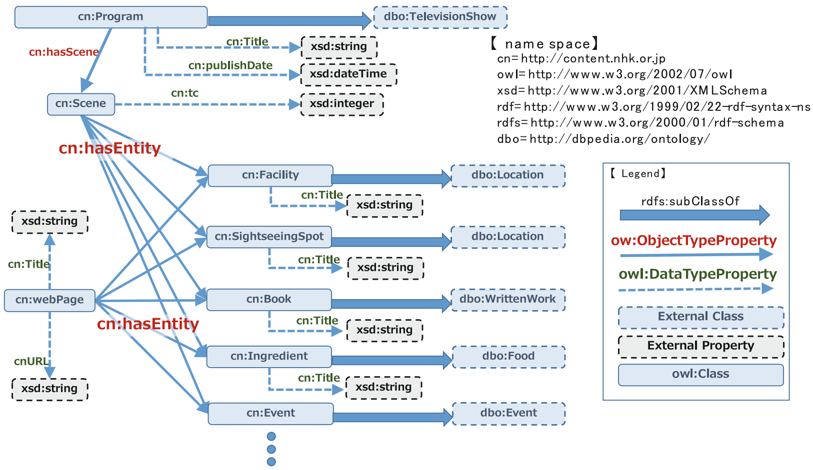

Fig. 2. Content ontology

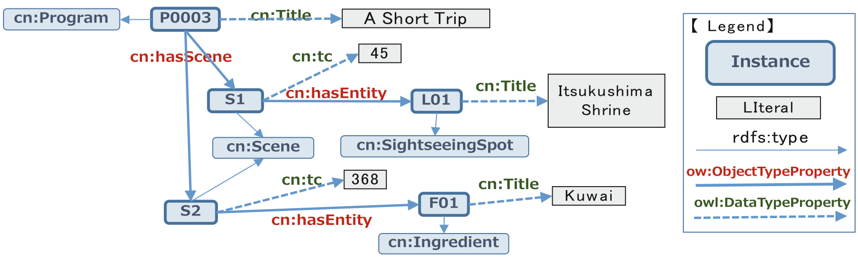

Fig. 3. Example of content data instances

to each scene. The "cn" namespace (http://content.nhk.or.jp) shown in Figs. 2 and 3 was created temporarily for this research.

The Protégé ontology editor was used to format data according to the RDF, based on the ontology in Fig. 2. Figure 3 shows an example of these instances. The example in Fig. 3 shows how instances are configured for a television program called "A Short Trip." The "Itsukushima Shrine" instance in the cn:SightseeingSpot class is set for S1, which is a scene that occurs at 45 s of the program. Next, the "Kuwai" (an edible variant of a three-leaf arrowhead) instance in the cn:Ingredient class is set for the 368 s mark. Entities and relevant classes are described using the RDF as content entities, and entities are stored in the entity layer as content knowledge, and the classes are stored in the class layer as category knowledge.

4.3.2 Data Structure for Service Description

Web service information is also structured for linking content with web services. More specifically, this describes actions and the targets of actions—such as a map application if the user wants to "go" to a "location," or an online supermarket application if the user wants to "purchase" a "food." This description of action target matches content with web services based on category information (such as “food”), making it possible to launch web services linked with content.

However, users still need to enter entity-related information manually on services if content entities are merely associated with services—content will not be seamlessly connected with user action. To obtain parameters required for launching web services, a framework is needed to obtain information from entities and to set this information as parameters. Descriptions on the service must therefore separately structure information for detecting services and information for launching services.

Figure 4 shows an ontology diagram for web services and Fig. 5 shows an example of web service instances. The “app” namespace (http://app.nhk.or.jp) shown in Figs. 4 and 5 was created temporarily for this research.

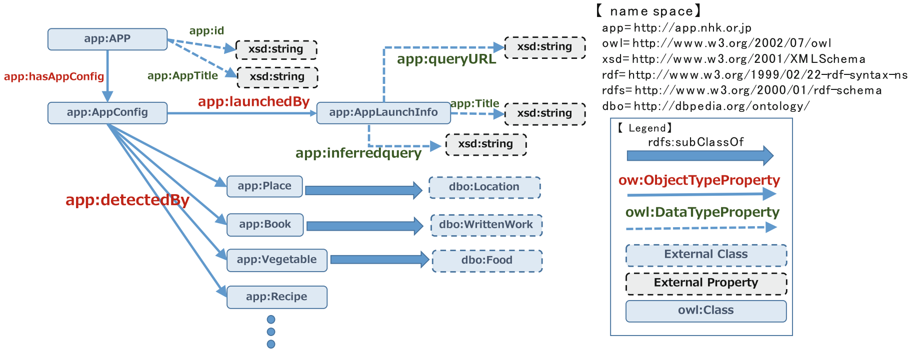

Fig. 4. Service description ontology

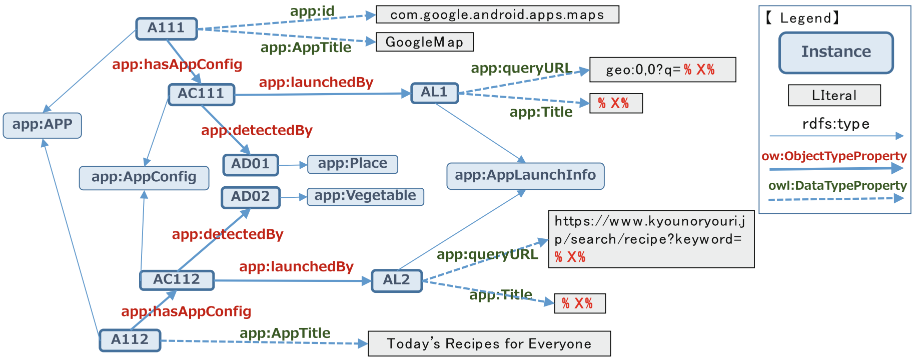

Fig. 5. Example of service description instances

The service description ontology makes it possible to configure the class information in order to be matched with content, by using the app:detectedBy property. This class information represents service category offered by service providers. Furthermore, app:queryURL can be retained in instances associated with the app:launchedBy property for launching web services. Data are structured so that the results inferred from app:queryURL are retained in the app:inferredquery property, allowing them to be changed dynamically according to media content. Note that provisional range (rdfs:range) classes are defined for app:detectedBy (in the service description ontology) and for cn:hasEntity (in the content data ontology), under the assumption that these range classes will be created independently by content holders and service providers. In addition, to make the descriptions of inference rule more efficient, classes to which the content entities belong and class information used to detect application information use rdfs:subClassOf to reference external classes. For this reason, DBpedia ontology classes are referenced in this figure. As shown by the data model, it is possible to reference other external ontologies.

In Fig. 5, the "Google Maps" service (detected by the app:Place class instance) and the web site for recipes "Today's Recipe for Everyone" (detected by the app:Vegetable class instance) are set as instances in the service description ontology. "geo:0,0?q = {%X %}" is used as a service launch information for launching "Google Maps" (%X% is a parameter inherited from the other resource). This "Title" property for a parameter to launch services is obtained from the "Title" property found in the matched content, allowing services to be launched according to the media content. This property information is used by the inference process described later in this study.

4.4 Inference-Based Matching/Query Generation Processing

This section describes a method for connecting media content with web service, and for inheriting parameters required for launching web services from linked entities. Protégé is used here as the inference tool, owing to its ability to make inferences via RDF schemas and OWL and to generate data using Semantic Web Rule Language (SWRL). Protégé version 5.2.0 and Pallet (inference engine) version 2.2.0 were used in this study.

In this section, matching a place-related scene of a TV program and a map application is introduced as an example.

4.4.1 Processing Matches for Content and Services

For the example instances shown in Fig. 3, linking the broadcast content with web services indicates associating S1 (a cn:Scene class instance) with A111 (an app:APP class instance). In this study, this relationship is represented by cn:hasRelation, which has relationships with cn:Scene (over rdfs:domain) and app:APP (over rdfs:range). cn:hasRelation here is generated through SWRL inference.

The example in Fig. 6 shows a rule for creating a triple from instances for app:Place (a class definition on the service information side) and cn:SightseeingSpot (a class definition on the media content side) via the cn:hasRelation relationship. In other words, various rules are required for combining many kinds of classes, and it is necessary to understand all the class definitions in advance. Classes from an external ontology (such as DBpedia) linked via rdfs:SubClassOf can be used for changing the inference rule from Fig. 6, as shown in Fig. 7. Figure 7 shows that they can be standardized in class definitions belonging to the external ontology. Without referring to an external ontology, many inference rules are required to be generated by all the combination of classes, such as app:Place and cn:SightseeingSpot, app:Location and cn:SightseeingSpot, app:Area and cn:SightseeingSpot, app:Place and cn:Address and so on.

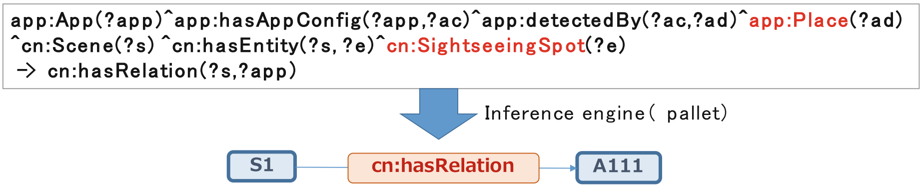

Fig. 6. SWRL rule and generated triple

```
app:App(?app)^app:hasAppConfig(?app,?ac)^app:detectedBy(?ac,?ad)^dbo:Location(?ad)
^cn:Scene(?s) ^cn:hasEntity(?s, ?e)^ dbo:Location(?e)
-> cn:hasRelation(?s,?app)
```

Fig. 7. Updated SWRL rule

4.4.2 Processing Query Generation for Launching Services

.Information used for launching web services linked with media content is also generated by inference. More specifically, service launch information is generated as the app:inferredQuery property in the app:LaunchInfo class instance. For example, “geo:0,0?q = Itsukushima Shrine” is generated for launching Google Maps during a scene introducing Itsukushima Shrine in the TV program. Figure 8 shows the SWRL inference rule, the generated triple, and the results screen on Protégé. SWRL contains built-in character string handling (as a subset). As shown in Fig. 8, it is possible to use swrlb:replace to replace parameter variables with media content.

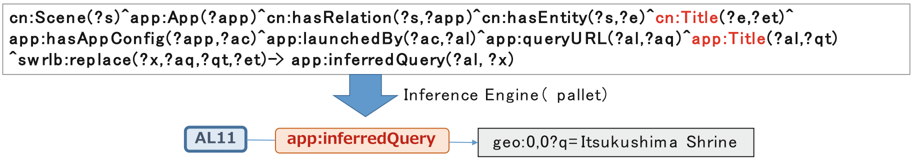

Fig. 8. SWRL rule and generated triple

Figure 9 shows an example of matching content and services described above, as well as the instances that are generated when inference processing is performed for service launch information. As shown in the figure, the “Itsukushima Shrine” entity is linked with “Google Maps” (used to search for directions), rather than with “Today’s Recipe for Everyone” (used to search for recipes). The figure also shows that the information for launching Google Maps is generated dynamically as a app:inferredQuery property.

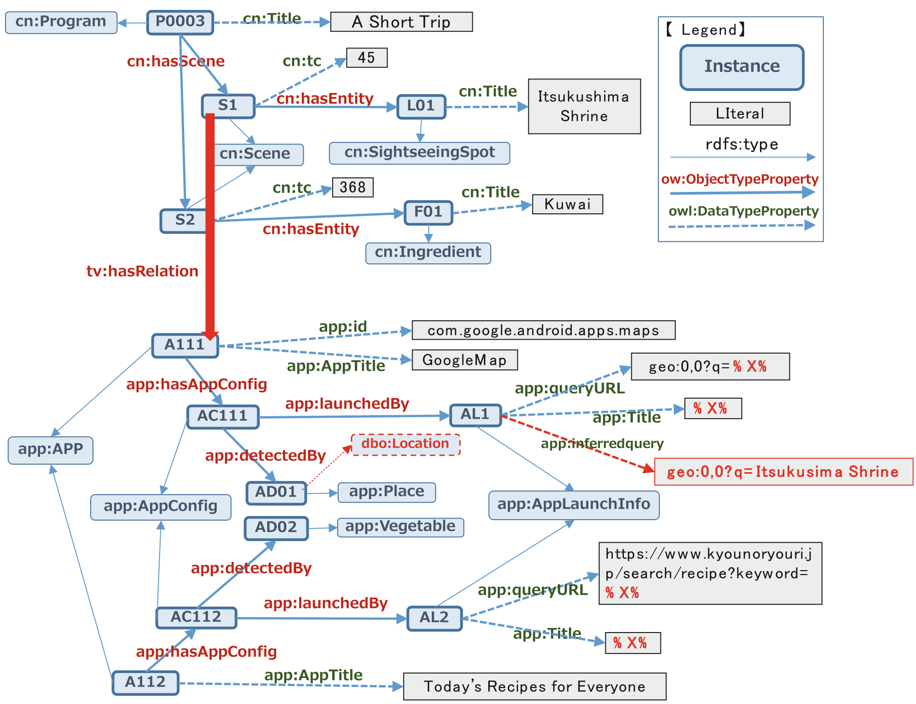

Fig. 9. Generated instances

5 Discussion

It is demonstrated that the proposed data model can expand existing services by adding user action-linked information to existing IFTTT services, and by developing action-linked services linked with content.

5.1 Service Linking When Watching Broadcast Content

To indicate the feasibility of the data model, we prototyped TV-linked application where they can start services triggered by entity shown on TV. This prototype used Fuseki as a data store on the matching layer, making it possible to obtain the data of content and matched web services using SPARQL queries. Figure 10 shows the system architecture. Hybridcast [13] and HbbTV [14] have been launched for the purpose of connecting TV and smartphones/tablets. In this study, to simulate a broadcast, a video playback client was used to play a video of a TV program stored on a web server. The TV-linked application obtains the playback position for identifying the scene being broadcast.

Figure 11 presents a screen from the TV-linked application. On this screen, various kinds of we services are shown by each related action during a scene in which asparagus is introduced. Since an asparagus is an instance of the dbo: Food class, web services and applications related to food are displayed as candidates. Note that action data (such as

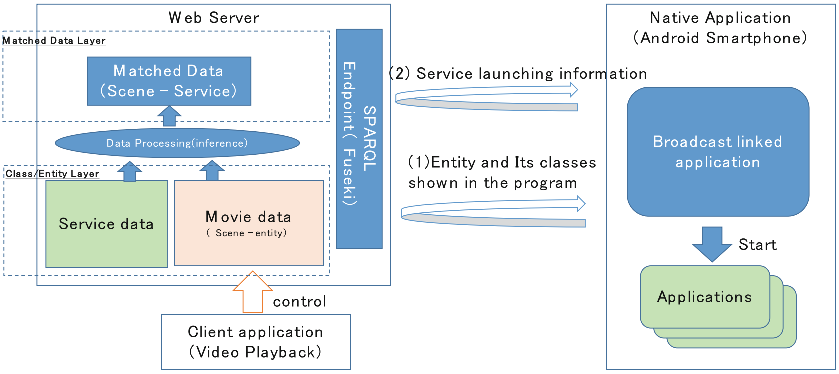

Fig. 10. System architecture of the prototype application

"purchase" or "search for recipes") has been converted to the RDF so that applications to "purchase" "food" or "search for recipes" can be displayed in a group. When the user presses the application's icon on the screen, "Today's Recipe for Everyone" is launched with "asparagus" set as the parameter[11], to allow the user to search for recipes without inputting the keyword "asparagus".

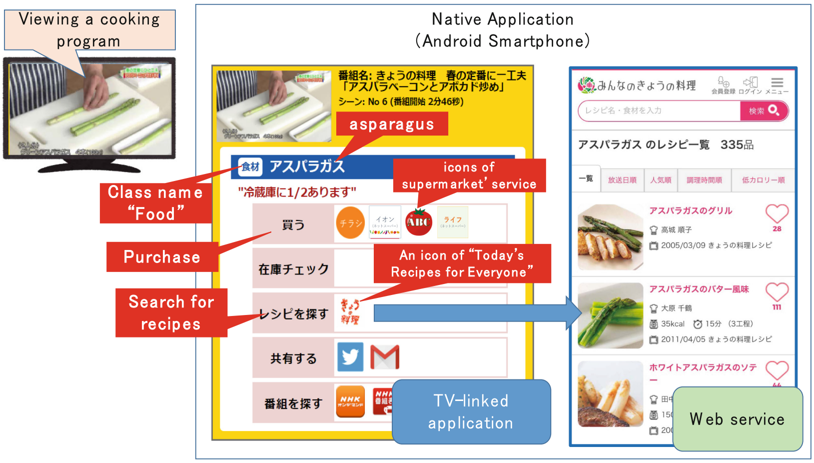

Fig. 11. Prototype application screen

5.2 Added Service Linkage to Existing IFTTT Service

This section presents an example of expanding NHK News RSS distribution, where the existing IFTTT service is used for creating a recipe forwarded over an e-mail.

[11] https://www.kyounoryouri.jp/search/recipe?keyword=%E3%82%A2%E3%82%B9%E3%83%91%E3%83%A9%E3%82%AC%E3%82%B9.

Figure 12 shows an example situation in which an e-mail is received according to an FTTT recipe, when NHK News reports via RSS distribution that Kowsi pears have been shipped from Takasaki, Gunma. As the RSS is originally written in Japanese, RSS itself and e-mail received via both IFTTT and this model are shown in Japanese. Under

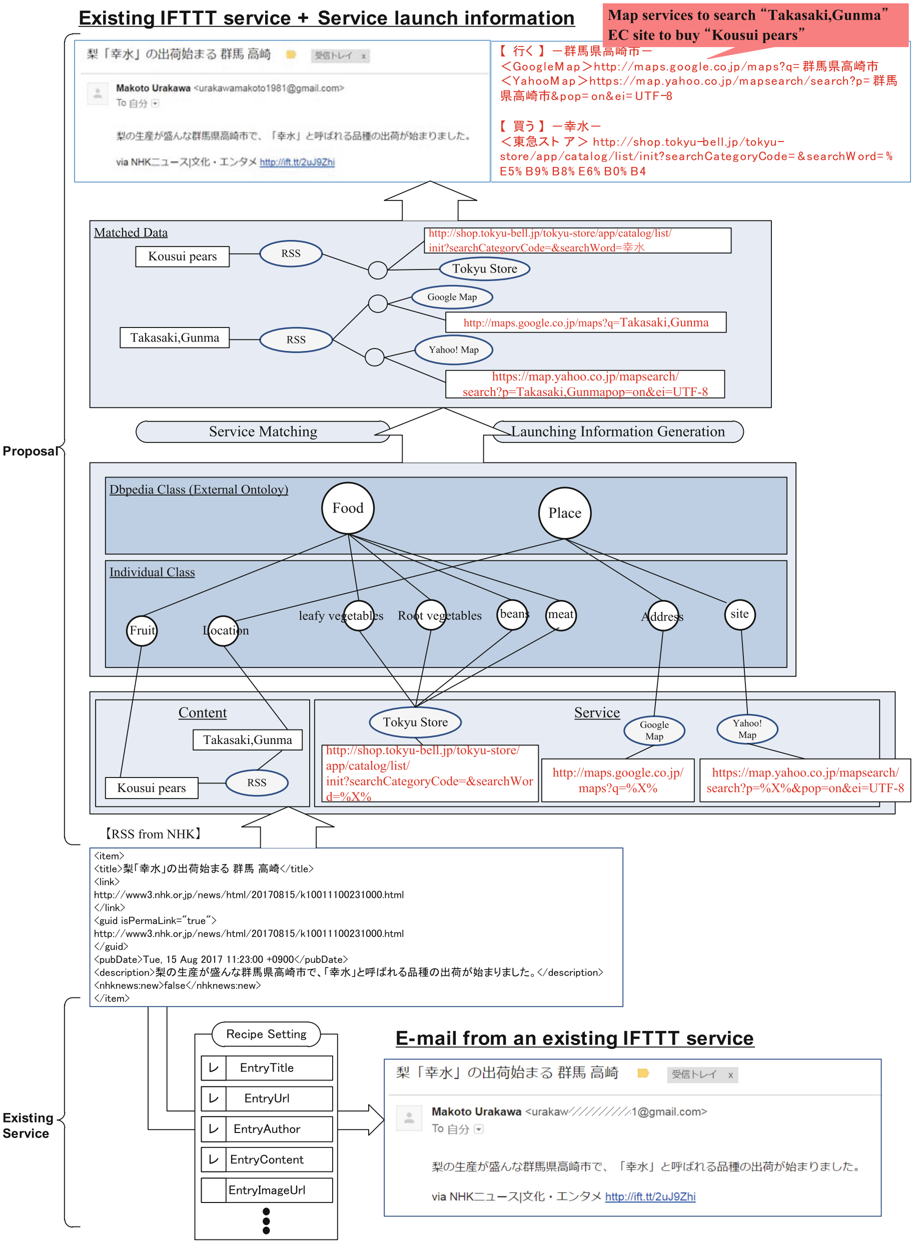

Fig. 12. Expansion of IFTTT

the existing IFTTT service where users can select the information to be transmitted by an e-mail, content distributed over RSS can only be transmitted over an e-mail with information configured in the recipe. On the one hand, the data model used in this study matches a map application (such as Google Maps) with "Takasaki, Gunma" in the "Place" class, and then generates launch information as a link. It also appends launch information for an online supermarket (such as Tokyu Store) for "Kosui pear" in the "Food" class. As shown above, the proposed data model is capable of going beyond mere information transmission to expand action-linked functionality connected to user action.

6 Summary

In recent years, there have been tremendous efforts to link broadcasts with the Internet and functions to enable actions (such as sharing) when browsing website content. However, users need to enter information manually on their devices such as smartphones or tablets. Although services (such as IFTTT) that use conditional expressions to connect various services have also matured, these services are limited to merely transmitting information. It is needed to match entities within content with services and to generate service launching information based on a related entity, for inducing viewers' action like buying or visiting beyond just information transmitting. Therefore, this study proposed a data model utilizing class structure and inference based on the semantic technology after analyzing 60 major web services and applications. Using this data model enables content holders to link other services even with keeping their own description for services they offer. On the other hand, from content viewers' perspective, they can start relevant web services on their devise like a smartphone without inputting a keyword related to a TV program or web site. That helps viewers to connect their behavior seamlessly.

The use case presented in this study also showed the possibility of expanding IFTTT services and expanding TV-linked user actions by implementing the data model. In future, a data model and architecture capable of deploying services closer to users—such as a framework that gathers user information to automatically select web services with which the user has contracted or that the user frequently uses—will be investigated. In addition, nonmonotonic reasoning can be used for more precise matching. For improvement of media services, we plan to implement the data model based on semantic technology into the existing Hybridcast services.

References

1. Ur, B., Pak Yong Ho, M., Brawner, S., Lee, J., Mennicken, S., Picard, N., Schulze, D. and Littman, M.L.: Trigger-action programming in the wild: an analysis of 200,000 ifttt recipes. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 3227–3231. ACM (2016)
2. Coronado, M., Iglesias, C.A.: Task automation services: automation for the masses. IEEE Internet Comput. **20**(1), 52–58 (2016). https://doi.org/10.1109/MIC.2015.73

3. Gu, T., Wang, X.H., Pung, H.K., Zhang, D.Q.: An ontology-based context model in intelligent environments. In: Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, vol. 2004, 270–275 (2004)
4. Rodríguez, N.D., Lilius, J., Cuéllar, M.P., Calvo-Flores, M.D.: Extending semantic web tools for improving smart spaces interoperability and usability. In: Omatu, S., Neves, J., Rodriguez, J.M.C., Paz Santana, J.F., Gonzalez, S.R. (eds.) Distributed Computing and Artificial Intelligence. AISC, vol. 217, pp. 45–52. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-00551-5_6
5. Coronado, M., Iglesias, C.A., Serrano, E.: EWE ontology specification (2015)
6. De Meester, B., Arndt, D., Bonte, P., Bhatti, J., Dereuddre, W., Verborgh, R., Ongenae, F., De Turck, F., Mannens, E., Van de Walle, R.: Event-driven rule-based reasoning using EYE. In: SSN-TC/OrdRing@ ISWC, pp. 75–86 (2015)
7. Muñoz, S., Llamas, A.F., Coronado, M., Iglesias, C.A.: Smart office automation based on semantic event-driven rules. In: Intelligent Environments Workshops (2016)
8. Ranasinghe, Y.S., Walpola, M.J.: Integrating Context-Awareness with reminder tools. In: 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE (2016). https://doi.org/10.1109/ICTER.2016.7829921
9. Akbar, Z., García, J.M., Toma, I., Fensel, D.: On using semantically-aware rules for efficient online communication. In: Bikakis, A., Fodor, P., Roman, D. (eds.) RuleML 2014. LNCS, vol. 8620, pp. 37–51. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09870-8_3
10. OWL-S: Semantic Markup for Web Services. https://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/
11. Kushniretska, I.: Semi-structured data dynamic integration mashup system. In: 2016 XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT). IEEE (2016)
12. Yamaguchi, A., Kozaki, K., Lenz, K., Yamamoto, Y., Masuya, H., Kobayashi, N.: Data Acquisition by traversing class-class relationships over the linked open data. In: International Semantic Web Conference (Posters & Demos) (2016)
13. Ohmata, H., Takechi, M., Mitsuya, S.: Hybridcast: a new media experience by integration of broadcasting and broadband. In: 2013 Proceedings of ITU Kaleidoscope: Building Sustainable Communities (K-2013). IEEE (2013)
14. Malhotra, R.: Hybrid broadcast broadband TV: the way forward for connected TVs. IEEE Consum. Electron. Mag. **2**(3), 10–16 (2013). https://doi.org/10.1109/MCE.2013.2251760

http://www.springer.com/978-3-319-70681-8

Semantic Technology
7th Joint International Conference, JIST 2017, Gold Coast, QLD, Australia, November 10-12, 2017, Proceedings
Wang, Z.; Turhan, A.-Y.; Wang, K.; Zhang, X. (Eds.)
2017, XIII, 352 p. 97 illus., Softcover
ISBN: 978-3-319-70681-8