Cloud Communication Service

Michael Cochez[1], Sami Helin[2], and Jiawen Chen[1]

[1] Department of Mathematical Information Technology
University of Jyväskylä
P.O. Box 35 (Agora), 40014, Jyväskylä, Finland
michael.cochez@jyu.fi, jiawen.chen@student.jyu.fi
[2] Steeri Oy
Tammasaarenkatu 5,00180, Helsinki, Finland
sami.helin@steeri.fi

Abstract. Cloud computing has opened the path to more on-line service oriented business models. Customers are interacting with enterprises' digital systems trough a multitude of interfaces. We regard each of these possible interfaces as a communication channel. When an organization owns multiple systems with each its own communication channels, a user might get a fragmented experience and use of the channels is likely sub-optimal. From our past research we will summarize a basis for a communication framework and how measuring quality of the semantic models in that framework can be achieved. Then we give a compact overview of techniques we proposed to process the knowledge extracted from high velocity communication streams, i.e. Big Knowledge, using a system which borrows concepts from biological evolution. From our current research focus, we present techniques which should help to reach a good user experience. We mainly look at how we could filter out unwanted messages and how we could make recommendations to users who might be interested in certain types of messages. Finally, we will suggest user interfaces for the proposed solutions.

1 Introduction

In this chapter we present the current state of the achievements in the Cloud Communication Service business case which is part of the Cloud Software Program [1]. The results presented here summarize prior published work and extend it with our current efforts.

The idea behind the Cloud Communication Service (CSS) is to solve problems which arise when communication oriented cloud services interact with customers and other systems. The components needed for this communication are typically replicated amongst the different systems which are in use by an organization. These components often work strictly in parallel, not allowing for any flow of information from the one to the other. The overall result is that the systems will not make efficient use of the available communication channels and the customer will realize this fragmentation when interacting with the company. One example could be a customer who has bought a new smart phone. When this person contacts the help desk with a certain problem at a later point, it would be helpful if the person who answers this request can help with knowledge about previous interactions. It would also be beneficial to know, or not have to care about, the way this customer prefers to be contacted. A broader analysis of the problem and potential benefits of solving it was given by Nagy [2], we will summarize its main points below. During previous research we have also devised a basic multi-channel communication framework for solving part of above mentioned challenges. [3] In parallel, we have started to build a model for evaluation of the quality of the ontologies used in this system. [4] We also worked on a way to extract the core information from the stream of data, which the system has to process at high velocity. [5] In this chapter we will give an integrated view on these parts and extend with our current work on human interaction.

The chapter is structured as follows: first, we introduce a business scenario in which the Cloud Communication Service would function. Then we give an overview of the past work followed by our current efforts. Finally, we present some of the user interfaces which were developed for the Cloud Communication Service.

2 Business Scenarios

In modern society, on-line shopping and cooperating are major trends in business. To buyers, it brings plenty of benefits, such as convenience and lower prices due to increased global competition. To companies, on-line shopping stimulates the activities of business, which can now sell on the global market to customers who were unreachable before. The on-line trade has however introduced a new challenge, namely the lack of communication and comprehension towards customers. For example, certain buyers do not plan to buy more as a single item from your web shop. In a face-to-face situation, the shopkeeper could advise the customers other related products or guide them to relevant services. This is more difficult in an on-line sale and many companies have spent great effort towards the delivery of targeted advertisement, often based on customer preferences and shopping history.

Further, customer buying processes have evolved to utilize the modern variety of channels available. This has made understanding customers and providing consistent customer service and communication more challenging. When thinking about the interaction with customers, they must be put at the core. Then, the channel and content selection has to be based on the individuals' profile which includes their settings and history of previous interactions. One of the goals is to reduce negative emotional effects to customers, such as these caused by trash mail.

An example scenario of a dynamic and customer focused communication would be a shopper wanting to buy a new TV. He is a member of an electronic retail chain's customer program. He then seeks information on the store website and downloads a brochure. Then he goes on and asks for referrals from social media websites. During the purchase process he starts receiving more TV focused ads in the newsletter and promotional discounts. The on-line website gets personalized for a more convenient shopping experience around topics of interest. And if he finally decides to go to store to make the actual purchase, he also receives suggestions relevant to his needs. After the purchase process he receives a simple feedback query from the company by SMS asking the customer about his experience with service and whether he would promote the shop to his friends.

In this example case, the complete buying process was made relevant with coordinated utilization of several channels:

1. Social Communication website followers
2. Surfing on-line on the web shop
3. An SMS notification
4. A newsletters via electronic or postal mail

We suggest that to achieve consistent multichannel customer dialog the company should:

- manage content for multiple channels; ranging from encountering the customer in person to digital channels with varying capabilities,
- split the content to atomic pieces for it to be possible to dynamically draw the finally delivered content based on the customer needs and channel choice,
- understand customer behavior and interaction in different channels to coordinate customer contacts and generate customer insights,
- instead of providing a set of static messages draw relevant content blocks to match customer interaction points, and
- measure how well different operations help to reach the goals set by the company.

These topics map to research questions related to:

- automated understanding of messages to increase the throughput in the system without losing quality,
- Big data related issues when trying to understand and measure behavior,
- the definition of a communication framework including messaging, profile management, and content and channel selection,
- Recommender Systems to handle dynamic content creation which benefits from similar tasks in the history of the system.

The research and development effort in cloud communication service tries to address these challenges. Some research results regarding these topics can be found in the next sections.

3 Past work

Up till now three main facets have been investigated. First, we created a framework to handle the complexity of the communication system. This framework is summarized in subsection 3.1. Then, we looked at how we can define quality of an ontology which is one of the building blocks for the framework. This effort is described in subsection 3.2. Finally, a book chapter [5] about evolving knowledge ecosystems for big data understanding was published. That chapter encompasses several facets related to this research topic. An overview of these topics can be found in subsection 3.3.

3.1 Framework

The results of the main effort towards solving some of the issues of the multi-channel communication problem can be found in the framework paper by Nagy [3]. At the current moment there is no complete implementation of the framework. Rather, the theoretical framework is used as a guideline for further extension of the current implementation described in section 5.

An overview of the framework can be seen in Figure 1. The structure consists of two main building blocks. The first one, the knowledge base, is depicted in the grayed box and is divided in several ontology fragments and sub-knowledge bases. The second one, the message conversion engine, is responsible for handling incoming and outgoing messages.

The ontology in the knowledge base is subdivided in 5 parts: the *customer ontology* to model users and the way they can be contacted, the *action ontology* used for the description of high level goals, the *message ontology* used for the classification of messages, the *commodity ontology* used to described goods and services, and finally the *channel ontology* used for classifying channels and their properties. These ontologies are defined as part of the framework. The article identifies that the ontology can be expanded depending on specific business needs. Further arguments for the use of ontologies from the original framework proposal [3] include:

Expressiveness: When using ontologies it is possible to represent the concepts of the framework and the ones from the business domain using the modeling language. Ontologies also allow for later extension when the business needs change.

Soundness and completeness: Several languages for ontologies have been formalized. These formalizations include ways to deduce new information from existing facts. This deduction, also called reasoning, is sound, i.e., correct and also complete, i.e., all facts which can be found will be found.

Computational complexity: The sound and complete reasoning happens with a reasonable computation complexity.

Tool support: There are several mature tools available for ontology design, management and reasoning.

The message conversion engine first converts messages templates into abstract messages. A message template is like a blueprint for a message and contains placeholders for information which the engine will fill. The engine then chooses a destination channel and the abstract message is converted to a concrete message with a structure depending on the destination channel.

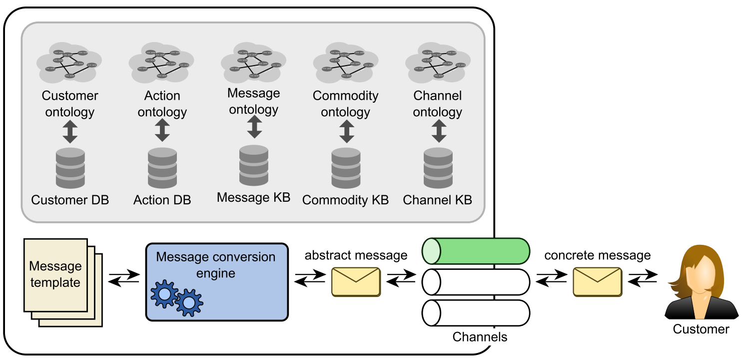


Fig. 1. Multi-channel communication framework overview (Picture credit: Michal Nagy)

3.2 Quality of an Ontology

When creating an ontology, there are numerous ways of defining its quality. The main contribution of the work performed in the frame of the Cloud Communication Service is that it states the finding of an ontology with the highest quality possible as a Context-dependent dynamic multi-objective optimization problem. [4] A minor contribution is the recognition of fuzzy ontologies as a way to state inexact knowledge.

First, we discussed different features of an ontology as can be found in the literature. These features include coverage, cohesion, and coupling which are metrics representing how well concepts and relations between them are covered, how well concepts in the ontology belong together, and how strong the linking between this and external ontologies is, respectively.

These features can be measured from a given, possibly fuzzy, ontology. When the features are combined with the context in which the ontology is used, it is possible to state a quality of the ontology. The context comprises many factors. Examples include the performance when used in a system, the memory needed, how easy it is to scale the ontology, and how well it integrates into frameworks and interfaces.[6]

Next, the article shows that the quality of an ontology can not always be considered to have a single dimension only. This because of the fact that there is no total order on the quality of an ontology in a context. Therefore, it is necessary to consider the search for an optimal ontology for a given context as a multi-objective optimization problem.

The article then defines the optimization problem as follows:

First, $opt'(F(t))$ was denoted to be the optimization problem with function $F(t)_1$ and constraints $F(t)_2$. Let C be the set of contexts, O be the set of ontologies and Q be the set of qualities. Now we can define the optimisation as:

Definition 1. *The function sol is the solution of the context-dependent dynamic multi-objective problem of finding an optimal ontology for a given context* $\Leftrightarrow sol : C \to O$ *and* $sol(c) = opt'(F(c))$

Where $F(C) \to (O \to Q)$ a function which maps the context c to a function which incorporates the context when evaluating the quality of an ontology and its associated domain.

3.3 Cloud Communication System as a Big Data Problem

When the Cloud Communication System is used to handle the complete flow of information in an organization, the amount of data handled becomes such that it will he hard to process with current technology. However, we would like our system to even make sense of all the data it processes. Hence, we arrive in the field of Big Data analytics. According to Fisher et al. [7] "Fundamentally, big data analytics is a workflow that distills terabytes of low-value data (e.g., every tweet) down to, in some cases, a single bit of high-value data (Should Company X acquire Company Y? Can we reject the null hypothesis?). The goal is to see the big picture from the minutia of our digital lives."

Our previous work [5] looked at how to solve the problem of handling streams of tokens arriving at high rate and altogether representing a huge amount of data. The system described in this chapter is a concrete example of such a system. In the following subsections we will, based on the previous work, describe the balance between volume and velocity, how big data can lead to big knowledge, and how a system inspired by the mechanisms of natural evolution could provide a solution.

Volume vs. Velocity Volume, the size of the data, and velocity, the amount of time allowed for its processing are clearly the main factors when talking about Big Data. Both of them manifest themselves when a high number of messages has to be handled within a reasonable time. When the system tries to extract information or knowledge from the data it does this effectively if no important facts are overlooked, i.e the analysis is complete, and the facts found are useful further inference, i.e they are expressive and granular. The ratio of the effort the system spends on finding a given result to its utility can be interpreted as the efficiency of the system. Note that when one tries to improve the effectiveness of the system, the computational complexity will be increased and hence the efficiency of the system might drop. Hence, if we would like to make a deeper analysis of the message stream, we would have a less efficient system.

Big Knowledge When we have a vast amount of data and try to extract all knowledge from it, we might end up with an unmanageable amount. From that observation we identified some aspects which should be taken into account while working with Big Data. We called this approach *3F+3Co* which stands for Focusing, Filtering, and Forgetting + Contextualizing, Compressing and Connecting. It should be noted here that these terms are not novel in the sense that they have been used in different domains and interpretations, see for example [8]. We gave an occasionally overlapping meaning to each of these terms in the context of Big Data analysis as follows.

Focusing is mainly concerned with the order in which the data is processed. An optimal focus will only scan the data which is absolutely needed to come to an answer for the question which is at hand and will hence lead to a higher efficiency. This facet will most likely play a less significant role in the messaging system since the data is arriving continuously and hence the focus will most likely be on the information which freshly arrives to the system.

Filtering is ignoring anything which is, hopefully, not of importance for future analysis. We use *hopefully* since deciding whether information is relevant or not can in most cases not be done with a hundred percent certainty. One way to filter is to only focus on specific features of the data, which also reduces the variety and complexity of the data. Similar to the focusing perspective, it is not possible to make the filter upfront since it is not feasible to accurately guess the future data.

Forgetting is a further step from filtering where data or knowledge derived from it is completely removed from the system. This trashing can remove potentially valuable information. It is very difficult to decide which part of the data can be removed. In the work which we did around Evolutionary Knowledge Systems (see section 3.3), we use the technique of "forgetting before storing ". This means that there has to be reason before anything is stored at all in the knowledge base.

Contextualizing comprises not only the change of sense of statements in different contexts, but also judgments, assessments, attitudes, and sentiments. There are various facets which contribute to the context of data. Examples include the origin of the data, the tools used, and the place in which the result will be used.

Compressing stands for both lossy and lossless compression. Where lossy compression is similar to Forgetting which was discussed above. The lossless compression might be very effective because the high amount of data leads to a high probability that repetitive or periodical patterns are present.

Connecting can be done if information is added to an already existing body of data. The whole body is build incrementally. The benefit of linking the data before processing it further is that data and knowledge mining, knowledge discovery, pattern recognition, etc can be performed more effectively and efficiently. A good example of this connecting used for building an index of the world wide web can be found in [9].

Evolving Knowledge Ecosystems When messages arrive to the CCS, the system tries to forward messages, which is still in abstract form, to the correct receivers over the preferred channel. These includes both inbound and outbound messages. The actual content is, however, likely to evolve over time due to many external factors. Examples include the variation in activity of customers or the company using the system, the economical situation, the season, and so on. To anticipate these changes the CSS should be able to change its inner working, if possible automatically. In the chapter [5] we proposed an Evolving Knowledge Ecosystem which is able to adapt to changes in the environment. This Ecosystem would, when implemented, assist in the understanding of the external world. It should, again, be noted that the proposed system is more general as the parts which could be used in the CCS. In this section, however, the focus will be on the relevant parts.

The core idea behind the Ecosystem is that

> The mechanisms of knowledge evolution are very similar to the mechanisms of biological evolution. Hence, the methods and mechanisms for the evolution of knowledge could be spotted from the ones enabling the evolution of living beings.

Starting from this idea, we derived that we could model the knowledge evolution inside the system using ideas from natural evolution. One of the core ideas is that, similar to the idea of natural selection proposed by Darwin [10], knowledge which is more fit for its environment, has a higher chance to survive as less fit knowledge. The environment here is formed by the incoming information to the system. The following concepts, borrowed from modern evolutionary biology, were further elaborated in relation to the system which processes a stream fo incoming messages. They are also illustrated in Figure 2.

Knowledge organism (KO) are the components which carry all knowledge in the system. They are an analog to living beings in nature.

Environment is the place where the KO reside. The environment is limited in resources and hence only KO which can consume the resources available can survive in the given place of the environment. These places are called environmental contexts.

Knowledge genome is the part of the KO which represents its terminological component, also called Terminological Box or TBox [11].

Knowledge body is the assertional component of the KO. It is similar to a ABox [11] with an ontology from the knowledge genome.

Knowledge tokens are influences from the environment, comparable to mutagens in evolutionary biology. These mutagens come either from analysis of the message streams or are excreted by a KO. They can then subsequently be consumed by a KO if that KO has capability to consume it, i.e. has a knowledge body which has enough similarity to the token.

Morphogenesis is a change in the knowledge body of a KO caused by the consumption of knowledge tokens.

Mutation is a change in the knowledge genome of a KO, which in most cases, leads to a change in the knowledge body.

Recombination is a process in which two or more KOs are combined into a new KO. This entity has a knowledge genome composed of parts of the parents. The newly created KO might be, but is not necessarily, more fit to the environment than any of its parents.

Excretion is a process by which a KO can expose of parts of its knowledge body or genome. These unused parts will be placed in back in the environment.

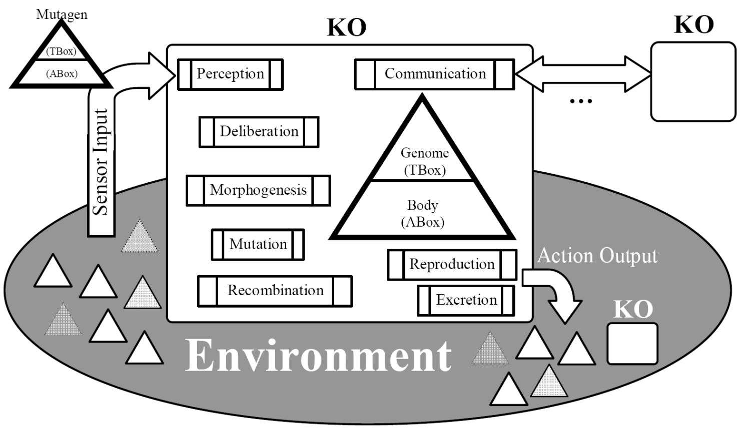


Fig. 2. A Knowledge Organism: functionality and environment. Small triangles of different transparency represent knowledge tokens in the environment – consumed and produced by KOs. These knowledge tokens may also referred to as mutagens as they may trigger mutations. (Picture credit: [5])

4 Human Interaction

As we mentioned above, the *3F*, focusing, filtering, and forgetting, was used earlier in different contexts. One of them is the context of management [8] where the terms are used as techniques to cope with the information overload which many people experience in our modern society. We did not investigate how much information the Cloud Communication System can send to its users before they would get a negative feeling about the system or perhaps about the company using it, which more of a social science topic. (See for instance [12]) We however want to look at how we can reduce the amount of times the user is contacted by the system, leading to a reduction of the load on humans interacting with it.

We looked into several techniques which have been used in other fields before. In the following subsections, these are described in the context of the system we are designing. First, we look from the perspective of Information Filtering and Recommender systems. Then we discuss filtering techniques which can be applied. The overall idea is that the CCS filters content out when it is irrelevant and merges what is related.

4.1 Information Filtering and Recommender Systems

Generally speaking, Information Filtering (IF) is a technique which could be used to automatically remove or add information according to preferences or behaviors. An IF system could be used to perform tasks like the creation of abstracts, the classification of information, and summarization. Also, IF primarily deals with unstructured or semistructured data, its most common use case is the classification of e-mail. [13] An Information Filtering system has a mediator role between the resources and its users.

There are several issues which could be solved using Information Filtering, like for instance finding a good route for messages which pass trough the system, removal of unwanted messages like spam, and the classification of messages based on their meta-data.

Recommender systems are a specific type of Information Filtering systems which can, given a set of items, propose other related items. This type of IF has been proposed in the mid-'90s [14] and is commonly used to offer on-line shoppers suggestions based on their shopping history and what other similar customers bought. Another example is the articles proposed to readers in accordance with the interests set on their profile page. In our system we could use this type of system finding related messages which could be merged together, finding users with similar interest to target messages correctly, and finding users with similar requests or history in order to correctly route a new incoming request in an optimal way. The framework which we proposed in section 3.1 uses semantic storages for all data in the system. Using recommender systems in combination with this kind of data stores has been studies in [15].

Information Filtering and specifically Recommender Systems can use several filtering techniques. They can be roughly divided into Content-based Information Filtering, which takes mainly the content of an individual item into account while filtering, and Collaborative Information Filtering, which looks at similar context for the query, like for instance users with similar interests as the current user. There are also approaches which combine both techniques. A good introduction to the topic can be found in [16]. These two classes of filtering techniques are further elaborated in the next subsections.

4.2 Content-based Information Filtering

Content-based Information Filtering uses mainly attribute of items while searching for recommendations. For example, if a visitor of the NBA.com website has read a number of articles related to the Lakers basketball team, then the system could suggest articles from the database which are tagged with the 'Lakers' tag.

A content-based filtering system often uses a search profile which contains characteristics of the interests, in other words, it tries to relate properties of items with factors found in the search context. In our message merging case, we could use this technology to filter incoming and outgoing message. The incoming messages could be filtered based on similarity with customer profiles in the database and when an outgoing message is sent to a more abstract receiver, we could limit the actual reception to the interested parties only. Most likely, these use cases would also benefit from collaborative filtering described in the next section.

4.3 Collaborative Filtering

Collaborative information filtering offers suggestions according to similarity between users and items. Software like email, calendar and social bookmarking often make use of this type of techniques. [17]

Traditionally, collaborative filtering is used to gather and analyze information from users' behaviors instead of using properties from items, in other words, it is utilized to measure similarity between items by taking other users' preferences and actions in relation to the items into account. [16] For example, collaborative filtering could be explained as follows. If the preferences or characteristics of a single user X are similar to the ones of the members of a user group A, then the system will to recommend items to user X if they have been appreciated, i.e. bought, used, read, and so on, by the members of the group A. In our Cloud Communication Service, the message merging will probably benefit from the use of this technique. We could, for instance, add or propose users to groups and decide what the content of a message is about based on similarity between its sender and other users of the system.

4.4 Knowledge Based Recommendation

Recommender systems are usually classified into categories based on the technique used. Besides the content based and collaborative type introduced above, there is another type, namely, the knowledge based recommender system. This type of recommender system firstly sets up a knowledge foundation which consists of a model of the processed items.

This model can, for instance, consist of users, business items, etc. The system then makes recommendations through reasoning with the data combined with the model. If, for instance, users and items are matched or certain requirements are fulfilled then matching items are proposed.[18] According to [19] Knowledge based recommendation should be preferred in marketing over other recommender systems. Some features of this type of recommender system include

1. Simplicity: a large amount of data is not necessarily required in the knowledge based type.
2. Quickness: new users with a detailed personal profile could receive recommendations at once.
3. Humanity: the system knows what the user needs and why the user needs this item.

A concrete example would be a system which matches users with products which are for sale in a web shop. A new customer fills a short questionnaire during his registration on the website. From these data preferences are extracted and stored in a profile in the database. This step is called *data collection*. Next, there is the *knowledge foundation*. This knowledge foundation will contain the products and their associated tags. The tags are given in accordance with the products' attributes and popularity.

With the Knowledge Foundation and the profile information in place it is possible to generate recommendations for the customer. This recommendation process is often augmented with the visiting record of the customer.

5 Implementation and User Interfaces of the Cloud Communication System

As mentioned in the introduction, the primary goal of the cloud software program is to improve the performance of the Finnish industry. Therefore, in parallel with the efforts to create scientific artifacts related to the business case, we also worked on a concrete implementation of at least part of the ideas. The scientific parts, including the theoretical framework, described above are used as guideline for further extension.

In this section we show some of the user interfaces for channel preference management and multichannel communication coordination which are in use in the application which is under development at Steeri Oy.

An abstract overview of the components of the service are depicted in Figure 3. The framework described in section 3.1 overlaps greatly with the service presented here. A short description of its key elements follows.

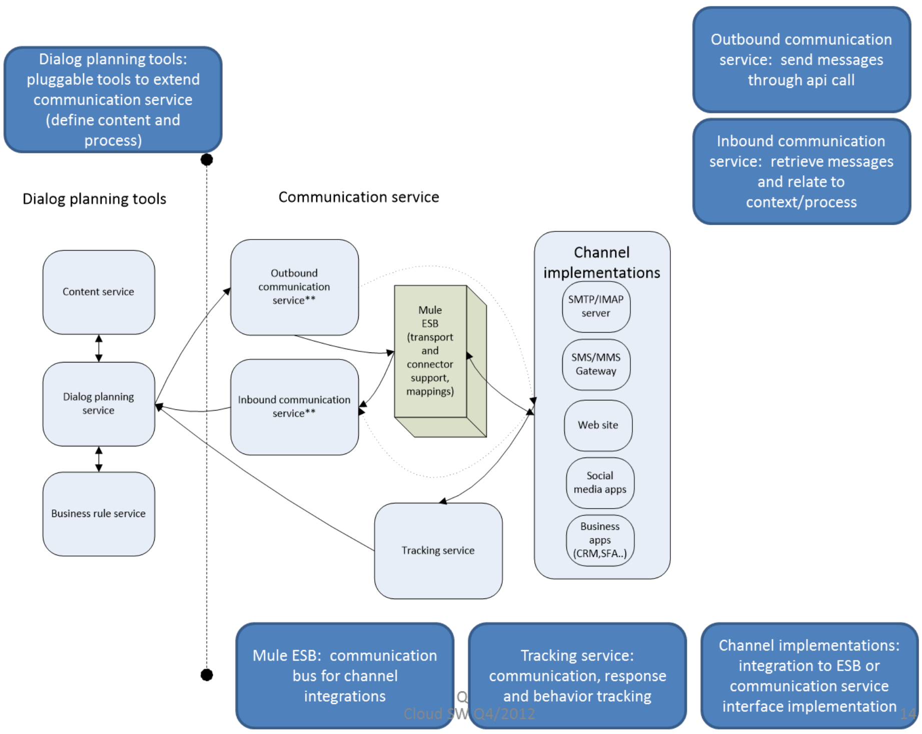


Fig. 3. Overall architecture

The *Outbound communication service* enables dynamic content creation. It is also used to set channel specific tracking mechanisms. In section 4.1 information filtering and recommender systems were discussed. These technologies aim at improving the experience the customer gets from the Outbound communication service.

The *Inbound communication service* maps the incoming communication to the right customers and communication rules. It is in this component where the techniques described in sections 3.3 can be used in the future. The automatic recognition of the meaning of the messages could improve the handling efficiency and its correctness.

The *Mule Enterprise Service Bus (ESB)* [3] and *Channel implementations* are technical tools for implementing the set of interaction channels. Thus, they play a key role in the service, but contain only little or no logic at all from the communication perspective. The logic resides at the level that understands the customer behavior, communication rules and is able to relate the dialog conducted to a specific set of company goals and plans.

To be able to orchestrate the communication with the customer a set of dialog planning and coordination tools has been devised. The key part of these tools is the *Content service*, which enables writing channel independent and specific content blocks. The *Content blocks* relate to specific objectives the company has for its communication. They can, for example, provide offers for customers, inform about new services, or suggest ways of utilizing purchased products. The actual implementation could range from a short text message delivered to the mobile phone to a full story on how to present the information in a phone call. A user interface for the definition of web content for a Content Block is shown in Figure 4.

Using the *Business rule service* the company can define communication plans, i.e. it can create rules about situations in which it wants to take actions and define which customers should be involved. An example of how a business rule could be created can be found from Figure 5. These rules can be applied upon actions and communication from the customer or they can be instigated by goals the company wants to proceed in general.

The *dialog planning service* provides means for defining the multichannel communication process. It connects business rules and content together, and also enables the definition of the steps in the communication process. Each step may have its own rules. An interface for editing

[3] http://www.mulesoft.org/

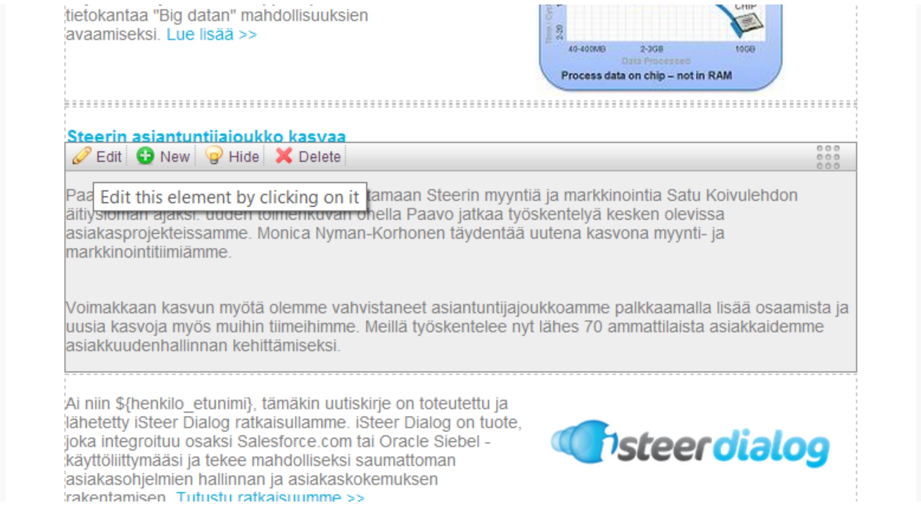


Fig. 4. Defining web content from content blocks

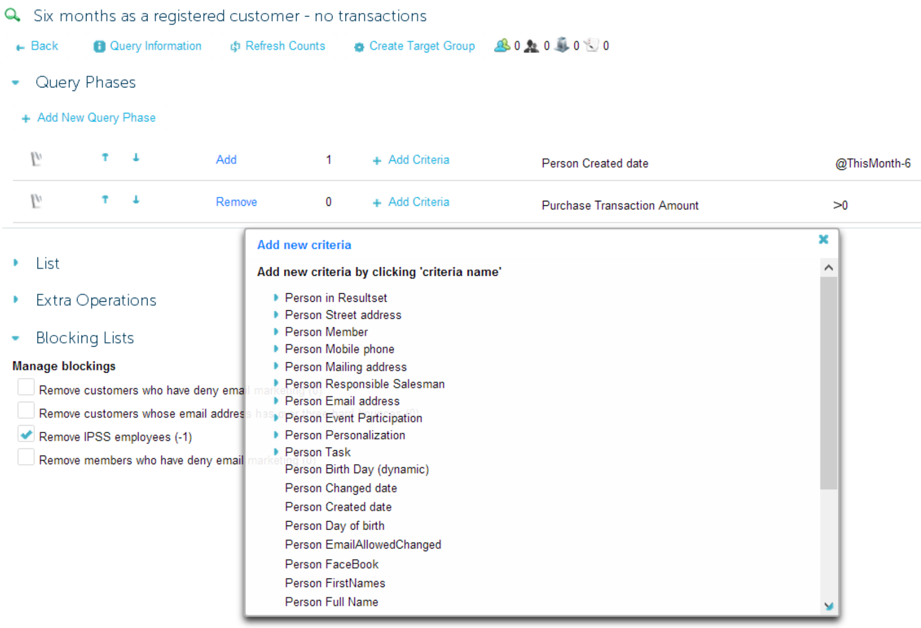
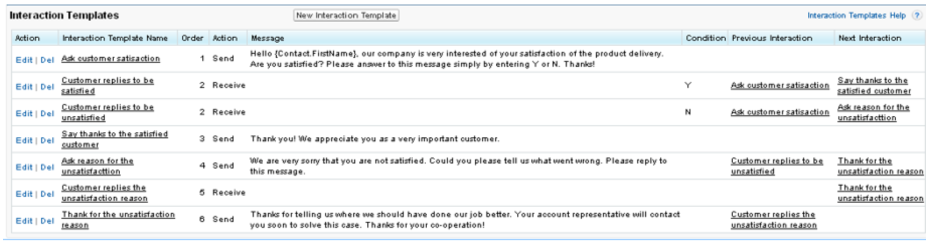


Fig. 5. Interface for the query phase of a business rule. This query consists currently of two phases: In the first phase all persons which have been created 6 months ago (@ThisMonth-6) are added to the result set. In the second phase all persons who have a transaction amount greater as 0 are removed.

the communication steps can be seen in figure 6. Different actions can be performed depending on the communication channel as shown in the screenshot in Figure 7.



Interaction Templates New Interaction Template Interaction Templates Help

Action	Interaction Template Name	Order	Action	Message	Condition	Previous Interaction	Next Interaction
Edit | Del	Ask customer satisaction	1	Send	Hello {Contact.FirstName}, our company is very interested of your satisfaction of the product delivery. Are you satisfied? Please answer to this message simply by entering Y or N. Thanks!			
Edit | Del	Customer replies to be satisfied	2	Receive		Y	Ask customer satisaction	Say thanks to the satisfied customer
Edit | Del	Customer replies to be unsatisfied	2	Receive		N	Ask customer satisaction	Ask reason for the unsatisfacttion
Edit | Del	Say thanks to the satisfied customer	3	Send	Thank you! We appreciate you as a very important customer.			
Edit | Del	Ask reason for the unsatisfacttion	4	Send	We are very sorry that you are not satisfied. Could you please tell us what went wrong. Please reply to this message.		Customer replies to be unsatisfied	Thank for the unsatisfaction reason
Edit | Del	Customer replies the unsatisfaction reason	5	Receive				Thank for the unsatisfaction reason
Edit | Del	Thank for the unsatisfaction reason	6	Send	Thanks for telling us where we should have done our job better. Your account representative will contact you soon to solve this case. Thanks for your co-operation!		Customer replies the unsatisfaction reason	

Fig. 6. An interaction template for different communication steps and conditions

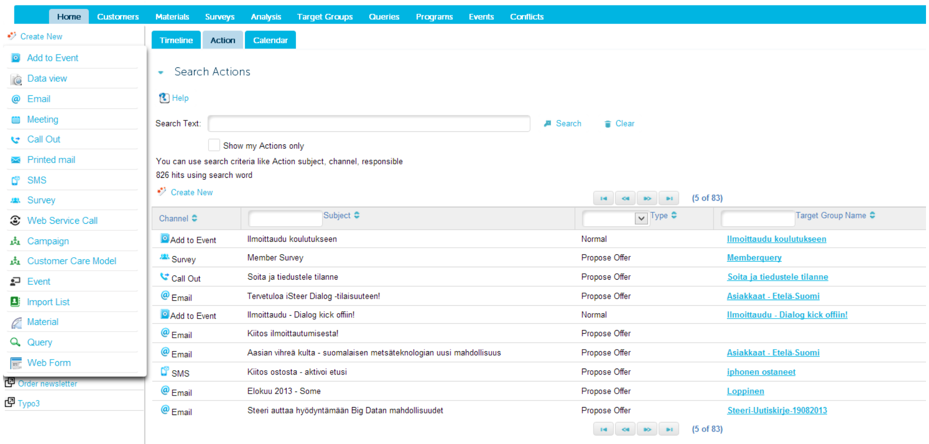


Fig. 7. Defining actions to different channels

The dialog planning service also provides tools for customer preference management and managing communication rules from a specific customers point of view. Managing customer preferences may take place on channel level and based on the type of communication topics one is interested in. The customer is responsible for prioritizing topics of interest and opting in to optional customer care processes. Customers also have the possibility to cancel their subscription to mailings they do no longer which to receive.

Communication plans are run at a scheduled interval. As a result, it is possible to show which customers will receive which information based on their current profile and behavior, as well as the communication rules. As in shown in Figure 8, it is possible to see which planned actions would be targeted to which customers in the future. Furthermore, it is possible to define rules for managing future conflicts like for instance overwhelming the customer with too many messages, possibly handling conflicting offers in different channels, etc. . . This is illustrated in Figure 9. The techniques used here are concrete examples of the methods for filtering and recommendation described in section 4.

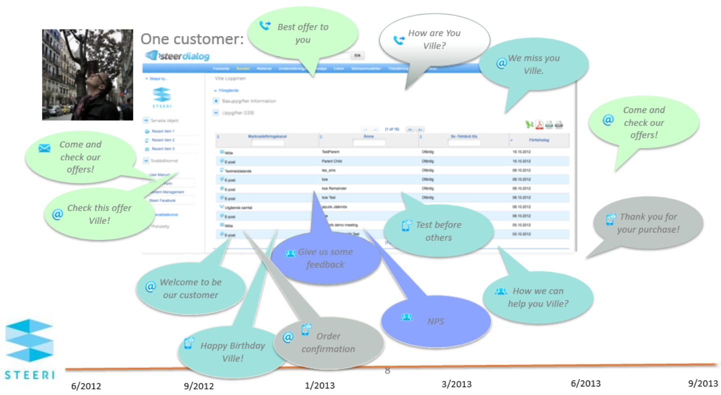


Fig. 8. Future communication view from one customer's perspective

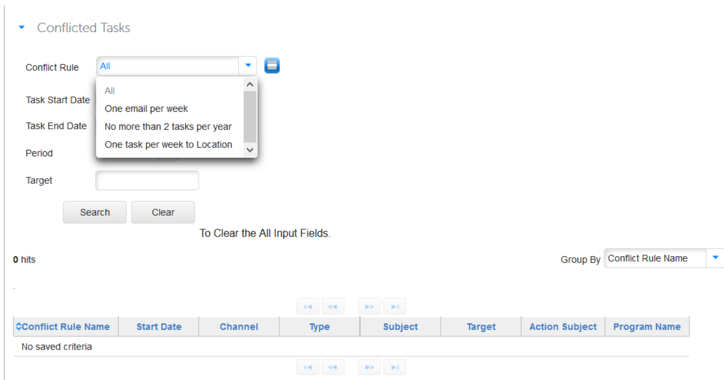


Fig. 9. Conflict manager UI for managing communication hygiene rules

6 Conclusion

Enterprises have often a multitude of systems in place to communicate with internal and external partners, like customers, students, and employees. The communication is usually fragmented over these systems and it is hard to obtain an integrated communication experience. In order to get closer to this target, we proposed the Cloud Communication System (CCS). This system, which is delivered as a SAAS, uses cloud computing technologies and has been elaborated in the frame of the Cloud Software Program.

In this chapter we presented some of our previous research topics. First we showed a general framework for the communication system which uses a semantic data storage to integrate the data which flows trough the system. Then, we discussed how the finding of an ontology with an optimal quality can be defined. This ontology would be stored in the semantic data storage as defined in the framework. Finally we discussed the CCS as a Big Data problem which works on streams of data with a high velocity and propose an Evolving Knowledge Ecosystem to manage the data stream. The knowledge which the system collects from this stream should be kept within feasible limits. We proposed several approaches to limit the amount of stored knowledge which we refer to as *3F+3Co*. This abbreviation stands for by focusing, filtering, and forgetting + Contextualizing, Compressing and Connecting. The core idea behind the Evolving Knowledge Ecosystem is that we can use methods similar to the mechanics of biological evolution for managing knowledge which dynamically changes over time, i.e. which evolves.

From our current research focus we showed our work on human interaction with the system by showing how research in Information Filtering and Recommender Systems can help us to solve specific problems in the CCS. In the last section we showed a selection of user interfaces for the management of communication.

ACKNOWLEDGMENTS This chapter has been written as a part of our activities in the Cloud Software Program organized by the Strategic Centre for Science, Technology and Innovation in the Field of ICT (TiViT Oy) and is financially supported by the Finnish Funding Agency for Technology and Innovation (TEKES). We would further like to thank the technical staff of Steeri Oy for their excellent work in implementing the first prototypes and the members of the Industrial Ontologies Group (IOG) of the university of Jyväskylä for their support in the research. We would also like to thank the reviewer for helpful comments and improvement suggestions.

References

1. Järvinen, J.: Cloud software program. `http://www.cloudsoftwareprogram.org/cloud-software-program` Last accessed on 31 May 2013 (2010)
2. Nagy, M.: On the problem of multi-channel communication. ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer (2012) 128
3. Nagy, M.: A multi-channel communication framework. In: ICT in Education, Research, and Industrial Applications. Springer (2013) 72–88
4. Cochez, M., Terziyan, V.: Quality of an ontology as a dynamic optimisation problem. ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer (2012) 249
5. Ermolayev, V., Akerkar, R., Terziyan, V., Cochez, M.: Towards Evolving Knowledge Ecosystems for Big Data Understanding. [20]
6. Aruna, T., Saranya, K., Bhandari, C.: A survey on ontology evaluation tools. In: Process Automation, Control and Computing (PACC), 2011 International Conference on. (july 2011) 1 –5
7. Fisher, D., DeLine, R., Czerwinski, M., Drucker, S.: Interactions with big data analytics. interactions **19**(3) (May 2012) 50–59
8. Dean, D., Caroline, W.: Recovering from information overload. McKinsey Quarterly (January 2011)
9. Peng, D., Dabek, F.: Large-scale incremental processing using distributed transactions and notifications. In: Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation. (2010)
10. Darwin, C.: On the origins of species by means of natural selection. London: Murray (1859)
11. Nardi, D., Brachman, R.J.: An introduction to description logics. In: The description logic handbook: theory, implementation, and applications. Cambridge: Cambridge University Press (2003) 1–40
12. Bawden, D., Robinson, L.: The dark side of information: overload, anxiety and other paradoxes and pathologies. Journal of information science **35**(2) (2009) 180–191
13. Belkin, N.J., Croft, W.B.: Information filtering and information retrieval: two sides of the same coin? Communications of the ACM **35**(12) (1992) 29–38
14. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Recommender Systems Handbook. Springer (2011) 1–35
15. Ziegler, C.N.: Semantic web recommender systems. In Lindner, W., Mesiti, M., Türker, C., Tzitzikas, Y., Vakali, A., eds.: Current Trends in Database Technology - EDBT 2004 Workshops. Volume 3268 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2005) 78–89
16. Rajaraman, A., Ullman, J.D.: Recommendation Systems. In: Mining of massive datasets. Cambridge University Press (2011) 277–309
17. Bogers, T., Van den Bosch, A.: Collaborative and content-based filtering for item recommendation on social bookmarking websites. Submitted to CIKM **9** (2009)
18. Burke, R.: Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction **12**(4) (2002) 331–370

19. Dell'Aglio, D., Celino, I., Cerizza, D.: Anatomy of a semantic web-enabled knowledge-based recommender system. In: Proceedings of the 4th international workshop Semantic Matchmaking and Resource Retrieval in the Semantic Web, at the 9th International Semantic Web Conference. (2010)
20. Akerkar, R., ed.: Big Data Computing. Taylor & Francis group - Chapman and Hall/CRC (to appear in September 2013)