Introducing Advanced Service Catalog Capabilities in Croatian Telekom NGOSS Infrastructure

Damir Medved[1], Hrvoje Tutman[1], Dominik Periškić[2], August-Wilhelm Jagau[3]
[1]GDi GISDATA, R&D Department, Petra Kobeka 13/1, HR-51000 Rijeka, Croatia
[2]Hrvatski Telekom d.d., IT Department, Kupska 2, HR-10000 Zagreb, Croatia
[3]Telcordia Germany GmbH**,** Herriotstrasse 1, 160528 Frankfurt am Main, Germany
damir.medved@gisdata.com, hrvoje.tutman@gisdata.com, dominik.periskic@t.ht.hr, ajagau@telcordia.com

Abstract - We are living in the era of pervasive computing – and that fact has tremendous impact for operations of typical telecom operator.

Customers are more and more demanding, picky, and often better informed than CSR in Telco shops. They are expecting real time provisioning and multiple choices – especially ability to tailor products to their needs – anytime, anywhere and on every possible device they use (from notebooks, tablets like iPad to smartphone). We are moving into area of CMR (Customer Managed Relationship) - more advanced concept than CRM, where customer is taking control over products/bundles and tailoring to his own need.

Sharing control allows co-creation of value and better bounding between customer and operator, but on the other side, creates huge demand for operator IT infrastructure often not ready for this type of interactions.

This paper is describing key aspects of improving Croatian Telekom NGOSS infrastructure with new capabilities introduced by COMPASS project that will enhance customer experience and allow Croatian Telecom to maintain leading technological position on Croatian market. COMPASS project introduced new approach to order management by implementation of the new paradigm – Catalog Driven Order Management.

I. INTRODUCTION

Focused on tracking telecom products life, we can identify the following important challenges for typical telecom operator:

- The introduction and market launch of new products in an effective and cost efficient manner
- Standardization of definitions and business interactions, which implies seamless product introduction into existing BSS and OSS systems (BSS - Business Support System and the OSS - Operations Support System) based on leading industry standards [1,5,6,7,8]
- Improving business processes involved in bringing new services to customers [2,3,4]
- Improving customer satisfaction through the improvement of quality of supplied services / products - preparing for CMR

The optimal methodology for the operational implementation of these demands is a new reflection on the nature of the products. Products should be decomposed to all services it actually contain - and moving forward down the chain – identify which are basic building blocks (resources/ components) needed to build such a product.

It is essence of the so-called PSR (product-service-resource) model implemented in the form of Product /Service Catalogue creating foundation of Catalogue Driven Order Management [9].

Typical example is the so-called Triple-Play product that has service for broadband Internet access, IP television and IP telephony and implemented over copper or fiber infrastructure of telecom operators.

In such, Catalogue driven process, requirements for new products as defined in the order are first decomposed to building blocks (components) and services together with necessary / desirable infrastructure. After the identification of requirements, in accordance with business ruled defined in Catalog, preliminary feasibility check is performed within Unified Resource Layer (Network Inventory and other applicable sources like historical account data, fraud management data, etc.) to find options to implement the required services / products.

At this stage of the exercise, various optimization are performed (depending of business rules that could be economically or QoS driven) to ensure optimal utilization of technological infrastructure. The consequence is the possibility of

recombination and grouping a number of tasks into logical units and further optimization in the field work domain.

Such an approach allows the team to develop new products within a unique platform for the creation of standardized technical specifications, which include the definition of the product itself, the components that comprise it and the resources needed for implementation. Using defined specifications, this simple product can create variations of new products, exploring the complexities of implementation and finding optimal variant of either technological or economic sense. Thus, products can be designed in a very short time to offer to the market. An integrated system for the implementation of business rules allows further optimization of business processes and properly selecting infrastructure in accordance with prescribed rules.

The quality and uniqueness of PSR model is of great importance for efficient and rapid introduction of new services. To innovate/improve the existing service it is very important to ensure that the P-S-R model is a simple enough and well described, on the other hand must efficiently and more accurately portray the technological capabilities of telecommunications infrastructure owned. Last but not the least, BPM segment is another critical factor for the operator – significant amount of possible optimizations is buried underneath often too complex processes (especially in field activities domain) therefore holistic approach is highly recommended when optimizing entire E2E process.

II. DYNAMIC SERVICE CATALOG (DCAT)

The DCAT service catalogue is storing all details of the products and services supported by the NGOSS Solution. Hence it is imperative to carefully layout and organizes the capture of these product and service data. DCAT does this by mandating a specific structure for the product models and by supporting a tool called Service Designer.

This chapter provides an introduction to the DCAT product model lay-out and to the role of the Service Designer in creating product models. It concludes with some thoughts concerning the re-use of product models and the benefits derived from that.

1.1 Model Structure

All product models stored in DCAT are organized along the same data modeling and information structuring guidelines.

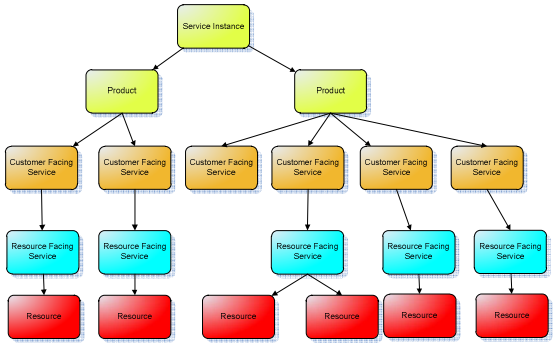

Figure 1: COMPASS Product Topology

Product models consist of the following components:

- products,
- services, and
- resources

The following sub-sections along with the picture shown in (Figure 1) provide an overview about these three data objects and provide some details on how they will be used in the NGOSS Solution.

1.1.1 Product Level

The product level represents the top most level of the product specification tree structure. In this context it is important to distinguish between commercial products and technical products. Commercial products and product bundles1 are distinguished by the fact that they are associated with price bands, discounts, sales channels, and geographical and seasonal availability. In short (commercial) products deal with the commercial and contractual aspects of a telecommunications service offering and, hence, it is no surprise that they are handled by CRM and/or the billing systems. Simple commercial products usually map on a one-to-one basis directly to the topmost service component which – for this reason – is oftentimes called either a technical product or a customer facing service2. Simple products are carry attributes that allow a customer to ask for a product profile which suits her/his specific needs.

To enable the technical provisioning of commercial products there must be a precisely defined mapping between commercial and (technical) OSS products. This mapping is usually

[1] A product bundle, in effect, is just a collection of products sold together, usually at a bundle price.

[2] Customer facing services are the first level of decomposing a product into its constituent technical components. They could be viewed as services offered to customers.

bi-jective. DCAT supports the possibility to have so called root products (product which can be linked) at various levels. This functionality facilitates the creation of arbitrarily deep structures of products where any one of the sub-assembly type products can be selected as root product.

The name of the OSS product as the topmost element represents the entry point of a specific model.

DCAT supports a high degree of flexibility to model products from already existing products. As an example consider the alternatives available for creating complex products. If a new creation should simply combine existing (top-level, aka customer facing) services then by making these services into products it is possible to generate the new offering as product bundle. As a case in point consider the Triple-Play product combining Internet, VOIP, and IPTV as customer facing services. However, if the new product depends on some specific characteristics like consistently high and reliably delivered bandwidth as is necessary for an IP-TV offering, for example, then the better choice is to create the product from a combination of customer facing services, each of which has a specific configuration dedicated to the new product. In the case of IP-TV such customer facing services could be contents, authentication, and broad-band access. And broad-band access as a customer facing service could be specified in this case to require a fiber-optics link with a minimum bandwidth and minimum availability.

As shown in Figure 1 products are first decomposed into customer facing services. Customer facing services are in turn ultimately decomposed into resource facing services. Resource facing services are services which combine various resources.

1.1.2 Resource Level

A resource represents either a physical or a logical resource in a DCAT product specification. It generally stands for the smallest unit of value-add that can be directly configured in the network or its support infra-structure. Hence, with resources the decomposition of a product and its constituent services (c.f. Figure 1) reaches its most detailed level. The resulting resources at this level are used to provision the various features and functions of a product.

Logical resources can be directly mapped to software components which are triggered by the DCAT Engine as part of the process of preparing an activity list. In this case a logical resource points to a software service which is exposed on ESB as an SOA component. Such SOA components are called Resource Components.

In addition to logical resources there are also physical resources, which represent physical elements in the network.

1.1.3 Product Model

A product model represents a set of entities that fully specify the product, its constituent services and resources so that instances of them, which represent their per-customer usage profile, can be provisioned and activated in all systems implicated by a given product.

Modeling in DCAT is like working with Lego bricks. Each component in DCAT represents one of these basic bricks. At the lowest level the bricks represent resources. Resources can be combined in various ways to form more complex structures, generally called services or resource facing services. As shown in Figure 1, the resource facing services can be further structured to finally yield customer facing services. Customer facing services represent elements of products which are exposed to a CRM system like Croatian Telecom own Donat.

1.2 Service Designer

The DCAT Service Designer is an application used to construct models. Service Designer uses a drag and drop approach to create a tree of modeling artifacts that represent a specific product model.

DCAT should be unaware of the logic behind resources and should be used only for modeling but with some constraints in designing resource relationships. DCAT takes care of creating the structure of the model, binds design-time attributes to values, and specifies run-time attributes along with their value domains.

The specification of a model defines a tree of components (products, services, and resources) which is refined into one or more variants, called implementations. Each of the components used in a specification must be associated with at least one implementation (if this rule is violated, a product model cannot be commissioned).

1.2.1 Specifications

Specification represents a data structure which consists of both the hierarchy of the components

and the characteristics associated with each and all of the components.

Specifications are used to describe information about a particular variety of products, services, resources, or business interactions. Specifications are defined by the set of attributes and other associations including specification characteristics, specification characteristic groups, and specification rules set packages. Specifications are also definitions that provide structure for the creation of offerings which can be presented to other systems through the service instance. Each specification will have one or more implementations in order to run recommendation engine.

The DCAT supports 4 types of specifications: the business interaction, the resource, the service, and the product specification type.

The resource specifications are positioned on the lowest level of the implementation tree, so they could have a huge number of combinations of its own parent and children, therefore we omit the details of component parent and child attributes for these DCAT objects.

1.2.2 Offerings

Offering is a DCAT object that specify particular product through assignment of actual values to the characteristics contained in associated specification. Each DCAT specification type, except business interaction, could have an offering, and the actual characteristic values differentiate one offering from another. In the context of the COMPASS project, the offering is a criterion object where recommendation engine is acting in order to generate the best product candidate solution for customer, therefore each of components would have one offering.

A bit different case is with product level offerings. These offerings are directly mapped to the GDi product packages.

III. COMPAS END TO END VIEW

COMPASS [9,10] is fundamentally a service fulfilment solution (Figure 2). As such it is able to accept customer orders issued from the DONAT CRM system. The COMPASS orders are received by the Expediter order manager, enriched and then sent on to the DCAT Engine for order decomposition. When the DCAT Engine sees an order it builds an implementation data structure, the data items of which are populated by running the DCAT Recommendation Engine.

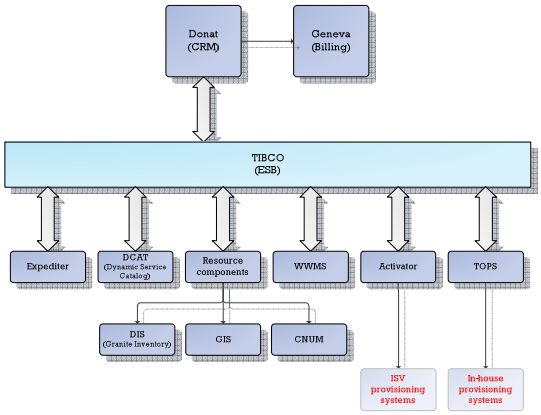

Figure 2: COMPASS SOA Architecture

There are generally two approaches for implementing the order management processes. One is using a “GENERIC” flow in Expediter which is product / service agnostic and within that approach all the details specific to each product / service resides in DCAT including calls towards Resource Components and task list creation.

The second approach is with an Expediter specific flow for a given scenario which comprises all defined actions such as calling of individual Resource Components (part of the provisioning engine), calling external systems through specific tasks and correlating them. In the second approach the DCAT is used to yield the tree of a decomposed product / service structure, however, the role of DCAT stops after decomposition. Unlike the first approach where the DCAT Engine goes "one step further" and consumes all of the resource components and builds a task list that is then sent to Expediter for distribution to destination systems sequentially. In the second approach, there is no such limitation, and it is possible to have parallel execution tasks and dependent task (next task depends on the results of the previous).

These are the key differences between the two different approaches of the design and execution of specific order management process scenarios.

In addition to the order management process, it is necessary to fulfill other process requests.

Suspension and Resumption processes are solved within the same context as the order management process by treating them as special cases of change orders. The fault repair process can reuse the same North Bound Interface (NBI) as order management, only the flow execution will be different. Such processes will be mainly executed in WWMS (Workflow and Workforce Management System), and Expediter will support it for specific tasks.

The Quick Order process is a special case of a process which is initiated by some technical system and executed inside the OSS system before any message is sent to CRM. As every technical system has its specific characteristics, it is not possible to define a generic interface which all of the technical systems will be able to deploy for initiating a Quick Order process inside Expediter. Therefore, all of the possible quick order processes have to be identified and treated separately. Quick Orders are mainly executed inside Expediter with calls to specific Resource Components if changes to systems such as telecom inventory are necessary, calls to automatic activation platforms or WWMS for a manual activities and ultimately informing BSSs (CRM) thusly ending process execution. Information towards the CRM system at the end of the Quick Order processing will be sent by invoking a separate CRM channel for carrying such messages.

IV. FURTHER WORK

In previous chapters we explained key concepts of Service Catalog and P-S-R relationships. These concepts are successfully applied within company, but key question for future development is how these concepts could be extended to inter – company domains. Key idea is ability for customer (in already mentioned CMR paradigm) to create products/services bundles that are crossing boundaries of each particular company (Figure 3). Taking apart economical and competition type of impact, it is obvious that one single service catalog for all providers could not exist, as everybody should have its own solution. One of proposed solutions is to user so called Semantic Web [11] approach that is giving key ingredients for the semantic architecture of Web Services. In general by applying mentioned concepts using OWL-S (Web Ontology Language for Services, http://www.w3.org/Submission/OWL-S) every Web Service can be described with ontology as defined in OWL (Web Ontology Language), a semantic mark-up language for publishing and sharing ontologies on the WWW. Therefore, the semantic architecture of Web Services supports knowledge exchange among collaborating electronic business (e-business) actors in the digital economy.

Ontologies provide a shared vocabulary to represent the meaning of entities while knowledge representation provides structured collections of information and inference rules for automated reasoning. As a result, intelligent software agents can interpret and exchange semantically enriched knowledge for users [12].

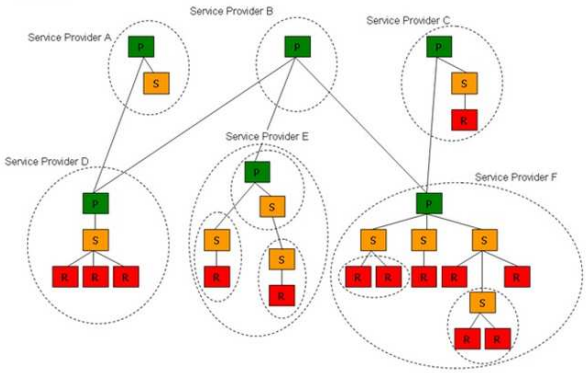

Figure 3: Inter-Company relationships

Regarding practical aspects, almost all COMPASS components could be accessed via Web Services, therefore are able to be described with OWL, but creating all necessary ontologies will require significant resources – and Go/NoGo decision to complete this activity is depending on further socio-economic developments and business justification.

V. SUMMARY AND CONCLUSIONS

By completing first phase of the COMPASS project Croatian Telecom achieved several benefits:

- Implementation of the pure SOA architecture that completely fulfilled expectations both in terms of functionality and performance (in six months over 650.000 orders where processed without substantial issues).
- Full compliance with leading standards - TMF eTOM, TAM, SID, etc. In fact Croatian Telecom has one of most „standards-aligned“ NGOSS architecture in the industry.
- Clearly set boundaries between systems, maximum utilization of specialized functions of each system and facilitated upgrade of individual systems. Decupling functionalities already demonstrated value when dealing with components upgrade reducing significantly both effort and necessary time.
- Deployment time to implement RFC significantly reduced, reduced maintenance costs, complexity, system configuration is smaller. Actually, deployment time was reduced in same cases from months to weeks, and form weeks to days. Further enhancements will be achieved when project will be completely finished.
- IT Infrastructure ready for CMR type of operations, and that should bring Croatian Telekom significant advantage over competition in years to came.

VI. REFERENCES

[1.] TM Forum GB921: eTOM Concepts and Principles 9.10

[2.] TM Forum GB921-D: Process Decompositions and Descriptions 9.10

[3.] TM Forum GB921-E: End-to-End Business Flows 9.10

[4.] TM Forum GB921-F: eTOM Process Flow Examples 7.50

[5.] TM Forum GB921-P: An eTOM Primer 8.10

[6.] TM Forum GB921-U: User Guidelines for eTOM 8.10

[7.] The Information Framework (SID) Suite (GB922) 9.00

[8.] TM Forum GB929: The TAM - The BSS/OSS Systems Landscape 3.00

[9.] Gartner Competitive Landscape Catalog Sept 2010

[10.] TM Forum - COMPASS Catalyst Results V1.1 Final

[11.] http://infomesh.net/2001/swintro/

[12.] Podobnik,Vedran; Tržac, Krunoslav; Ježić, Gordan; Lovrek, Ignac: "Agent-based Discovery of Data Resources in Next-generation Internet: An Auction Approach", proceedings of the 2007 Networking and Electronic Commerce Research Conference (NAEC'07) Riva del Garda, Italy

VII. GLOSSARY

Donat – Croatian Telecom CRM system

WWMS – Workforce Workflow Management Systems by Fornax

Expediter – Telcordia Technologies Inc. - Order Management