Generating Service Value Webs by Hierarchical Configuration: An IPR case

Jaap Gordijn
VU University Amsterdam
De Boelelaan 1081
1081 HV Amsterdam, The Netherlands
gordijn@cs.vu.nl

Pieter De Leenheer
VU University Amsterdam
De Boelelaan 1081
1081 HV, Amsterdam, The Netherlands
pieterdeleenheer@cs.vu.nl

Ivan Razo-Zapata
VU University Amsterdam
De Boelelaan 1081
1081 HV Amsterdam, The Netherlands
izapata@few.vu.nl

Abstract

Enterprises increasingly jointly participate in service value webs. They do so because enterprises can then offer services they could not offer on their own. In earlier work, we have developed the e^{3}value *methodology to conceptually model such networks, as a design task. This paper proposes an approach to semi-automatically generate such service value networks based on service profiles and a stated consumer need. To this end, we apply skeleton design and hierarchical configuration techniques. The configuration process is illustrated by a case study in the field of intellectual property right clearing.*

1 Introduction

Enterprises increasingly participate in *networked value constellations* [19] or *value webs*. A networked value constellation is a network of enterprises, jointly offering something of value to its environment. Consider for instance the music industry, our case study domain. If a person listens to music on the radio, apart from the radio station itself, Intellectual Property Rights (IPR) societies will be needed to collect money for artists, songwriters, producers, and other IPR owners. All these actors are part of the same value web, needed to listen to a music track on the radio.

The IPR case study is an example of a *service value* web; Clearing rights and repartitioning the collected money over the right owners are *commercial services*, which are of value to the rights owners and the radio station. Our work contrasts with web services (see e.g. SOAP [9, 10], UDDI [1], WSDL [5], and WSMO [6]) ; whereas web services provide a platform to solve interoperability and orchestration between software components, our services should be understood as real life *commercial* services. Commercial services are often intangible [16, 12], have a processual nature [12, 8], and produce valuable outcomes [12, 18].

In earlier work [7] we have proposed the *e^{3}value* methodology to design and evaluate service value webs. In *e^{3}value* , we model the actors part of the service value web, as well as what they exchange of economic value (e.g. service outcomes) and what they require in return for that (e.g. money). So, the *e^{3}value* methodology helps in building a conceptual model for the value web at hand, and analysing the value web for its economic sustainability. As such, *e^{3}value* is a consultancy instrument that helps to design value webs and to articulate the business case [1].

Our long term goal however, is to *semi-automatically generate* service value webs, based on a given consumer need. For our IPR case study this means, that given the need to play a particular music track, a service value web of relevant IPR societies and IPR owners will be generated automatically. Such a service value web can then be the starting point to also generate the required IT-support (e.g. consisting of orchestrated web services) for the commercial services at hand.

To enable semi-automatic generation of service value webs, the specific contribution of this paper is twofold. Firstly, we lift *e^{3}value* to the industry level, that is to model the music industry, rather than just an individual business case as *e^{3}value* is normally used for. To do so, we present *e^{3}value skeletons*. Secondly, we employ the

[1] Software tool support can be obtained from http://www.e3value.com/.

skeleton to generate e^3*value instance* models for specific business cases. These instance models describe a service value web consisting of concrete actors, exchanging concrete things (e.g. services and money) of economic value with each other. The generation process uses an hierarchical configuration approach, plus a service profile of each actor involved.

The rest of this paper is structured as follows. Sect. 2 presents the IPR case study in more detail. In Sect. 3, we briefly introduce the e^3*value* methodology. Sect. 4 presents the e^3*value* skeleton for the IPR case. Sect. 5 introduces the notion of service profiles. Sect. 6 presents the hierarchical configuration process. Sect. 7 discusses tool support for the configuration process at hand. Sect. 8 discusses related work, and in Sect. 9 our conclusions are presented.

2 Case study: Clearing intellectual property rights

Our case study domain concerns the music industry and more specifically intellectual property right clearance. If a radio station in The Netherlands plays a music track, the radio has to pay to right owners. A right owner can be an artist, a producer, a composer, or a lyricist. So-called IPR societies act as a man in the middle; they collect money from radio stations and distribute this money over the right owners they are working for. These societies clear a specific right namely the right to *make content public*. Radio stations make content public, but bars, discotheques, supermarkets, etc. do also if they play music on their premises. Therefore, they also have to pay to right societies in order to make content public. In The Netherlands there exist two relevant societies concerning the right to make public, namely the BUMA and the SENA, the latter our case study partner. These two societies both collect money from intellectual property right users and pay the money to right owners. BUMA and SENA differ in the type of right owner they are working for; BUMA clears rights for publishers, composers, and lyricists, whereas SENA clears rights for the artists and producers. In other European countries a similar situation exists, but with different societies operating for different right owners.

As of today, right societies have contracts with branch accociations (e.g. with the accociation of discotheques, supermarkets, etc.) that mention a fixed fee to be paid (depending on the number of square meters floor space of the right user, e.g. the number of square meters of a supermarket). The future trend is towards *pay-per-play*, which is payment per played track, rather than a fixed fee. Another trend is to increase competition amongst IPR societies. Nowedays, a right owner can *not* select the right society that represents him. Rather, the artist will be represented by the society operating for his country. A future trend is however that right owners *can* select their society of choice. We take both trends as our point of departure. As a consequence, for each music track a unique service value web has to be generated, which is capable of clearing the intellectual property rights of that track, and which can pay the right owners the fees they are entitled to.

3 The e^3*value* methodology

To design and analyze service value webs, we have developed in earlier work the e^3*value* methodology [7]. We explain e^3*value* along the lines of the music case study, which is the IPR situation for The Netherlands. Key notion in e^3*value* is the conceptualisation of a service value web in terms of a graphical model, about which stakeholders can agree. An e^3*value* model depicts a network of enterprises creating, distributing, and consuming objects of economic value [7], such as service outcomes or money. The focus of the model is on *what* kind of objects enterprises must exchange to each other in order to cover consumer needs [2]. Fig. 1 shows (at the bottom) the modelling constructs of e^3*value* as well as (at the top) a model for the music case study.

The most important e^3*value* constructs are as follows: *Actors*, such as a BUMA and SENA, are economically independent entities (Fig. 1). A special kind of actor is the *Market segment*, such as “IPR user” (e.g. a supermarket), “artists”, “producers”, “publishers”, “composers”, and “lyricists”. A market segment of “artists” indicates that there are *many* “artists” rather than just one.

Actors transfer *value objects* (“money”, “music stream”, “right to make public”) by means of *value transfers*, which in turn connect *value ports*.

For value objects, some actor should be willing to pay, which is shown by a *value interface*. A value interface models the *principle of economic reciprocity*: Only if you pay, you can obtain the services and vice versa. So for instance, the “IPR user” pays “money” to obtain the “right to make tracks public”.

Actors perform *value activities*, which create something of economic value. Examples of those activities are “Playing background music”, “Collecting fees”, “Repartitioning fees”, etc. Value activities are supposed to make a profit, therefore actors are interested to perform such activities.

Finally, an e^3*value* model contains a dependency path, starting with a consumer need (“play a track”) and ending with a boundary element. Along the path are value transfers, value interfaces and connection elements. The dependency path shows how many value transfers are executed as a result of a consumer need. The dependency path is e.g. used to analyze the net cash flow for each actor involved.

---

[2] We do not focus on these exchanges are executed as this is the focus of a business process

Part of the dependency part are *AND* elements. For instance, AND element #3 represents that *both* “artists” and “producers” are paid as a result of executing the dependency path. Additionally, Fig. 1 uses so-called *explosion* elements, which are marked as EE #1 …EE #5. Such an explosion element represents that for each consumer need (“play a track”), *multiple* “artists” obtain “money” (as a music track usually is performed by multiple artists). The same holds for “producer”, “publisher”, “composer”, and “lyricist”.

An elaborated formalisation of the concepts in e^3*value* can be found in [7]. Fig. 2(a) illustrates the most important e^3*value* concepts and their relationships as a high level UML class diagram.

4 A skeleton for clearing IPR

The model in Fig. 1 is a model that holds for The Netherlands. However, for other countries the model can be different. For instance, there can be separate intellectual property right societies for collecting and repartitioning fees; or there is one society per right owner.

We have made a series of models as in Fig. 1, but for different countries, to analyze for commonalities and differences. Based on this analysis we develop a value *skeleton*, see Fig. 3. A value skeleton is supposed to hold for a specific industry (e.g. for the right to make content public), and should be reusable for the various countries. Therefore:

- Actors and market segments are omitted; as the assignment of actors and segments to a specific value activity varies per country.
- Similar activities (e.g. the “Creating’ activity”) are generalised into one activity.

The skeleton in Fig. 3 represents the following.

There is a customer need to “Play a track”. Also, there is an activity “Playing music”. This activity needs to obtain a music track and to pay accordingly. Also, this activity needs to obtain the “right to make public” (“RIGHT_MP”) for a specific track from the activity “Collecting fees”. The value interface is annotated with a ***** which means that many of these value interfaces may exist, one per specific right/right owner type combination.

From an IPR perspective, the “Streaming music” activity is also making a track public (namely to the activity that plays a track). Therefore, the “Streaming” activity needs also to obtain the right to make public (“RIGHT_MP”) for a specific track. Similarly to the “Playing music” activity, multiple value interfaces may exist, as the interface is annotated with a *****.

The activity “Collecting fees” obtains money from right users and provides the “right to make public” (“RIGHT_MP”) in return. The activity needs the right to

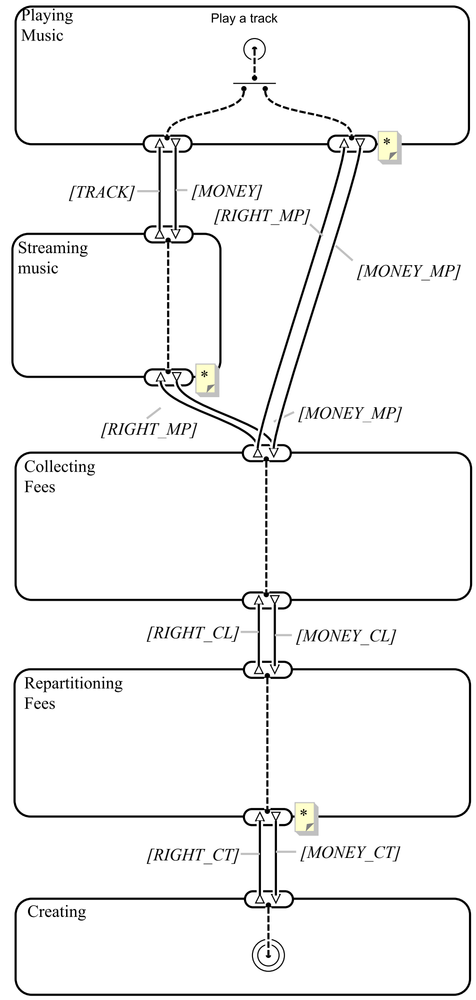

Figure 3. A skeleton for the IPR case study

“collect fees” (“RIGHT_CL”) and provides a payment in terms of money in return.

The activity “Repartitioning fees” provides the “right to collect fees” (“RIGHT_CL”) for a specific track. The activity pays a certain amount of money to the “Creating” activity (as performed by the right owners). In return, the ’Repartitioning activity’ obtains the “right to clear a track” (“RIGHT_CT”). The ***** annotating the value interface of “Repartitioning fees” indicates that multiple interfaces may exist; one per right owner.

Finally, the boundary element in the activity “Creating” indicates that we have considered all relevant activities.

5 Actor service profiles

The skeleton provides the groundwork to develop *service profiles*. A service profile shows (1) that a particular actor provides a service, (2) what is requested in return for a provided service, (3) what additional services are needed to produce the offered service, and (4) information about service instances, if required.

Service profiles are the building blocks to build an e^3*value* instance model in a semi-automatic way. Because all actor profiles are built with having the same e^3*value* skeleton in mind, these profiles can be connected with each other as lego blocks.

5.1 SENA's service profile

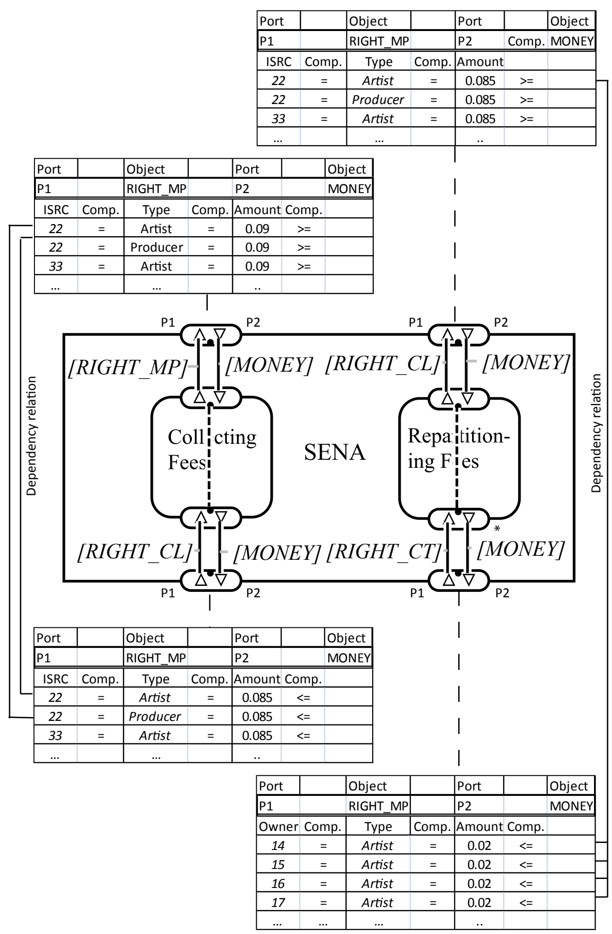

Figure 4. A service profile for the SENA

As an example, fig. 4 shows the service profile for the SENA. First, it shows the services or value activities SENA can perform: “Collecting fees” and “Repartitioning fees”.

For each service, the value interfaces are shown. For example, the “Collecting fees” service has a value interface with service outcome “RIGHT_MP” (“right to make public” in return for “money”. In order to produce “RIGHT_MP”, a “RIGHT_CL” (“right to collect fees”) has to be obtained, also in return for “money”.

Each value interface is annotated with a table. This table models the various instances of services SENA can provide. For example, the left-topmost table in fig. 4 shows which tracks SENA can clear, and can do so for which right owner type (“artist” or “producer”). The table tells that the interface has two ports, P1 and P2, offering and requesting “RIGHT_MP” and “money” respectively. The “RIGHT_MP” service outcome has a few attributes. First, music tracks are identified by a so-called ISRC-code [2]. Such a code identifies an existing music track in a unique way. The left topmost table thus shows that the SENA can collect fees for the ISRC track # 22 (“ABBA, Money, Money, Money”) and #33 (and in fact a lot more). Second, there is a right owner type (“artist” or “producer”) for who the service clears tracks.

Between value interface instances, dependency relations may exist. For example, for track #22, the “Collecting Fees” activity needs to obtain the “right to collect fees” (“RIGHT_CL”), for both the artists and the producers, as is indicated by relating the two left tables with each other by means of a dependency relation.

Fig. 4 shows also that SENA performs the activity “Repartitioning fees”. The activity offers the “right to collect fees” (“RIGHT_CL”), and requires the “right to clear a track” (“RIGHT_CT”). This activity has also tables associated. As the right most tables are also related with each other by means of a dependency relation, it can be concluded that for the track with ISRC code #22, four artists have to be paid: the artists with identifier #14, #15, #16, and #17.

A similar service profile can be constructed for the BUMA, the other intellectual property right society in The Netherlands. The tables for BUMA are similar as for the SENA, only the right owner type differs, as BUMA is operating for other right owner types (“publishers”, “composers”, and “lyricists”). Also, the right owners and the streaming parties have their own service profile, as based on the role they play in the skeleton.

5.2 Representing service profiles

To enable automated reasoning, service profiles should be represented in a computational way. Therefore, Fig. 2(a)&(b) shows the constructs to represent service profiles.

First, an actor *performs* a value activity. Performing a value activity is about producing service outcomes, which are of value for other actors. In case of SENA, it performs the activities “Collecting fees” and “Repartitioning fees”.

An actor has one or more value interfaces *assigned*. A value interface shows the objects of value (e.g. service outcomes or money) to be produced and consumed by the actor at hand. The value interface models the notion of economic reciprocity: "one good turn deserves another". Value interfaces *consist of* value ports. These ports produce or consume the actual value object. In case of the SENA service profile, the two value activities of SENA have each two value interfaces, and each interface consists of precisely one ingoing and one outgoing port.

Value ports can *have* attributes. In case of the SENA service profile, attributes are the ISRC code of a track, the actor type the right society is operating for, and the amount of money requested for service production.

In order to be able to generate a service value web with concrete actors, we have to represent the concrete services which can be offered, in terms of specific tracks, and specific right owners. Therefore, we have to extend the e^3*value* ontology to represent these concrete instances. In Fig. 4 these concrete instances are shown as tables near the relevant value interface. To represent such tables a value interface *has* value interface instances. Such a value interface instances refers to one row in the tables in Fig. 4. Value ports *have* value port instances, which in turn *have* attribute instances. The attribute instances correspond to the cells in the tables in Fig. 4.

Attributes *have* a comparison operator. Service profiles are matched by comparing the various attributes of ports of two actor with each other. During the matching the comparison operator ($<$, $\leq$, $\geq$, $>$, $=$, $\neq$, in range, ignore) determines how the attribute matching is done (see for an example Sect. 6.3).

In the case of SENA, the tables in Fig. 4 represent an excerpt of the internal databases SENA already has, to perform collecting en repartitioning of fees.

6 Hierarchical configuration of an IPR web

If a specific music track, say "Money, money, money" from "Abba", is played by a right user (e.g. a supermarket), a service value web has to be configured (or instantiated) based on the earlier defined skeleton and the service profiles. To this end, we use the skeleton, and the actor service profiles. Then, the topmost activity in the skeleton is assigned an actor, and next, this assigned actor configures the service value web, which is hierarhically seen below him (cf. [4]). This process continues, until we encounter a boundary element in the skeleton. A boundary element signals that no further actors need to be considered.

The instantiated service value web is presented in Fig 5. Below, we stepwise discuss the instantiation process.

6.1 Starting point: The consumer need

The instantiation process starts with the consumer need "Play a track", as mentioned in the skeleton in Fig. 3. We assume that the IPR user having this need is a "supermarket". This consumer need is instantiated with "Money, Money, Money" with ISRC 22. We assume that there exists a way to find the ISRC code for the corresponding track. This can for example be accomplished by a directory service on the Internet.

The skeleton (see Fig. 3) now prescribes two things: (1) a track should be obtained from the "Streaming music" activity; (2) one or more "RIGHT_MP" ("right to make public") should be obtained from the "Collecting fees" activity.

Starting with the latter, all actors performing the "Collecting fees" activity are asked whether they can provide the "RIGHT_MP" object. Each actor consults its service profile, and both SENA and BUMA reply that they can provide the 'RIGHT_MP" object for the specific track identified by an ISRC code. They do so by matching the requested ISRC code with the ISRC codes in their service profiles. The SENA finds two matches and so replies two times - one reply for the actor type "artist" and one reply for the actor type "producer". The BUMA replies three times - each reply for the specific actor type BUMA represents. As a result, the IPR user "Supermarket" has two value interfaces to represent the rights of making public of the "artists" and "producers" (as offered by SENA), and three interfaces to represent the rights of making public of "publishers", "composers", and "lyricists" (as offered by BUMA).

Additionally, there is one additional value interface, representing that a "track" with ISRC code 22 should be obtained from a streaming music provider, here "SkyRadio".

6.2 Sky Radio

As the skeleton shows, music is made public two times: (1) by the IPR user - here the "supermarket" - if the supermarket plays the music on its premises, and (2) by the streaming music provider - here "Sky Radio".

Therefore, "Sky Radio" needs to obtain the right of making public from all relevant right owners. As a consequence, "Sky Radio" asks for actors who can provide the "RIGHT_MP" for ISRC track 22. Similar to the "supermarket", both SENA and BUMA respond for their specific right owner types.

6.3 SENA and BUMA

So far, Fig 5 shows that the SENA is selected to provide "RIGHT_MP" for the track with ISRC code 22. SENA provides two times the "RIGHT_MP"; once for the right owner type "artist" and once for the right owner type "producer".

According to the skeleton, in order to provide the "RIGHT_MP", the performer of the "collecting fees" activity should obtain the "RIGHT_CL". By consulting its service profile, SENA knows that it should obtain the

"RIGHT_CL" two times: one time for the artists and one time for the producers.

Coincidentally, the SENA itself provides the "RIGHT_CL" for ISRC track 22, as can be seen from the actor service profile of the SENA. In terms of matching, (see Fig. 4), the lower interface instance of the activity "Collecting Fees" matches with the upper interface instance of the activity "Repartitioning Fees". Both instances refer to same ISRC code (#22). The "Collecting Fees" activity requires €0.085 or less, whereas the "Repartitioning Fees" activity requires €0.085 or more. Therefore, there is a match, one for the artist and one for the producer.

In other countries the situation may be different; e.g. another actor may provide "RIGHT_CL". In such a case, a new actor would appear in the instantiated model.

As a result, Fig 5 shows that the SENA performs both the "collecting fees" and the "repartitioning fees" activities for ISRC track 22.

Due to lack of space, Fig 5 only shows the SENA; the BUMA has a similar structure, only with different right owners.

6.4 End point: The right owners

The actor service profile (see Fig. 4) shows that in order to provide "RIGHT_CL" for track 22, the "RIGHT_CT" should be obtained from 4 different right owners, namely right owners with identifiers 14, 15, 16, and 17. Again, a match is found for all the right owners (14, 15, 16, and 17) by comparing the right owner identifier with the service profiles of the various right owners. In a similar way it is deduced which producer is required in order to obtain its "RIGHT_CT".

The skeleton shows for the 'creating' activity a boundary element, indicating that the instantiation process stops.

7 Tool support

For the configuration process at hand, we have developed a first version of a configuration tool. This software tool supposes the availability of service profiles. These profiles are represented as RDF [13] files. Furthermore, the tool uses the stated consumer need as an input for the configuration process. Then, a conceptual model cf. Fig. 5 is generated, which can be visualized by the e^3*value* software tool [3].

8 Related work

One of the first attempts to achieve automatic configuration of services was proposed by Omelayenko [14]. The author applies a semantic web approach called open-world skeletal planning. In this way the configuration problem is solved through a planning-based reasoning which uses a skeleton to guide such configuration process. Even though this approach solves a configuration problem, the so-called skeletons are not aligned with value modeling. Consequently the configured bundle focuses more on the ordering of services to be executed rather than the value flow among those services, which is more important in our research.

Baida [3] proposes an ontological approach for service bundling. This approach uses three types of ontologies, the first two represent the demand and supply perspectives respectively, as a third ontology performs the configuration process. Therefore by matching two perspectives it is possible to generate a set of service bundles, nevertheless this approach does not offer a straightforward methodology for selecting one of them.

On the other hand, de Kinderen *et.al.* [11] aim to a (semi)automatic service bundling through a process of matching ontologies. As in Baida's work, in this approach there are two main ontologies for representing consumer needs and provider offerings. The approach applies a Propose Critique-Modify problem solving method for iteratively matching consumer preferences with service benefits, *i.e.* user interaction helps to configure service bundles.

None of the aforementioned works is able to generate an "final/instance" e^3*value* model that can be later deployed through web services, which is also one of our main goals.

As another line of research, semantic Web research resulted in a set of design principles, collaborative working groups, and a variety of enabling technologies including WSMO/X/L [15], XML, RDF(S), and OWL [17] that are becoming de facto formats for structuring and exchanging data and software services on the Web. Currently a number of large-scale integrating research projects (e.g., SOA4ALL [4] are running to align commercial and software perspectives on services, but are still much emphasizing more on the technical challenges like interoperability, scalability, orchestration, and discovery of services, without considering the economical feasibility or profitability of running them.

9 Conclusions and future work

In this paper, we have proposed an approach to automatically generate service value networks, based on service profiles and a consumer need. Our approach contrasts with web-service configuration since we focus on *commercial* services whereas web-services are facilitating interoperability and orchestration between *software components*.

Service profiles represent what actors offer of *value* to their environment, and what they request in return for that.

[3] see http://www.e3value.com/ to download this tool

[4] http://www.soa4all.eu

The reasoning process starts at the consumer need and then adds services from the service profiles until the boundary element is reached. Services are added if the requested service of an actor matches to an offered service of another actor. To this end, service profiles contain value interface instances of the performing actors, which are to be matched with each other.

We have developed elementary tool support to facilitate the configuration process. The tool assumes service profiles expressed as RDF triples, and a stated consumer need, and then configues a service value web (also expressed as RDF triples) by hierarchically reasoning.

Regarding future work, our approach can be extended into a number of directions. Currently, the software tool does not use the skeleton during the reasoning process. The skeleton plays a prescriptive role while developing service profiles, thereby guaranteeing that the service profiles can be connected to each other by the configuration process. The service network is generated by hierarchically matching requested services with offered services only. Adding knowledge about the skeleton to the reasoning process restricts the search space during the matching process, and therefore results in a more efficient matching process.

For now, we assume a fixed consumer need, which results in obtaining a number of services to satisfy that need. In a future version, the reasoning process should include reasoning about how (by which services) a specific consumer need can be satisfied (cf. [11]).

Our case study does not include service selection and prioritization yet. This is because rights of specific right owners are cleared and repartitioned by precisely one right society. However, in case of competition multiple, alternative service value webs can be generated, satisfying the same consumer need. A selection and prioritization procedure is then necessary to make a reasonable choice for a particular service vale web.

Finally, we use now hierarchical reasoning, meaning that each actor configures the value web that is hierarchically below him. In future work, we will focus more on self-organizing networks. Ultimately, by removing the current assumptions and limitations, and devising richter service profile descriptions, we will be able to develop more self-organising eco-systems.

Acknowledgements. The authors wish to thank Bert Hazelaar of SENA for providing the case study in this paper. This research has been partially funded by the NWO/Jacquard project VALUE-IT no 630.001.205.

References

[1] Introduction to uddi: Important features and functional concepts. october 2004. Available from http://uddi.org/pubs/uddi-tech-wp/pdf.

[2] International standard recording code (isrc) handbook, 3rd edition. 2007. Available from http://www.ifpi.org/content/library/isrc_handbook.pdf.

[3] Z. Baida. *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*. PhD thesis, Vrije Universiteit, Amsterdam, NL, 2006.

[4] R. I. Benjamin, T. W. Malone, and J. Yates. Electronic markets and electronic hierarchies. *Communications of the ACM*, 30:484–497, 1987.

[5] R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana. Web services description language (wsdl) version 2.0 part 1: Core language. *Recommendation 26 june 2007, W3C*, june 2007. Available from http://www.w3.org/TR/wsdl20.

[6] D. Fensel, H. Lausen, A. Polleres, J. de Bruin, M. Sollberg, D. Roman, and J. Domingue. *The Web Service Modeling Ontology*. Springer, Berlin, D, 2006.

[7] J. Gordijn and H. Akkermans. Value based requirements engineering: Exploring innovative e-commerce idea. *Requirements Engineering Journal*, 8(2):114–134, 2003.

[8] C. Grönroos. *Service management and marketing: customer management in service competition*. Wiley India Pvt. Ltd., 2007.

[9] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon. Soap version 1.2 part 1: Messaging framework, second edition. *Recommendation 27 april 2007, W3C*, april 2007. Available from http://www.w3.org/TR/soap12-part1.

[10] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon. Soap version 1.2 part 2: Adjuncts, second edition. *Recommendation 27 april 2007, W3C*, april 2007. Available from http://www.w3.org/TR/soap12-part2.

[11] S. Kinderen, de, J. Gordijn, and H. Akkermans. Reasoning about customer needs in multi-supplier ict service bundles using decision models. In *Proceedings of the 11th International Conference on Enterprise Information Systems*, pages 131–136, 2009.

[12] C. Lovelock. *Service Marketing - People, Technology, Strategy*. Prentice Hall, 2001. Fourth edition.

[13] F. Manola and E. Miller. Rdf primer. 2004. Available from http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[14] B. Omelayenko. *Web Service Configuration on the Semantic Web*. PhD thesis, Vrije Universiteit, Amsterdam, October 2005.

[15] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, Polleres, C. Feier, C. Bussler, and D. Fensel. Web service modeling ontology. *Applied Ontology*, 1(1):77–106, 2005.

[16] G. Shostack. Breaking free from product marketing. *The Journal of Marketing*, pages 73–80, 1977.

[17] M. K. Smith, C. Welty, and D. McGuinness. Owl web ontology language guide, http://www.w3.org/tr/owl-guide/, accessed, 2004.

[18] J. Spohrer, L. Anderson, N. Pass, T. Ager, and D. Gruhl. Service Science. *Journal of Grid Computing*, 6(3):313–324, 2008.

[19] D. Tapscott, D. Ticoll, and A. Lowy. *Digital Capital - Harnessing the Power of Business Webs*. Nicholas Brealy Publishing, London, UK, 2000.

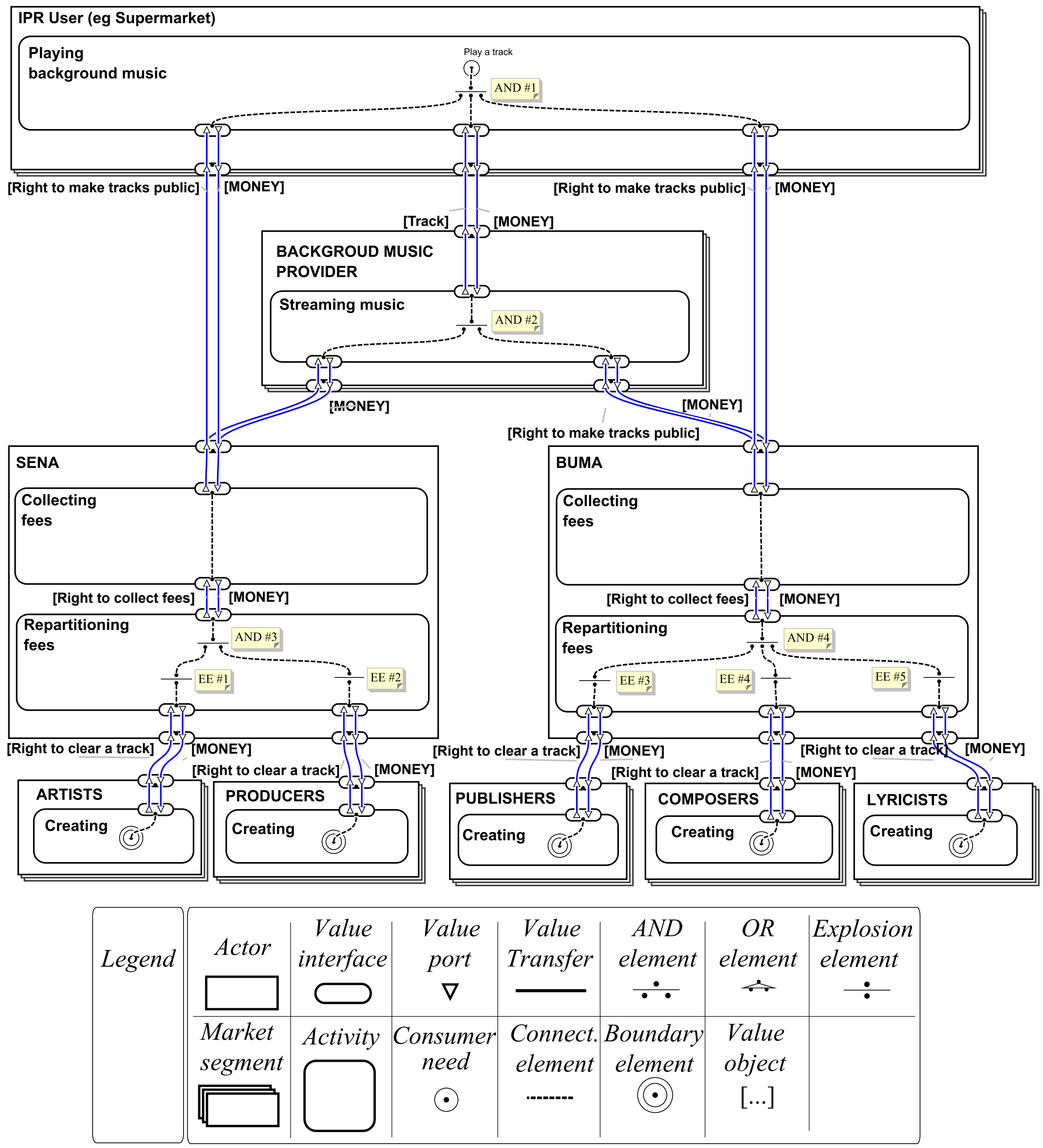

Figure 1. The IPR case study

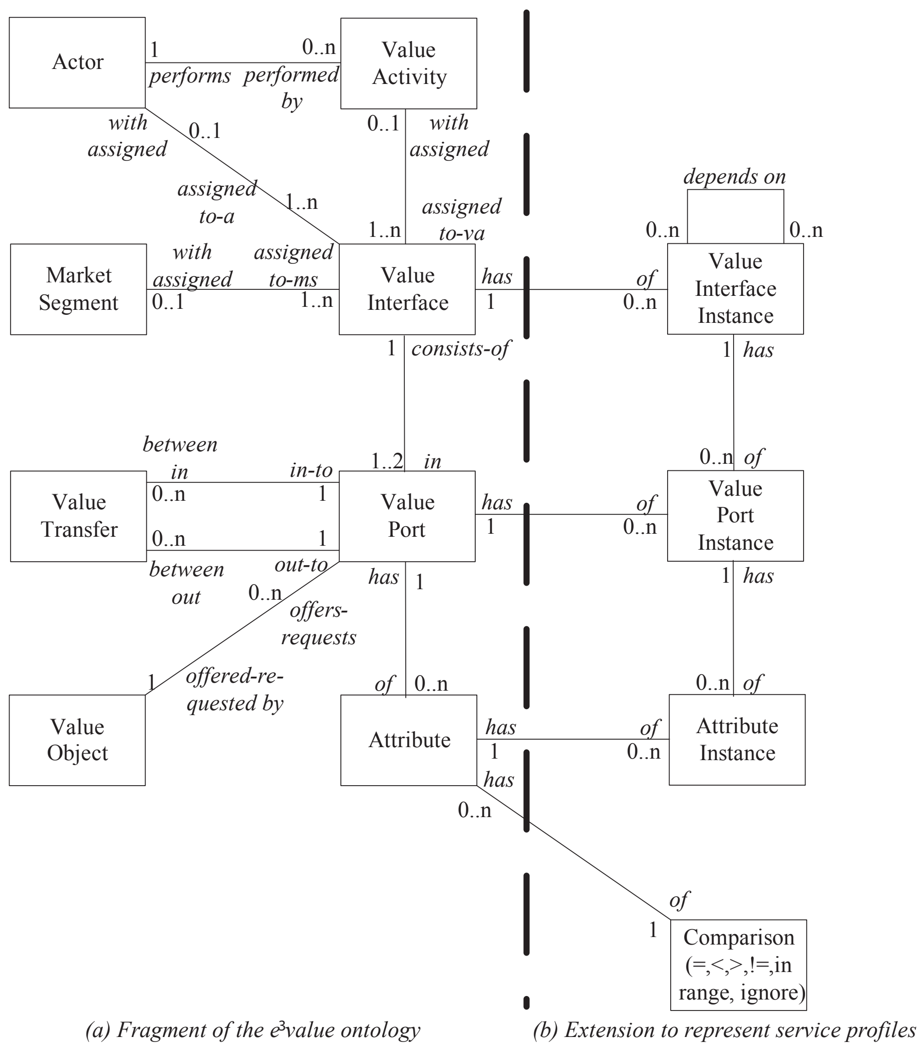

Figure 2. The e^3*value* ontology and its extension to represent service profiles

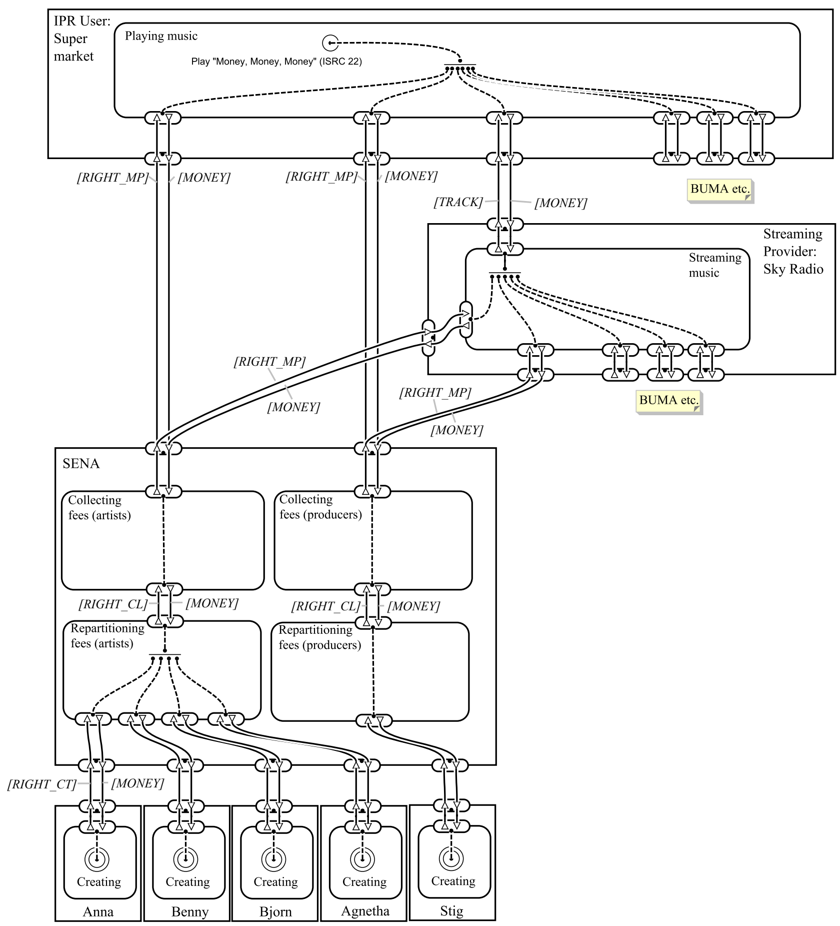

Figure 5. An instantiated service value web