An Information Systems Reference Architecture for the CRM domain

Summary of dissertation for the degree of Master in Information Systems and Computer Engineering

André Cruz

1Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal
andre.d.cruz@ist.utl.pt

ABSTRACT

The work presented in this article focuses on defining a Reference Application Architecture for the CRM domain and in its application on case studies from the Portuguese Public Administration to assess the benefits and pitfalls of the proposed architecture. The definition of the Reference Architecture is done by extracting the best practices from five CRM solutions: SugarCRM, Microsoft Dynamics, Sage CRM, Oracle Siebel and Salesforce. The CRM Reference Architecture was developed considering the shared functionalities and information entities among these commercial solutions. In the Reference Architecture we identify six modules in the CRM system and five systems, which interact with the CRM system. The six CRM modules are: Account module, Sales module, Marketing module, Service module, Scheduler Module and Administration module. The five interacting systems are: Contact Center, Portal, Document and Knowledge Base Management system, Workflow system and Reporting and Analytics system. We applied the Reference Architecture in two case studies: the High Commissioner for Migration and the Citizen Spaces. We make a comparison between the current state architecture of the case study with an architecture reached from the best practices of the Reference Architecture, using chosen metrics to evaluate information systems. At the end, we take conclusions on the results reached, and say what are the benefits and pitfalls of the proposed architecture.

Keywords

Information Systems Reference Architecture; CRM; modules; systems; best practices; case studies.

1. INTRODUCTION

A great change occurred in the business world in the recent years, the focus of businesses changed from product focus to customer focus. (S. Rezaian Faroie and M.A.S. Monfared, 2009) Nowadays, more and more companies adhere to Customer Relationship Management (CRM) solutions, in order to gain more loyal customers. However to implement a true Customer Relationship Management system, proper architecture is required. (S. Rezaian Faroie and M.A.S. Monfared, 2009)

With this adherence to CRM solutions, a Reference Architecture for this domain is expected to provide a way to approach usual occurring problems by documenting good architectural design practices. (Cloutier, 2010) In this work we present a Reference Application Architecture for the CRM domain, which is applied in two cases of the Portuguese Public Administration provided by the Agency for Administrative Modernization (AMA). In order to reach the Reference Architecture it is necessary to gather the industries best practices. We extracted the best practices from five CRMs known solutions in the market: SugarCRM, Microsoft Dynamics CRM, Sage CRM, Oracle Siebel and Salesforce.

1.1. Thesis Problems

The main problems that we propose to solve with this work are: if it is relevant to define a Reference Application Architecture for the CRM domain, considering industry best practices extracted from commercial solutions, if Reference Architectures for CRM are useful for defining specific Enterprise Architectures for the CRM domain and if is the Reference Architecture adequate to be the Reference Architecture for the CRM domain for the Public Portuguese Administration. These problems bring with them a set of other smaller problems, regarding how to define the Reference Architecture, that need to be answer. Those problems are: what are the main features of the CRM solutions, what are the main information entities of the CRM solutions, what patterns compose the Reference Architecture for the Customer Relationship Management domain, what are the main Enterprise Architecture Principles in CRM domain. These are the problems to whom we propose a solution.
1.2. Research Methodology and Document Structure

To accomplish the proposed work, we follow the Action Research Methodology from (Baskerville, 1999). This methodology is composed by five steps and the document is structured in accordance to those steps. The first step is where the problem is defined, to draw a scenario of what is to be done. This step is present in the first section, the Introduction section, when we define the problems that we propose to solve. The second step of the methodology corresponds to the gathering and organization of information about the problem, in order to create a theoretical and practical basis. This step is present in the Related Work section. The third step is the creation of the solution with the information from step two. This step corresponds to the section three, the Architecture Solution section. The fourth step is where we validate the reached solution with a case study. This step is accomplished in the Case Studies section, in the fourth section. Finally, the fifth step corresponds to the evaluation, analysis and conclusions taken in those case studies. This step is present in the fourth and fifth section of our document.

2. RELATED WORK

In this section are presented the theoretical concepts required for the development of the solution and its evaluation.

2.1. Enterprise Architecture

Enterprise Architecture (EA) is an instrument to define the future direction of the enterprise, and also the mechanism that coordinates the actual transformation of the enterprise. Mark Lankhorst defines EA objective by stating: “Enterprise architecture tries to describe and control an organization’s structure, processes, applications, systems and techniques in an integrated way.” (Lankhorst, 2005)

2.1.1. Enterprise Architecture Framework

An Enterprise Architecture Framework is as stated by (Lankhorst, 2005): “a conceptual structure of what an EA should contain and how to create it, i.e. models, principles, approaches, standards that guide the development of enterprise architectures”.

In this work, for the representation of our architecture and to represent the case studies, we use the ArchiMate notation. (Haren, 2012) We choose ArchiMate, because it offers in a detailed and comprehensive way the representation of the application domain, and its relation with the business layer and the data domain.

2.1.2. Reference Enterprise Architecture

A Reference Enterprise Architecture is a way to approach usual occurring problems by documenting good architectural design practices. (Cloutier, 2010) We need Reference Architectures, because they improve effectiveness through: managing synergy, providing guidance, providing an architectural baseline and blueprint and by capturing and sharing architectural patterns. (Hole, 2007)

2.1.3. Methodology for Specific Architectures Generation

To reach a specific architecture from our Reference Architecture, we use a methodology from (Bauer, 2012). This methodology consists on having the requirements from the properties of the desired system, and using the Reference Architecture for guidance, by providing best practices, patterns and an architectural blueprint. Also used in this methodology in the guidance of those requirements are the engineering strategies. In our case those strategies are the ArchiMate and the CRUD matrix.

2.1.4. Enterprise Architecture Principles

Enterprise Architecture Principles are as defined by TOGAF: “general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission.” (The Open Group, 2009)

We use to make our EA Principles selection, the principles defined by (Greefhorst, 2011). We analyse the list of principles and choose the ones that are satisfied by our Reference Architecture. Those principles are going to be considered the main EA Principles to follow when developing an Enterprise Architecture for the CRM domain.

2.1.5. Metrics and Heuristics for Information Systems evaluation

For the evaluation of our work, we have to have some metrics to assess the benefits and pitfalls of our architecture in relation to what is in place. We decided to use the metrics and heuristics from (Sousa, 2005) (André Vasconcelos C. M., 2005) (Sousa, 2003), to measure the alignment quality attribute in three vectors: the alignment between Business Architecture and Information Architecture, the alignment between Business...
Architecture and Application Architecture and the alignment between Information Architecture and Application Architecture. We also used other set of metrics, the metrics list from (André Vasconcelos P. S., 2008). From that list we chosen the following three metrics: the Average Number of Operations in «IS Blocks», the Response for a Service Factor and the Lack of Cohesion in «IS Block» Factor.

2.2. Customer Relationship Management

Kenneth C. Laudon and Jane P. Laudon defined CRM systems as a way to: "capture and integrate customer data from all over the organization, consolidate the data, analyze the data, and then distribute the results to various systems and customer touch points across the enterprise." (Laudon, 2012)

The CRM software companies provide solutions for three major areas: sales force automation, marketing automation and customer service. The CRMs have also three major technologies: collaborative technologies, operational technologies and analytical technologies. (Laudon, 2012)

2.2.1. Customer Relationship Management Features

To specify the functionalities of our CRM Reference Architecture, we extracted the features of five known CRM solutions and made a comparison on what features were common between them. The five CRM solutions are: SugarCRM, Microsoft Dynamics, Sage CRM, Oracle Siebel and Salesforce. The common features between them were the features chosen to be the functions of our Reference Application Architecture. We grouped the features in seven groups, in accordance with the CRM solutions datasheets. The extracted features are:

- Sales features: account management, activity management, approvals, competitor tracking, contact management, contract management, sales literature, lead management, opportunity management, product management, quote management, sales forecasting, territory management, order management, quota management and sales pipeline;
- Marketing features: campaign management and execution, email marketing, newsletter management, marketing campaigns, list management, lead management and web to lead capture;
- Customer Service features: case escalation and notification, case routing and queuing, case management, contact center, customer self-service portal, email management, knowledge base, customer information and service contracts;
- Reporting features: custom reports, dashboards, sales analytics, marketing analytics and service analytics;
- Integration features: email integration, social networks, integrated third-party apps, web services API – SOAP, web services API – REST, computer telephone integration, automatic call distributor, Microsoft office integration and cloud connectors;
- Security features: role based security, advanced password management, audit trail, field level security, user based security and team based security;

2.2.2. Customer Relationship Management Information Entities

After the identified features, we extract the common information entities from the data models of the five CRM solutions chosen. For the identification, we did a mapping between the identified information entities of each CRM solution, to reach the common information entities. In this mapping of information entities we also used the (Buttle, 2009) reference for the identification of the information entities and their relations.

The common information entities identified are: customer, account, organization, person, partner, contact, sales activity, competitor, competitor product, contract, lead, prospect, opportunity, quote, forecast, quota, territory, customer product, order, invoice, product, price, product catalogue, campaign activity, campaign, campaign wave marketing funds, marketing segment, marketing budget, marketing plan, campaign response, marketing list, service activity, case, case solution, email, call, communication, portal, user, group, sales team, service team, scheduler, calendar, report, dashboard, workflow, document, knowledge article, person address, order item, user role, user login, user contact. In some cases of these information entities, in order to simplify, we cluster and

1 http://www.salesforce.com/us/developer/docs/api/Content/data_model.htm
3 http://creately.com/diagram/example/hhbgpyd91/New+Sage+CRM+Data+Model

divide some them: the campaign activity aggregates the campaign activity and campaign wave; opportunity aggregates opportunity and opportunity item; order aggregates order and order item; person aggregates person and person address; user aggregates user, user role, user login and user contact; contract is divided in two specialized information entities the sales contract and the service contract.

3. ARCHITECTURE SOLUTION

In this section we present the Reference Architecture solution proposed. To reach this CRM Reference Architecture we use a methodology. The methodology has three steps. In the first step we define the mission, vision and strategy of the Reference Architecture, which are:

- Mission: provide guidance, knowledge, architectural blueprints and improvement in CRM domain;
- Vision: be the most complete Reference Application Architecture in CRM domain;
- Strategy: extract best practices regarding CRM domain, define the Reference Architecture based on those best practices and evaluate the Reference Architecture in case studies, in order to improve it;

The second step of the methodology corresponds to the representation of the common features identified in section 2.2.1 in application architecture, becoming the first view of our Reference Architecture (Cruz, 2015). The reached Reference Architecture is composed by ten identified CRM modules, which are: Sales module, Marketing module, Service module, Calendar module, Reporting module, Security module, Mobile module, Document Management module, Integration module and Workflow module. Each of the modules has a set of features chosen from the features identified in section 2.2.1. The features common to at least three of the CRM solutions, were the ones chosen to be part of the Reference Architecture.

The third step is where we reach a complete Reference Architecture. To reach the Reference Architecture we use the previous identified functionalities, with exception to the Integration features, because we considered that the Integration features are more regarding the Technological domain than the Application domain. Those functionalities are going to be mapped in a CRUD matrix with the common information entities, identified in section 2.2.2. Those information entities allowed also the definition of a Reference Information Architecture. With the CRUD matrix we reach a Reference Application Architecture, illustrated in Figure 1.

![Figure 1 - Reference Application Architecture](http://pt.slideshare.net/PhilipJung/siebel-data-model-reference-82)
The reached Reference Application Architecture is composed by six modules: account module, sales module, marketing module, service module, scheduler module and administration module. The Reference Architecture is composed also by five systems, which interact with the CRM system: portal system, contact center, document and knowledge base management, reporting and analytics system and workflow management system.

Each module and system has a set of functionalities which are:

- **Account module:** contains account management, contact management and customer information functions;
- **Sales module:** contains activity management, sales pipeline management, competitor tracking, contracts management, lead management, web to lead capture, opportunity management, quote management, sales forecasting, quota management, territory management, order management and product and catalog management functions;
- **Marketing module:** contains campaign management, campaign execution, email marketing, marketing campaigns, newsletter management and marketing list management functions;
- **Service module:** contains service contracts, case escalation and notification, case routing and queuing, case management, email management functions;
- **Administration module:** contains user authentication, team authentication, role authentication and field permissions functions;
- **Scheduler module:** contains calendar management function;
- **Portal system:** contains customer self-service portal, mobile and offline access functions;
- **Contact Center system:** contains contact center function;
- **Document and Knowledge Management system:** contains sales literature, document management and knowledge management functions;
- **Workflow Management system:** contains workflow processes automation management function;
- **Reporting and Analytics system:** contains dashboards, reports, sales analytics, service analytics and marketing analytics functions;

The final part of the work regarding the proposed Reference Architecture solution is the identification of the Enterprise Architecture Principles that are satisfied by it. We identified eleven principles that are satisfied in the Reference Architecture, which are: customers have a single point of contact, processes are standardized, front-office processes are separated from back-office processes, the status of customer requests is readily available inside and outside the organization, documents are stored in the document management system, reporting and analytical applications do not use the operational environment, applications are modular, application functionality is available through an enterprise portal, processes are supported by a business process management system, authorizations are role-based, access to it systems is authenticated and authorized. (Greefhorst, 2011)

4. **EVALUATION**

In this section we present the evaluation done in the case studies. We start by explaining the methodology followed for evaluating each case. The methodology consists in two steps:

- **Step 1:** we analyze and extract from the provided documents of each case, the business processes, the information entities, the CRM modules and the systems that interact with the CRM system. From this information we reach an Application Architecture for the current state of the case study. Then with the same information extracted from the documents, we verify how our Reference Architecture satisfies them and we reach a specific architecture for the case study based on our Reference Architecture.
- **Step 2:** we apply the chosen metrics and heuristics for information systems evaluation, present in section 2.1.5, in the architecture of the current state of the case study and in the architecture reached from our Reference Architecture. In the end we take conclusions on the results obtained.

In this article we only present one of the case studies evaluated.

4.1. **Case Study**

In this section we present one of the case studies where we evaluate our Reference Architecture. All the cases were provided by the Agency for Administrative Modernization (AMA). It is important to refer that the cases which we evaluate are all in the public sector domain, despite our Reference Architecture being based on CRM solutions for mid-market companies. If in the case studies are identified some patterns that aren’t covered by the Reference Architecture, those patterns may be added to the Reference Architecture in the end of the evaluation, in order to the Reference Architecture in terms of the public sector domain.
4.1.1. High Commissioner for the Migration (ACM)

The ACM is a public institute with the goal of cooperating in the definition, execution and evaluation of public politics, transverse and sectorial policies on migration, relevant to the attraction of migrants in national and international contexts. The ACM is also responsible for the integration of immigrants and ethnic groups, in particular Roma communities, and for managing and valuing diversity among cultures, ethnicities and religions. We experienced the CRM from ACM (Ruivo, 2012), and from that we extracted the following business processes regarding CRM domain:

- Customer Management Processes: create customer form, associate customer, edit customer, list customers;
- Processes Management Processes: create process, edit process, list processes, close process, write a note;
- Tasks Management Processes: create task, edit task, list tasks, write a note;
- Follow-Up Management Processes: edit follow-up, list follow-ups, do a follow-up on a process;
- Service Management Processes: begin service, close service, resume service, list services;
- Scheduler Management Processes: create schedule, edit schedule, assign ticket, list schedules, call customer;
- User Management Processes: user sign up, user login, user profile, password recovery, visualize notifications;
- Administration Management Processes: list users, create new user, user profile, edit user, list teams, create team, team profile, edit team, list profiles, create new profile, profile of profile, edit profile, list modules, module profile, edit module;

After the identification of the business processes, we identify the information entities used in this CRM, which are: note, process, task, request, scheduler, ticket, document, follow-up, customer, user, report, permission, profile and team.

We also identified the main modules of this CRM and the interacting systems, and with all these information we can represent the CRUD matrix, and through it reach the Application Architecture, illustrated in Figure 2.

![Figure 2 - ACM Application Architecture](image-url)
In the Application Architecture of the ACM CRM, we identified eight modules and four systems which interacted with the CRM:

- Processes Module: provides processes management functionalities;
- Follow-Up Module: provides follow-up management functionalities;
- Task Module: provides tasks management functionalities;
- Client Module: provides client account management functionalities;
- Customer Service Module: provides customer service management functionalities;
- Scheduler Module: provides scheduler management functionalities;
- User Module: provides users management functionalities;
- Administration Module: provides administration management functionalities;
- Document Management System: provides document management functionalities;
- Reporting System: provides reporting management functionalities;
- SIGA System: provides ticket management functionalities;
- Portal System: front-end platform that communicates with CRM API’s and provides an easier interface. This system wasn’t identified in the CRUD matrix, instead it was identified through the contact with AMA responsible;

We arrived at the Application Architecture of the ACM CRM, so is time to reach a specific architecture based on our Reference Architecture. We reach an Application Architecture with the Reference Architecture guidance, illustrated in Figure 3. The changes made because of the best practices are:

- We grouped the Request, Follow Up, Process and Task modules into one module, the Service module, because the Service module in our architecture has the case management functions, which includes the functions of these four module;
- The User and Administration modules correspond to a single module in our architecture, the User/Administration module;
- The Client module in our architecture goes by the name of Account module;
- The system that wasn't covered by our Reference Architecture stayed the same. In this case is the SIGA system;

![Diagram](Figure 3 - ACM Application Architecture from Reference Architecture)
In the Application Architecture for the ACM CRM case based on our Reference Architecture, we identify four modules of the CRM System and four systems that interact with CRM System:

- Account Module: provides client account management functionalities;
- Service Module: provides customer service management, processes management, follow-up management and the tasks management functionalities;
- Scheduler Module: provides scheduler management functionalities;
- Reporting and Analytics System: provides reporting functionalities;
- User/Administration Module: provides administration and user management functionalities;
- Document and Knowledge Management System: provides document management functionalities;
- SIGA System: stays the same;
- Portal System: stays the same;

The final step is to evaluate both Application architectures with the metrics and heuristics to evaluate information systems chosen in section 2.1.5. The results obtained are present in the Table 1.

<table>
<thead>
<tr>
<th>Evaluated Qualities Criteria</th>
<th>ACM Current State Architecture</th>
<th>ACM Architecture by Ref. Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Facility (Application)</td>
<td>0,305</td>
<td>0,194</td>
</tr>
<tr>
<td>Test Facility</td>
<td>0,207</td>
<td>0,235</td>
</tr>
<tr>
<td>Change Facility (Information/Application)</td>
<td>0,9935</td>
<td>0,9897</td>
</tr>
<tr>
<td>Alignment between Business Architecture and Information Architecture</td>
<td>0,666 (66,6%)</td>
<td>0,666 (66,6%)</td>
</tr>
<tr>
<td>Alignment between Business Architecture and Information Systems Architecture</td>
<td>1 (100%)</td>
<td>1 (100%)</td>
</tr>
<tr>
<td>Alignment between Information Architecture and Information Systems Architecture</td>
<td>0,714 (71,4%)</td>
<td>0,857 (85,7%)</td>
</tr>
</tbody>
</table>

From the Table 1, that presents the results of the evaluation done on ACM case, we can conclude that the solution from the Reference Architecture has a better alignment between Information Architecture and Information Systems Architecture and a better test facility, but it has worst change facility. The test facility result is better because the solution from Reference Architecture uses less CRM modules. The change facility is worse because of that same factor, the current state of ACM architecture has more CRM modules, so is easier to change than an architecture with less modules. Regarding the better alignment between Information Architecture and Information Systems Architecture, this happens because in the ACM current architecture there are various modules that manage the same entities, and in the solution from the Reference Architecture this doesn't happen.

Observation

In this subsection we identified the limitations of our solution in this case. In this case study two information entities weren’t covered by our Reference Architecture, the note and ticket information entities. The process and follow up entities correspond to the case information entity in our Reference Architecture. In relation to the interacting systems our Reference Architecture also didn't took into account the SIGA system for ticket management. Regarding ticket management our Reference Architecture only took in account that the CRM manage the cases and route those same cases.

4.2. Case Studies Pattern

After the evaluation done, we identified a pattern from Portuguese Public Administration regarding CRM present in both cases evaluated that wasn't covered by our Reference Architecture. The pattern is regarding the SIGA system, a system responsible for ticket management, which appears in both case studies interacting with CRM.
The interaction between that system and the CRM, involves in the two case studies two different CRM modules. In the first case study (the case demonstrated in this article) the SIGA system interacts with the Scheduler module and in the second case study interacts with the Service module. From this observation we can represent a pattern regarding the CRM domain in Portuguese Public Administration that wasn't satisfied by our Reference Architecture and that can be added to it for improvement in this sector. The pattern is illustrated in Figure 4, and we named it the SIGA pattern.

![Figure 4 - SIGA pattern](image)

5. CONCLUSION

In this section is where the main conclusions of the work done are taken. Where are explained the achievements reached with this work, the principal contributions, the limitations of the solution and thoughts regarding what can be done for future work.

5.1. Major Contributions

The whole work done in this thesis was in order to provide a Reference Architecture for the CRM domain. Therefore the main contribution is the proposal of a Reference Application Architecture for the CRM domain. This Reference Architecture was defined based on the analysis done on five CRM commercial solutions. Through that analysis, we identified six CRM modules and five systems that interact with the CRM system. The six CRM modules are: Account module, Sales module, Marketing module, Service module, Administration module and Scheduler module. The five systems that interact with the CRM are: Portal, Contact Center, Document and Knowledge Base Management system, Workflow Management system and Reporting and Analysis system. In order to construct this Reference Architecture, we used a methodology to reach this same Reference Architecture. This methodology is a cyclical work, with three steps but more steps can be added to it in future work. Those three steps are: the definition of the mission, vision and strategy, the definition of a first view of the Reference Architecture using only the CRM features and the definition of the final view of the Reference Architecture using the features and information entities mapped in a CRUD matrix.

Regarding the evaluation, we provide other contribution on the benefits and pitfalls of the proposed solution architecture. We evaluated our Reference Architecture in two case studies from Portuguese Public Administration: the High Commissioner for Migration and the Citizen Spaces. For each of these cases we applied a methodology for the evaluation of the specific quality attributes. This evaluation methodology, which we defined consists in two steps: in the first step we analyse the documents from the case study and with the information extracted from these documents we can define the current state Application Architecture and also, define an Application Architecture specific for the case study based on the best practices from our Reference Architecture. In the second step we apply the chosen metrics for information systems evaluation in both architectures and we make an analysis on the results obtained.

For each case we obtained a better test facility, a worst change facility and for the High Commissioner for Migration case we obtained also a better alignment between information and systems information. The reason for these results is due the fact that the solutions derived from our Reference Architecture have fewer CRM modules and also less systems that interact with the CRM, in relation to the current architectures of the provided case studies. These aspects result in a better test facility, because it requires less modules and systems to test, and a worst change facility because the more the processes are divided in various systems and modules, the easier is to make a change.

Also in the evaluation done we provide other contribution. We have identified a pattern from the CRM domain from the Portuguese Public Administration. The pattern consisted of the interaction between the CRM system and a system responsible for ticket management, the SIGA system. We named this pattern the SIGA pattern that
is present in the two case studies used in this work. In conclusion, these were the major contributions in this thesis.

5.2. Limitations

In this section we present the limitations of the architectural solution. One limitation identified at the evaluation was the fact that our Reference Architecture was developed from five CRM mid-market suites and not CRM commercial solutions specialized for public sector. This limitation implied that during the evaluation some aspects of the case studies weren't covered by our Reference Architecture. The limitation referred above raised some limitations during the evaluation.

We identified that our Reference Architecture didn't cover the following aspects in the case studies: the SIGA system present in two of the case studies; the EasyVista CRM, the Active Directory and IPContactos systems present in Citizen Spaces case study; a set of information entities in each case study and the Cashier CRM module in Citizen Spaces. Another limitation is the few number of case studies where the Reference Architecture is evaluated. With more case studies, more validated would be the solution. In the next section we explain what can be done in future work.

5.3. Future Work

For future work the first idea is to continue to evaluate the Reference Architecture in more case studies from the Portuguese Public Administration, in particular, the case from Social Security, which is a complex case study that we couldn't get access in time.

The goal with the continuous evaluation in case studies, is to improve the Reference Architecture so that it can be more adapted to the Portuguese public sector. In the two case studies evaluated, we made an observation at the end of each of them to analyse what requirements weren't covered by our Reference Architecture. If in these observations a pattern was found, that pattern would be added to the Reference Architecture. With a greater number of case studies and their observations, more patterns would be found and would be added to the Reference Architecture. This would result in a Reference Architecture more adequate to the Portuguese public sector.

Another idea for future work is the definition of a Reference Architecture for the CRM domain, specialized in evaluating architectures in CRM domain. The goal with definition of a Reference Architecture as described, is to provide grades to Enterprise Architectures in the CRM domain, to verify if these architectures follow the best practices in the CRM domain. Note that the Reference Architecture in this work intended to be a guide for the definition of Enterprise Architectures for specific cases in the CRM domain and it isn't focused on evaluating other architectures. The evaluation done in this work is to prove that our Reference Architecture based on CRM commercial solutions offer benefits and pitfalls in relation to other architectures implemented on the field and that satisfies the requirements of those cases. These are the ideas for the future work regarding the theme approached in this thesis.

References


