

CSI 5112. WINTER 2008. DR. DANIEL AMYOT.

BORLAND TOGETHER FOR ECLIPSE & RATIONAL SOFTWARE ARCHITECT

A COMPARISON OF REQUIREMENT INTEGRATION SUPPORT FROM AN MDA PERSPECTIVE

OMAR BAHY BADREDDIN.

Executive Summary

In our efforts to enhance our organization's MDA capability, we have undergone a relentless evaluation of modeling tools available in the market today. This document reports our extensive experimentation to evaluate the two short listed tools: Borland Together for Eclipse and IBM Rational Software Architect.

Our evaluation is focused on integration with requirements throughout a typical model centric project. Specifically, we evaluated the tools integration support with requirements during a number of model transformations. We rely on Gartner's report for an overall evaluation of the tool ability to execute and completeness of vision. In addition, we evaluated the tools according to a number of supporting features.

Our experimentations indicate that IBM Rational Software Architect has a more complete and consistent support for requirement integration. Our findings are in alignment with Gartner's report. Based on our findings and evaluation, we have made the decision to recommend the adoption of IBM Rational Software Architect to our organization.

Revision History

	Version	Description	Summary of changes	Date
1	V 1.0	Initial Draft	Created outline	03-07-2008
2	V 1.1	Revised Draft	Experiment and results documented	03-21-2008
3	V1.2	Revised Draft	Revised experiment results	03-28-2008
4	V2.0	Final Document	Document ready for final review	04-04-2008
5	V2.1	Final Document	Final reviews comments implemented	12-04-2008
6	V2.2	Final Document	Peer reviewed	17-04-2008
7	V2.3	Final Document	Peer review comments implemented	19-04-2008
8				
9				
10				

Tables of Contents

- Executive Summary..... 2
- Revision History 3
- Tables of Contents 4
- List of Figures 5
- List of Tables 5
- Glossary of Terms..... 6
- 1. Introduction 7
 - 1.1 How to read this report 7
 - 1.2 Business Context: 8
 - 1.3 Background 9
- 2 Evaluation Criteria: 12
 - 2.1 Integration with Requirement 13
 - 2.2 CIM to PIM support..... 14
 - 2.3 PIM to PSM support 15
 - 2.4 Soft Criteria 15
 - 2.5 Evaluation Methodology..... 16
- 3 Experiment..... 17
 - 3.1 Experiment Prerequisites..... 17
 - 3.2 Scenario..... 17
- 4 Evaluation of the Candidate tools..... 18
 - 4.1 Integration with Requirement 18
 - 4.2 Overall Evaluation 28
- 5 Conclusion and Remarks..... 30
- 6 References 31

List of Figures

FIGURE 1: BORLAND INTEGRATION VIEW	10
FIGURE 2: RSA REQUIREMENT INTEGRATION VIEW	11
FIGURE 3: EVALUATION CRITERIA	16
FIGURE 4: EXPERIMENT INPUT	18
FIGURE 5: NO REQUIREMENT ERROR MESSAGE	20
FIGURE 6: GARTNER'S MAGIC QUADRANT [4]	27

List of Tables

TABLE 1: EVALUATION MEASURES	17
TABLE 2: REQUIREMENT LINKS TYPES	19
TABLE 3: REQUIREMENT MODEL NAVIGATION AND EDITING	21
TABLE 4: MODEL ELEMENTS AND REQUIREMENT CREATION	22
TABLE 5: REQUIREMENT INTEGRATION PREFERENCES	22
TABLE 6: SUPPORTED REQUIREMENT VIEWS	23
TABLE 7: IMPACT ANALYSIS	24
TABLE 8: REQUIREMENT LINKS PORTABILITY (BPM TO USE CASE MODEL)	25
TABLE 9: REQUIREMENT LINKS PORTABILITY (BPEL MODEL)	26
TABLE 10: THIRD PARTY EVALUATION	27
TABLE 11: INTEGRATION WITH SOFTWARE DEVELOPMENT PROCESS	27
TABLE 12: MARKET SHARE AND FINANCIAL FACTORS	28
TABLE 13: HARDWARE AND PLATFORM SUPPORT	28
TABLE 14: OVERALL EVALUATION	28

Glossary of Terms

BPEL	Business Process Execution Language
CIM	Computation Independent Model
MDA	Model Driven Architecture
MDD	Model Driven Development
PIM	Platform Independent Model
PSM	Platform Independent Model
Requirement Link	Associate a requirement with a domain element, or a domain element with a requirement

1. Introduction

Like many small and medium size organizations, we are faced with numerous challenges. Our success depends on taking the right decisions to enhance our competitiveness and efficiency. Our organization, BPM Inc., has recently launched a Model Driven Architecture (MDA) initiative. The initiative stems from the growing adoption of model centric software development approaches in the industry as well as the potential benefits that MDA promises.

Stepping on unfamiliar grounds always comes with great challenges and risk. It is therefore of utmost importance that our organization makes every effort to take the right decisions. MDA adoption is a journey, and we expect the initiative execution to last over an extended period of time.

MDA will have a broad and lasting effect on our organization. Therefore, our organization has decided to assess every major decision against our MDA initiative.

Recently, our team has been assigned the task of evaluating the architecture tools available in the market. Borland Together for Eclipse and IBM Rational Software Architect were shortlisted for extensive experimentation and evaluation.

We have designed an experiment to evaluate the two candidate tools based on their suitability under our MDA initiative. This report specifically focuses on the tool handling of integration with Requirement Management tools. Since Model transformation is at the heart of MDA, our assessment of the tools takes into consideration the tool handling of requirement integration when model transformations are executed.

Both Borland Together for Eclipse and IBM Rational Software Architect are widely adopted tools and are well established in the market. For this reason, and because our experimentation and assessment are quite focused on requirement integration capabilities, we rely on Gartner's assessment for an overall evaluation of the candidate tools. We also take into consideration an evaluation of a number of supporting features that are of specific interest to our organization.

This report outlines our business context, followed by a brief background on Borland and IBM. We then specify our evaluation criteria against which each tool is tested and evaluated. Our experiment, the results and findings of each feature, as well as the numerical evaluation of the candidate tools are also presented. We finally conclude by stating general remarks and our recommendation.

1.1 How to read this report

This report is organized so that it can be read in a number of ways. This section summarizes the organization of the report, and guides you to efficiently utilize the presented information.

The main five sections of this report are:

1. Introduction
This section introduces the report and the business context of the evaluation.
2. Evaluation Criteria
We present in this section our evaluation core and soft criteria. This section defines what we mean by each criterion and defines our evaluation specifics.
3. Experiment
We present our experiment in this section. The experiment phases are executed using the candidate tools.
4. Evaluation of the candidate tools
This section summarizes our findings and presents the evaluation for each feature.
5. Conclusion and remarks
We conclude by summarizing the evaluation and stating our recommendation.

We recommend a thorough reading to promote a complete comprehension of the content of this report. This is specifically recommended if the reader is not familiar with Software Modeling tools, Model Driven Architecture, and Requirement Management disciplines.

For technical readers who are specifically interested in the evaluation of Borland Together for Eclipse and IBM Rational Software Architect, we recommend the reading of the Evaluation Criteria before reading the Evaluation of the candidate tools. The reader can also jump to a specific evaluation criterion, followed by the results of the evaluation.

1.2 Business Context:

BPM provides solution services centered on Business Process Consultancy. A large portion of their clients come from the banking and insurance industries. BPM has 200 employees, of which there are 20 Project Managers, 45 Business Process Analysts, and 80 Developers and Testers. They are faced with the following challenges:

- High levels of schedule and budget variances.
- Developers and clients are geographically distributed.
- Loosely defined Development Process.
- High rate of requirement changes.

BPM Success factors can be summarized in four points:

- Enhance communications amongst stakeholders.
- Enhance the ability to rapidly adapt to changes in the requirements.
- Improve reusability.
- Better alignment with business strategies and objectives.

BPM realizes the potential that lies with adopting more agile software engineering practices. BPM has taken the decision to adopt Model Driven Architecture initiative. They understand the value and the potential behind this adoption. In addition, they also understand the adoption will take time and effort. Decisions have to be made taking into considerations BPM's long term objectives.

As BPM evaluates the available architecture and modeling tools, they also evaluate how the tools fit within their broader set of tools, as well as the future technologies they may need to acquire on the path to more MDA alignment.

Model Driven Architecture is a broad initiative with relatively large number of supporting tools available in the market today. BPM's short list includes IBM Rational Software Architect and Borland Together for Eclipse as candidate architecture tools. BPM is evaluating the tools for their specific needs of MDA initiative, focusing on requirement integration capabilities.

1.3 Background

Borland Software Corporation is a software company headquartered in Austin, Texas [2]. It was founded in 1983 by Niels Jensen, Ole Henriksen, Mogens Glad and Philippe Kahn. In 2005, the company generated 277 Million in Revenues [7].

Towards the end of 2002, IBM has paid 2.1 Billion USD to acquire Rational Software Corp. IBM has maintained the Rational brand name, which is now operating under the IBM Software group.

In this report, we evaluate the following tools:

- Borland Together for Eclipse Release 2 Version 8.1.1 Build id: 4359.1 (2007)
- IBM Rational Software Architect Version 7.0.0.2 (2007)

During the experimentations, we also utilized the following tools:

- IBM Rational RequisitePro 7.0.1.0 (2007)
- IBM Rational License Server v 7.0.0.0.
- IBM WebSphere Business Modeler 6.2.0.0
- IBM WebSphere Integration Developer 6.2.0.0
- Borland CaliberRM version 2008.
- Borland StarTeam version 2008.
- IBM Rational Method Composer v7.0.0.0

All the software tools and configurations are installed on a single VM image to facilitate verification and future work.

A Glance on the Interface

We introduce the reader to the interface of both Borland Together for Eclipse and IBM Rational Software Architect. With a closer look at the interface, we anticipate this will assist the reader in visualizing the evaluation criteria presented in the following sections. Figure 1 illustrates a typical view of Borland showing a number of requirement related views.

Figure 2 illustrates a typical view of IBM Rational Software Architect portraying a number of requirement related views.

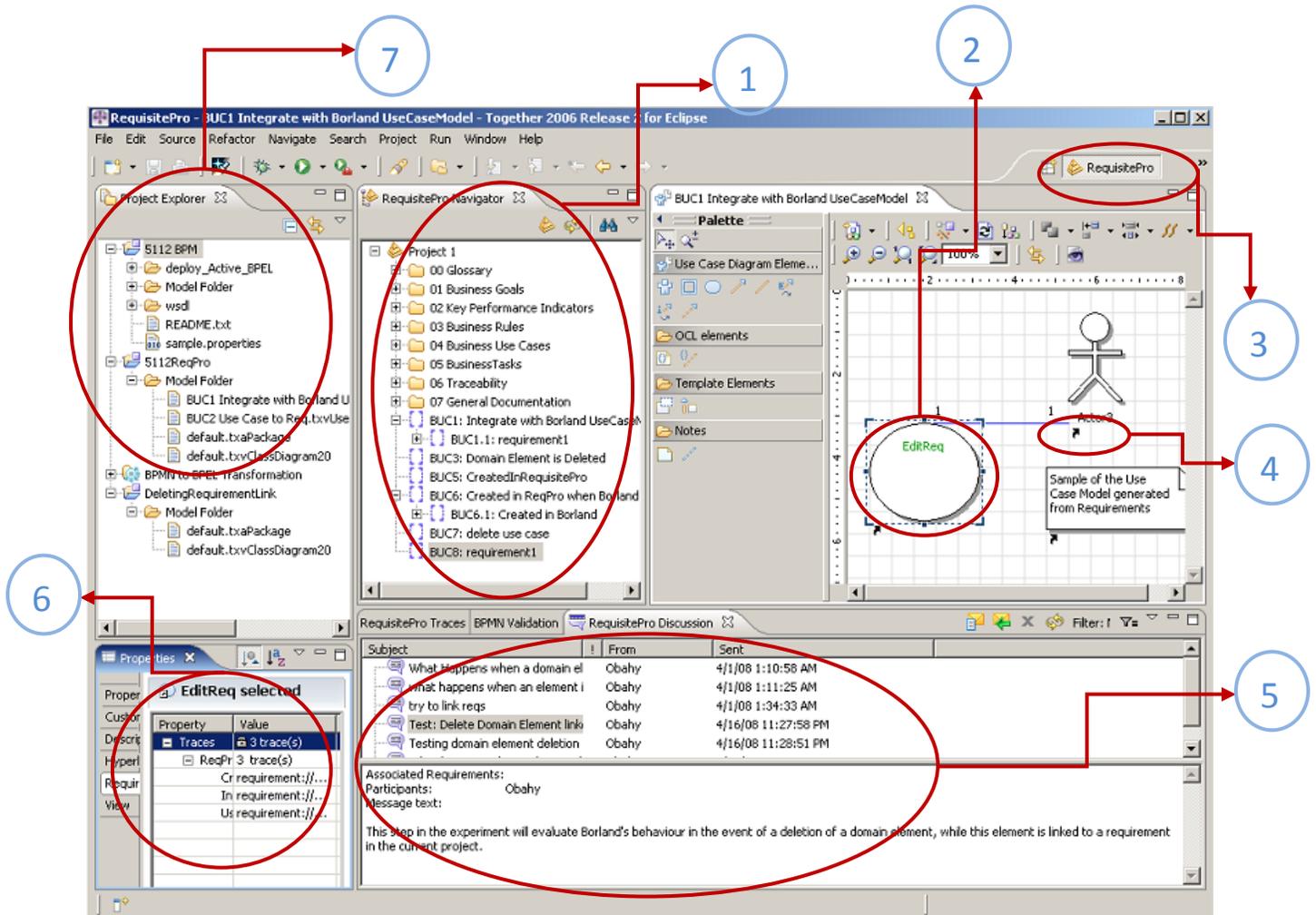


Figure 1: Borland Integration view

Legend:

- 1: A RequisitePro project opened in Borland workspace.
- 2: A use case with a green text indicating the existence of a link to requirements.
- 3: RequisitePro perspective.

- 4: Model link.
- 5: Borland's Requirement discussion view.
- 6: Requirement traces.
- 7: Borland's project explorer.

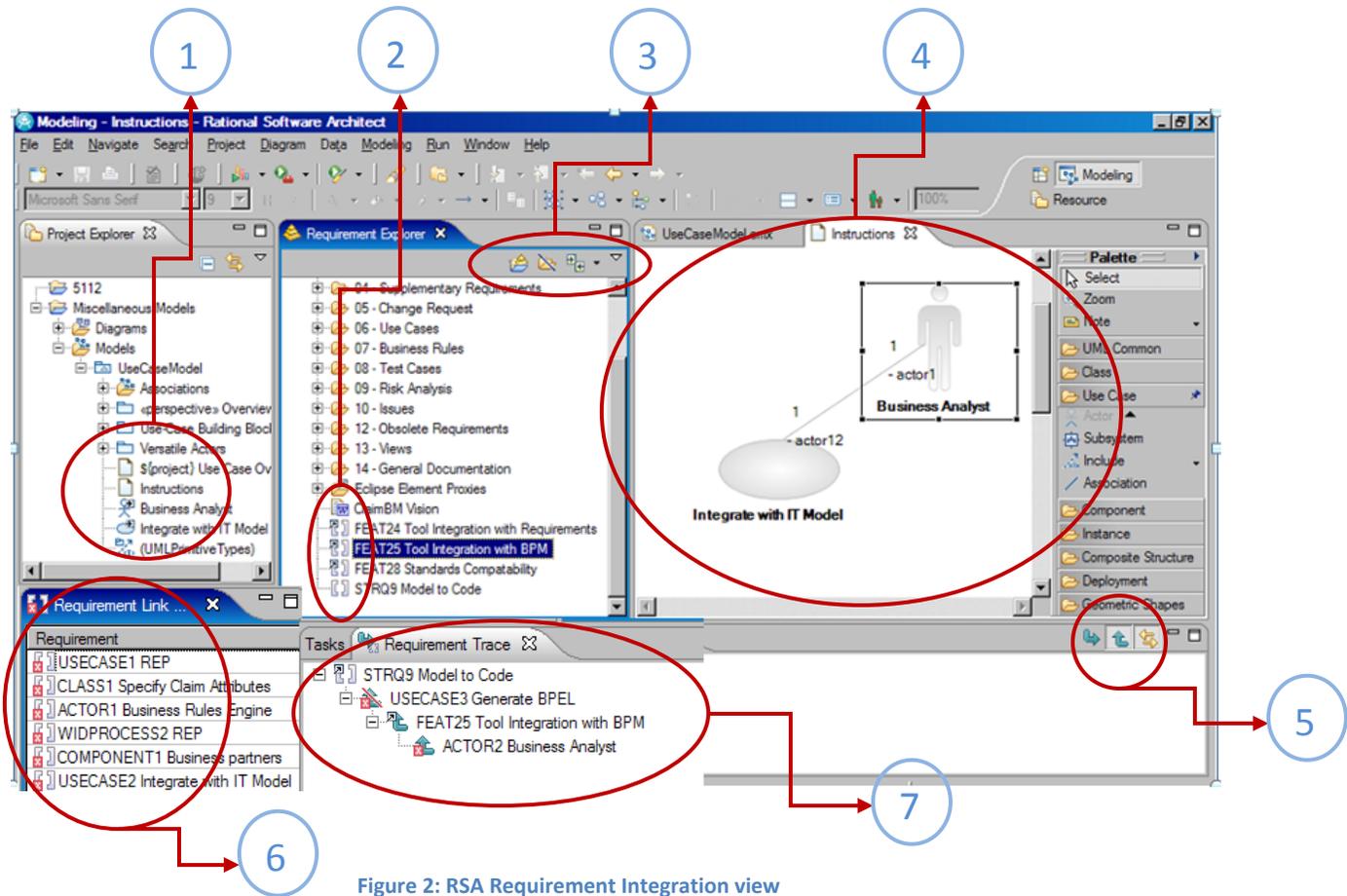


Figure 2: RSA Requirement Integration view

Legend:

- 1: RSA Project explorer showing requirements links
- 2: RequisitePro project explorer in RSA showing requirement links.
- 3: Icons for opening and controlling of RequisitePro projects.
- 4: A Use Case and an Actor in RSA model diagram view.
- 5: Icons to control requirement traces view.

6: Requirement traces view exhibiting broken links and link problems.

7: RSA's requirement links problem view.

2 Evaluation Criteria:

The evaluation criteria is aligned with BPM's MDA initiative, and is focused on Requirement integration and handling. The evaluation tries to uncover the tool integration capability with a Requirement Management tool. In addition, the evaluation investigates how this integration is being handled during the transformation from a Computation Independent Model (CIM) to a Platform Independent Model (PIM), and from a PIM to a Platform Specific Model (PSM). Specifically, this report uncovers how the tools handle requirement integration information and requirement links during model transformations. To ensure that our evaluation is of proper depth, we use a Business Process Model as a representation of CIM, a UML Use Case Model as a representation for PIM, and a BPEL Model as a representation for PSM. This scope has the following benefits:

- The evaluation can investigate the tool specific features with sufficient depth.
- This scope will allow the evaluation to reveal the tool handling of requirements before and after a model transformation has occurred.
- Initial experimentations indicate that for each tool, the handling of requirement integration is almost identical with respect to different models. Therefore, the selection of a Business Process Model, a Use Case Model, and a BPEL Model is a representative sample of the MDA CIM, PIM and PSM models.

BPM is evaluating the alternatives based on the following core and soft criteria:

- Core criteria
 - a. Tool integration with Requirements
 - b. CIM to PIM support
 - c. PIM to PSM support
- Soft Criteria
 - a. Third Party evaluation
 - b. Integration with Software Development Process
 - c. Market share and financial factors
 - d. Hardware requirement and Platform support

The following sections explain our evaluation criteria in further details.

2.1 Integration with Requirement

BPM needs to better align software development with business objectives. To do so, it is essential that development activities be driven by requirements.

This criterion will be evaluated based on the architectural tool integration support with requirements. Both, Rational Software Architect and Borland Together for Eclipse, have integration support with Rational RequisitePro. The criterion covers the following evaluation features:

2.1.1 Requirement Links types

To enable driving development by requirements, architectural elements should be linked to requirements. BPM is specifically interested in supporting the following requirements links types:

- **One-to-One Links:** This type of requirement links associates one architectural element to one requirement, or vice versa. In other words, there must be one element on each side in this type of links.
- **One-to-Many and Many-to-One:** This type of links will have one requirement on one end and a number of architectural elements on the other end, or vice versa.

2.1.2 Requirement Model Navigation and Editing

BPM expects the architecture tool to support the navigation between model elements and requirements. For example, starting from a domain element, the architect should be able to navigate to all linked requirements. This feature also assesses the tool handling of domain element and requirement editing. For example, we try to uncover the tool's behavior in the event of a domain model element editing or deletion. Similarly, we try to uncover the tool behavior in the event of a requirement editing or deletion.

This feature will be assessed based on consistency, usability, and the existence of views to facilitate the navigation.

2.1.3 Model elements and requirement creation

As software systems are developed iteratively, it is inevitable that the Architect may need to create new requirements based on his or her analysis and design of the software system. Additionally, the Architect may create additional Use Cases to realize specific requirements. We attempt to answer the following key questions. Does the tool support the use of the existing requirements to create Use Cases in the architecture tool? How about the creation of requirements based on Model elements?

2.1.4 Requirement Integration preferences

Each project is unique. The architecture tool should allow the user to specify preferences related to the requirement integration. For example, requirement editing privileges, links creation rules, and display related preferences should be user configurable settings.

2.1.5 Supported requirement views

BPM would like to have the Architect provided with all the necessary views to properly perform his tasks, without having to open the Requirement Management Tool.

This feature will evaluate the number and usability of the different requirement views available in the architectural tool.

2.1.6 Impact Analysis

Inevitably, Architects will change the architecture of the solution during the development lifecycle of the project. A change in the system architecture may, or may not, have an impact on the system requirements. The changes that the architect makes should be reflected in the requirements, and the concerned stakeholders should be notified to take appropriate action, if necessary.

This feature does not cover the Requirement Management Tool support for Impact analysis; rather, the feature evaluates the architecture tool capability to demonstrate the impact of the architectural changes on the requirements.

2.2 CIM to PIM support

In a typical MDA project, Models are transformed from CIM to PIM, PIM to PSM, and PSM to executable code. Since we are evaluating the tool integration support, we must investigate the tool handling of requirement links information in the context of model transformations.

Business Process Model to Use Case Model is a representative sample for CIM to PIM transformation. This criterion will evaluate the tool handling of requirement integration information portability from the Business Process Model to the generated Use Case Model. We will attempt to answer the following questions.

Has the Model transformation resulted in loss, or distortion of requirement link information?

Are the requirement links in the Use Case Model semantically equivalent to the requirement links in the Business Process Model?

How does the tool manage updates in requirement links after the transformation has been executed?

2.3 PIM to PSM support

This feature evaluates requirement information portability when a model is transformed from PIM to PSM. Both Borland and RSA supports the generation of BPEL Model from the Business Process Model.

This criterion evaluates the requirement links portability when such a model transformation is executed. Similar to CIM-PIM transformation, we expect the tools to handle requirement links and integration information in a semantically meaningful and consistent manner. We also investigate the tool handling of requirements links when an update occurs in the generated BPEL model.

2.4 Soft Criteria

These criteria are evaluated in fewer details. The evaluation goal is to identify high level support features without evaluating the specifics. The purpose of these features' inclusion is to ensure a proper coverage of the features of concern to our organization.

2.4.1 Third Party evaluation

Since our evaluation criteria are very specific and we are evaluating an unconventional functionality of the architectural candidate tools, we depend on a third party evaluation for assessing the overall performance and capability of the tool. In this report, we refer to Gartner's Magic Quadrant Report [4].

2.4.2 Integration with Software Development Process

To support MDA initiative, BPM has to follow a well defined development process. This feature will be evaluated based on the tool support for integration with a software development process.

2.4.3 Market share and financial factors

BPM prefers a tool that is well adopted, with a sizable market share. This will enhance the communication with clients and stakeholders. This criterion is evaluated based on the market share of the tool, and the cost of the tool license.

2.4.4 Hardware requirement and Platform support

This feature focuses on the tool level of support of operating system platforms, and the minimum hardware requirements. Generally, a wider range of platform support and less hardware requirements are desirable.

Figure 3 classifies our evaluation criteria from an MDA perspective. Criteria in blue boxes are core criteria, and those in white boxes are soft criteria.

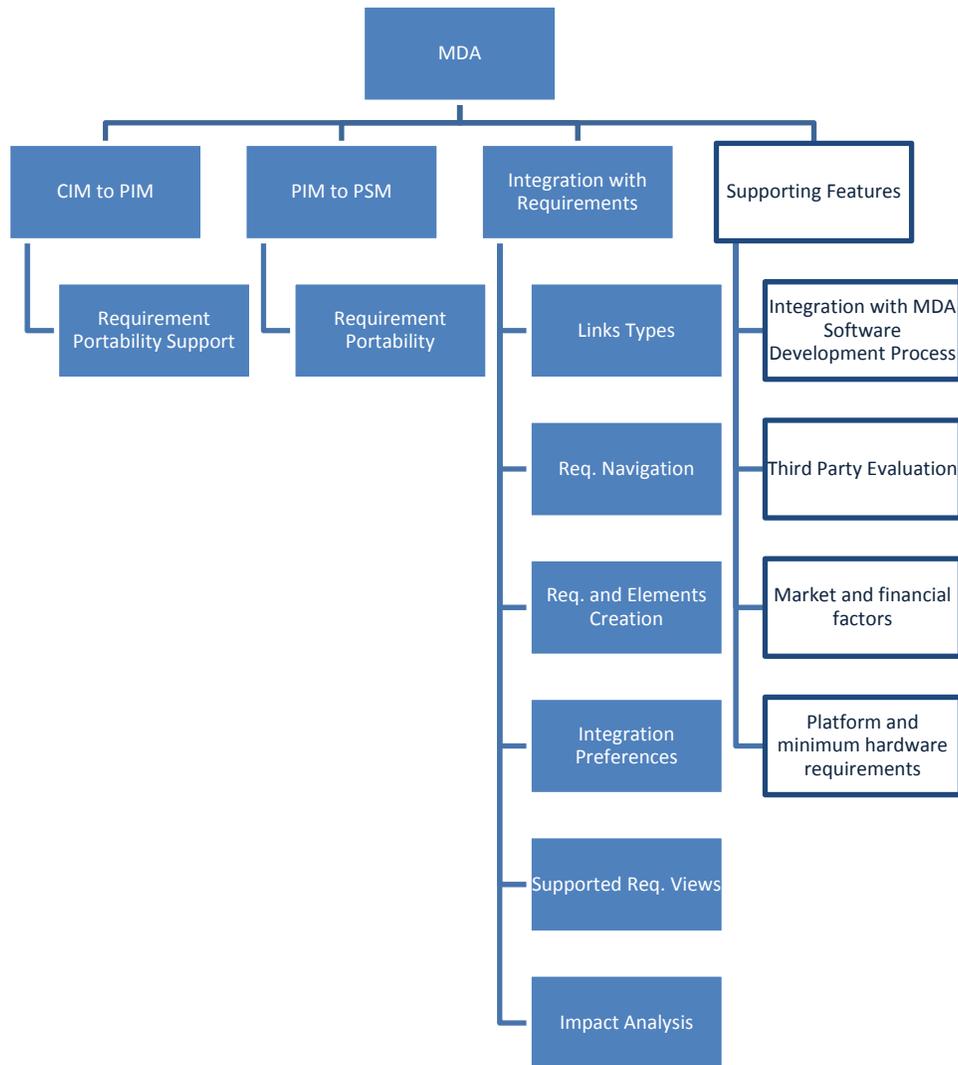


Figure 3: Evaluation Criteria

2.5 Evaluation Methodology

Core criteria are evaluated by the following roles from our organization: an Architect, a Business Analyst, a Requirement Specifier, a Project Manager, and a Business Process Analyst. The weight of core criteria is equal to two.

Soft criteria are evaluated by consensus; all evaluators will agree on a specific evaluation value for each criterion. The weight for each soft criterion is equal to one.

The evaluation is based on a scale from one to four; one being the least favorable and 4 being the most favorable. The following table summarizes the evaluation values:

Table 1: Evaluation measures

Value of evaluation	Significance
1	The feature is significantly below our expectations or needs and is not acceptable without good justifications.
2	The feature is below our expectations or needs, but is acceptable.
3	The feature is above our expectations or needs and is favorable.
4	The most favorable evaluation. The feature covers and exceeds our expectations and needs.

3 Experiment

In order to have an objective evaluation of the alternatives, we have designed an experiment that covers all of our core evaluation criteria. The experiment will be artificially simple, but realistic in representing a real MDA-like project. The experiment is based on a small number of requirements in Rational RequisitePro Repository and a simple Business Process Model composed of few processes and tasks. The next section presents our experiment in further details.

3.1 Experiment Prerequisites

To perform the experiment, these prerequisites are required:

- Rational RequisitePro Repository with fourteen requirements.
- Rational RequisitePro Client.
- Rational License Server.
- A valid and complete Business Process Model with seven tasks and transitions between each of them.
- A valid and complete Business Process Model created in Borland Together for Eclipse composed of seven tasks and links between each task.
- Links between the Business Model elements and the requirements in Rational RequisitePro Project.

3.2 Scenario

Phase one of the experiment is to enable and assess the integration of the architecture tool with the Requirement Management Tool. During this phase, we also evaluate the tool integration capability with the software development process tool.

The second phase of the experiment utilizes the Business Process Model, the Rational RequisitePro requirements, and the requirement links as input for the tools under study. Figure 4 illustrates the input for this phase of the experiment.

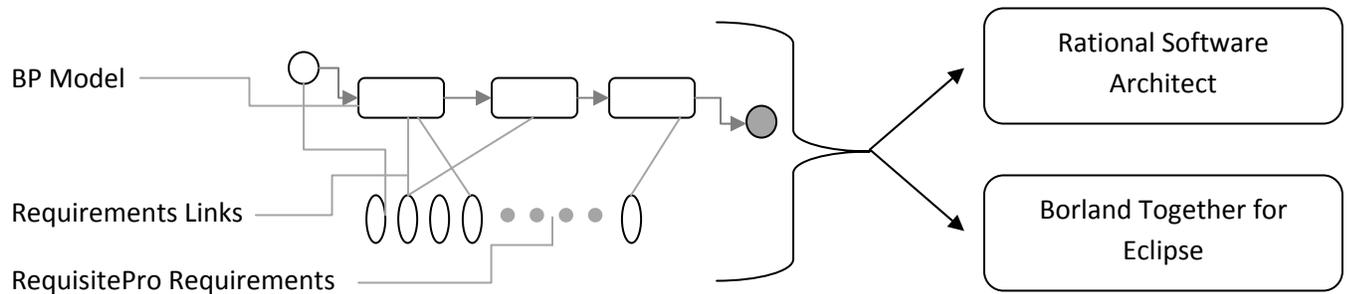


Figure 4: Experiment Input

The third and last phase of this experiment is concerned with the generation of a BPEL Model as a representative of a PSM model. During this phase we experiment with the generated BPEL model to evaluate the tool handling of requirement links during the transformation, and in the event of a Model update.

4 Evaluation of the Candidate tools

This section summarizes our findings and the evaluation of the committee members. This section is organized as follows. For each feature, the findings of our experiment are summarized for Borland and for RSA. This is followed by our analysis and the evaluation verdict of the committee members. The committee members make their evaluation independently based on their job requirements and tasks. At the end of the section we present the overall evaluation of the candidate tools.

4.1 Integration with Requirement

Both Borland and RSA support the integration with Rational RequisitePro. RequisitePro has to be installed on the same machine to enable the integration. However, RequisitePro and the candidate tools do not require the sharing of the workspace with RequisitePro.

You have to manually install and enable the RequisitePro capability into Borland to enable the integration. We have also experienced difficulties integrating Borland and RequisitePro when using a temporary license key for RequisitePro. We had to install Rational License Server for Borland to properly identify and authenticate the RequisitePro license. Otherwise, Borland reports an error stating that RequisitePro license has expired.

Requirements created in Borland, are not immediately reflected in RequisitePro Project. Borland had to be closed for the change to be committed to the RequisitePro project. Similarly, editing requirements in RequisitePro are not immediately reflected in Borland. It seems Borland uses a local temporary copy of RequisitePro Project, where changes are committed to the original copy upon closing the Requirement View. This results in a number of inconsistent behaviors. For example, you cannot open the Requirement View in Borland if RequisitePro is already running. However, if the Requirement View is open in Borland, you can open RequisitePro.

RSA was friendlier with the integration as expected, since RequisitePro and RSA belong to the same family of tools under IBM. However, even in the case of RSA, the tool does not support the integration with the Web Based RequisitePro Client. The next section summarizes our findings.

4.1.1 Requirement Links types

Borland Evaluation

Borland implements two types of links to requirements, Model Links and Element Links.

Model Links associate the whole Use Case Model to a Requirement Project. The link is displayed as a black arrow on every model element created within that model.

The second type of link is Element Link. This link is not displayed by any arrows; however, the link is displayed by having the model element name converted to green. In the Project Navigation View, this link is displayed as a green circular shape.

Borland also allows users to link Notes to requirements. This is an interesting feature specifically when there is a non-functional requirement, or a requirement that does not directly relate to a use case.

RSA Evaluation

RSA employs two types of requirement links, Direct Association Links and Indirect Association Links.

Direct Association Link is a one-to-one link where there is exactly one element on either end of the link.

Indirect Association Link can have more than one element on one side of the link. For example, you can link one requirement to more than one domain element in Rational Software Architect, or you can link one domain element to more than one requirement in Rational RequisitePro.

Analysis

We do not see a value of having Model links in the case of Borland. This type of links will be of value only when there is a need to link model elements to different requirements in different RequisitePro projects. Otherwise, as in the case of our experiment, the whole model is linked to the same RequisitePro Project.

We also question the value of the Indirect Association Links in the case of RSA. While this link type has semantic significance, it does add to the complexity of managing requirements by having to manage two types of links instead of one. In Borland, the same link type is used when the domain element is linked to a single requirement or a multiple requirements.

Table 2: Requirement links types

	Architect	Bus. Analy.	Req. Spec.	Proj. Mang.	Process An.	Average
Borland	3	3	4	3	4	3.4
RSA	3	3	3	3	2	2.8

4.1.2 Requirement Model Navigation and Editing

Borland Evaluation

To navigate from the Model element to the related (Linked) requirement, the user has to open the context menu of the domain element and navigate to the requirement. The properties view of the domain element also displays the linked requirements.

In addition, Borland supports “Element Traces View”. This view displays the linked-to requirements, along with the RequisitePro Project name and location.

Once a requirement link is created, and even when a synchronize traces command is executed, Borland does not synchronize the status of the requirement or the model element. For example, if you navigate to a requirement that has been deleted, Borland will result in an error message displayed in Figure 5.

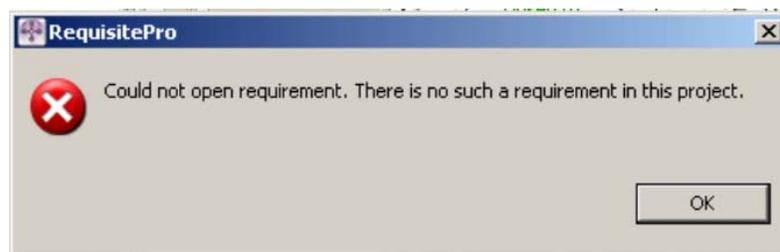


Figure 5: No requirement Error Message

This lack of synchronization is exhibited in the case of editing, deleting, or renaming of both, domain elements and requirements.

RSA Evaluation

RSA supports a somewhat intuitive navigation between domain elements and requirements. The context menu in the project explorer allows the user to select the linked-to requirement in the Requirement Explorer View within RSA, or the requirement in the Requirement Explorer View in RequisitePro. From a requirement, you can navigate to the requirement trace, from which you can navigate to the domain element(s).

The behavior in the case of deletion or editing of a domain element is somewhat more complex because RSA supports two links types. In the case of Direct Links, the deletion of the domain elements results in the deletion of the link itself with the requirement. In the case of the Indirect Link, this results in a broken link. In both links types, the editing of the domain element results in a change in the name of the requirement trace (The requirement trace is assigned the name of the domain element).

In the case of editing, RSA implements an incomparable feature to Borland. The user can synchronize the name and text of a requirement with the name and text of a Use Case. The synchronization policy is by means of prompt. The user is prompted to select which name and text to assign to the Use Case.

Analysis

Both Borland and RSA implement a comparable navigation methodology. The navigation is supported by means of specific views, and by the utilization of context menus.

The distinction between Borland and RSA in the case of Model and Requirement editing is evident. Borland does not synchronize or update requirement integration information when a model element or requirement is edited or deleted. As a matter of fact, if you delete a specific requirement, the domain element will still point to that deleted requirement, even though the requirement no longer exists. Borland does not report broken links in the event of editing or deleting of domain elements or requirements.

RSA, on the other hand, does report broken links, and supports a requirement links problem view in order to help the architect fix the broken links if a fix is required.

Table 3: Requirement Model navigation and editing

	Architect	Buss. Analy.	Req. Spec.	Proj. Mang.	Process An.	Average
Borland	2	2	1	1	2	1.6
RSA	4	4	3	2	3	3.2

4.1.3 Model elements and requirement creation

Borland Evaluation

Borland does not support the creation of Use Cases from requirements, or requirements from Use Cases. It is worth mentioning, it is possible to create a Business Process Model element from a requirement. We find this an inconsistent behavior among different models within Borland. The hidden assumption here is that developers create requirements first, from which they generate the Business Process Model, from which the tool supports the generation of both a Use Case Model and a BPEL Model. This is consistent with Borland's MDA vision [9]. We criticize this approach since both requirements and the Business Process Model may be developed iteratively and in parallel.

RSA Evaluation

You can create a Model Element from requirements by means of dragging and dropping the requirement onto the model diagram. This action will automatically create a direct link, as expected, linking the domain element and the requirement. The domain element name is given the same as the requirement name. However, the creation of requirements from domain element is not supported in the same manner. The user has to create the requirement and then create the requirement link. The drawbacks of this procedure are the overhead, and the naming convention. In this case, the requirement has to be manually given an appropriate consistent name.

Analysis

The lack of ability to create Use Cases from requirement is not favored. However, the workaround is simple, create a model element and give it the name of the requirement and link the two together. We therefore see this as a usability issue, rather than a functionality issue.

In a large project, the overhead of manually creating domain elements will be significant. The Architect and the Requirement Analyst will have to put more effort into creating the Use Case Model and the requirements.

We also criticize RSA’s inability of creating requirements from domain elements. We see this as an unjustified limitation, especially since the creation of a domain element from a requirement is supported.

Table 4: Model elements and requirement creation

	Architect	Buss. Analy.	Req. Spec.	Proj. Mang.	Process An.	Average
Borland	1	3	1	3	4	2.4
RSA	4	3	4	3	3	3.4

4.1.4 Requirement Integration preferences

Borland Evaluation

Borland’s preferences allow users to enable and disable requirement linking. In addition, users are allowed to enable or disable in-place editing of requirements.

Borland implements relatively more sophisticated requirement integration preferences for Caliber. Since the decision is to evaluate the tool integration with RequisitePro, the Caliber Preferences capabilities are not taken into consideration.

RSA Evaluation

The preferences of Requirement Management in RSA are limited to the settings of the default user for the logging into the RequisitePro Project, and the enabling or disabling of requirements links to linkable domain elements, and enabling or disabling of Requirement-Use Cases synchronization.

Analysis

Requirement links types, Links creation rules, and visualization of links are not configurable in either of the two tools. It is desirable to be able to configure the integration according to the specific needs of a project and client. Our assessment concludes that both tools have fallen short of our expectation of the configurability of requirement integration capability.

Table 5: Requirement integration preferences

	Architect	Buss. Analy.	Req. Spec.	Proj. Mang.	Process An.	Average
Borland	2	2	3	1	2	2
RSA	2	2	2	1	2	1.8

4.1.5 Supported requirement views

Borland Evaluation

Borland supports the following requirement related views:

1. RequisitePro Navigation View.
2. Traces View.
3. RequisitePro Discussion.
4. Element Traces.
5. Trace Synchronizer.

For details on the functionality of these specific views, we recommend the reader to refer to “*Borland Together for Eclipse RequisitePro Integration Guide*” [11].

RSA Evaluation

RSA supports the following requirement related views:

1. Requirement Explorer.
2. Requirement Links Problems.
3. Requirement Query Results.
4. Requirement Trace.

For details on RSA’s requirement views, we recommend the reader to refer to “*Model Rational RequisitePro requirements in Rational Software Architect*” [12].

Analysis

Our evaluation of the supported views can be summarized as follows:

The architect using Borland can participate and initiate discussions from Borland, with no need to open RequisitePro Project for this task. RSA, on the other hand, discussion view is not available in RSA. The Architect will have to open the requirement in RequisitePro and navigate to the discussion tab to participate or initiate discussions. Discussions cannot be viewed or opened in RSA.

Borland does not have a Requirement Links Problem View. During our experiment, we were unable to create any requirement link problem because Borland ignores all model or requirement updates after a link is created, hence, the lack of Requirement Link Problem View. On the other hand, RSA supports the Requirement Link Problem View. The view details the existing links problems and information to help the architect to identify and fix link problems.

Table 6: Supported requirement views

	Architect	Buss. Analy.	Req. Spec.	Proj. Mang.	Proc. Cheo.	Average
Borland	1	2	3	2	3	2.2
RSA	4	4	3	4	3	3.6

4.1.6 Impact Analysis

Borland Evaluation

Requirement Traces shows what requirements are linked to a specific domain element. For example, if the domain element is linked to three requirements, we see three traces in the Requirement Traces View displaying the domain element name, along with the traced-to requirement. However, any changes or updates in a domain element will not be reflected on the requirements or the trace, making it harder for the Architect to assess the impact of his or her updates. As a matter of fact, if a domain element is deleted altogether, there is no visual representation of the impact of the deletion on the requirements.

RSA Evaluation

RSA Requirement Traces View allows the users to track requirement traces to and from the requirement in focus. However, the view does not support the same for the domain element in focus. Broken traces are also shown in this view.

For example, if the domain element is deleted, the associated link(s) are automatically converted to a broken link. The user can open the requirement traces view to see what links are broken and what the related domain elements and requirements are.

Analysis

Borland support of impact analysis is absent. On the other hand, RSA supports impact analyses to a moderate extent. In addition, based on our experiment, the usability and training required to accomplish the tasks are significant. More importantly, there is no visual representation on the requirements or the domain element itself indicating the existence of a broken link. The developer has to check the traces view to discover any broken links in his project.

RSA has also fallen short from our expectation, which is to help the architect evaluate the impact of his changes on the requirements. We find that RSA better helps illustrate the impact of a change in requirements on the architecture, but not the other way around. This is due to the fact that the Requirement Traces View shows the linked requirements, but not the linked to or from elements.

Table 7: Impact analysis

	Architect	Buss. Analy.	Req. Spec.	Proj. Mang.	Process An.	Average
Borland	1	2	1	1	2	1.4
RSA	4	2	4	4	3	3.4

4.1.7 Requirement Links Portability (BPM to Use Case Model)

While there is a wealth of resources on the tools support for MDA and model transformations in Borland and in RSA, we have found no resources or information in the literature and related documentation on the tools support for requirement links portability. We have not been able to find any documentation on the specifications of links portability from model to model.

Borland Evaluation

All requirement links created in the Business Process Model are not carried to the Use Case model. While Borland supports the creation of requirement links in a similar manner in the Use Case model, with few exceptions, Borland does not support the portability of Requirement links information from BPM to the Use case Model.

RSA Evaluation

Requirement links in the Business Process Model are portable to the Use Case Model. RSA follows a simple rule, all requirement links (Direct Links and Indirect Links) are converted to indirect links. To illustrate this portability, we have to go into the details of the transformation of the Business Process Model to the Use Case model. For the purpose of evaluating the requirement links portability, we illustrate the transformation in a simple situation. A task in the Business Process Model is transformed into a Use Case in the Use Case Model. The role of the task becomes the actor on that Use Case. It can be clear in that simple example that the requirement that is originally linked to a single task in the Business Process Model is now linked to two domain elements, the task and its actor. Hence, the Direct Link is transformed into an Indirect Link.

It is important to note that once the transformation has occurred, the links in the Use Case Model and the links in the Business Process Model become totally separate. Updates in either model will not be reflected in the other one. For example, assume we have a link “L” linking requirement “R” to a task “T” in the Business Process Model that generated a Use Case “U” and an Actor “A” and an automatically generated Indirect Link “I” in the Use Case model. Further, assume the Architect has deleted or updated “U”, “A”, or “I”. The update the Architect makes has no effect on the originating “T” and “L”.

Analysis

There is a clear gap in feature support between Borland and RSA. In an MDA project, BPM expects to be able to generate Use Cases automatically from Business Process Models. The fact that Requirement links information is totally dropped in the case of Borland will induce some overhead to recreate the requirement links.

RSA, on the other hand, has managed to keep the links information during the transformation process. However, the generated Use Case Model is now totally separate from the Business Process Model. It is inevitable that the requirement links in the Business Process Model will be continuously updated during the lifetime of the project. Not only that the updates on either model are not reflected on the other mode, but also it is not possible to synchronize those updates to the Use Case Model. We anticipate a significant manual overhead to keep requirement integration information in synch in the two models during the development lifecycle.

Table 8: Requirement Links Portability (BPM to Use Case Model)

	Architect	Buss. Analy.	Req. Spec.	Proj. Mang.	Process An.	Average
Borland	1	1	1	1	2	1.2
RSA	4	4	4	4	3	3.8

4.1.8 Requirement Links Portability (BPEL Model)

Borland Evaluation

We have investigated the generated BPEL visual representation, as well as the XML representation. Requirement links are not portable. All requirement links, traces, and information are not portable to the BPEL model.

We note that Borland employs an elaborate transformation to BPEL model. However, our evaluation is exclusively focused on Requirements Links portability.

RSA Evaluation

RSA supports links portability. Unlike the Use Case Model where the Direct and Indirect Links are converted to Indirect Links, BPEL Model has retained the type of links after the transformation. In the case of BPEL Model, there is a clearer one-to-one transformation from the Business Process Model to the BPEL Model. For example, business process tasks are transformed into a BPEL Model node. This one-to-one transformation allows RSA to maintain the requirement links types.

Updates in the BPEL Model are not reflected on the originating model. In addition, all requirement links in the BPEL model are independent from the originating model and link information. Similar to Use Case Models, there will have to be manual updates to keep the requirement links information in synch.

Analysis

Borland does not support requirement links portability in the case of BPEL Models. In RSA, we note that the requirement links information is not embedded in the XML file. BPEL project has auxiliary files where such information is stored. This specific set up makes requirement links portability significantly harder if the model is exported to another BPEL Editor provider.

We also note that all requirement link information in the BPEL Model is independent of the originating model and link information. This behavior, while consistent with the behavior in Use Case Models, is undesirable.

Table 9: Requirement Links portability (BPEL Model)

	Architect	Buss. Analy.	Req. Spec.	Proj. Mang.	Process An.	Average
Borland	1	1	1	1	2	1.2
RSA	4	4	4	4	3	3.8

4.1.9 Third Party evaluation

Gartner's report specifies criteria for the inclusion and exclusion of a specific provider in its annual report. The inclusion and exclusion criteria are clearly defined in every report. Both IBM RSA and Borland Together tools have passed the criteria for inclusion. For this evaluation, we reference Gartner's Magic Quadrant for OOA&D Tools, 2H06 to 1H07 [4].

The magic quadrant shown in Figure 6, shows the overall position of RSA and Borland under the two axes; ability to execute and completeness of vision.

Both IBM and Borland are in the leaders and Visionary portion of the quadrant. However, the report shows a significant gap between IBM and Borland in the favor of IBM.

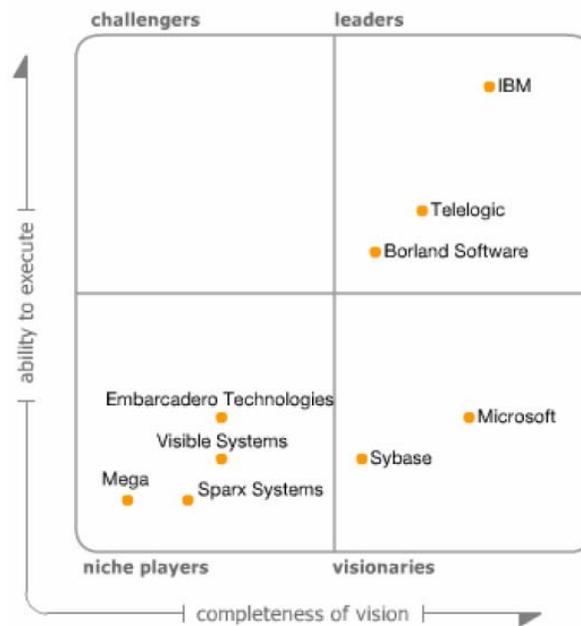


Figure 6: Gartner's Magic Quadrant [4]

Table 10: Third Party evaluation

	Evaluation
Borland	2
RSA	4

4.1.10 Integration with Software Development Process

Borland supports an elaborate integration with Application Life Cycle Management Process. The integration supports different roles and guides the resources to perform their tasks, according to the defined development process.

RSA supports the integration with Rational Unified Process. The RUP process can interactively show relevant content, templates and workflows for the Architect, while he is performing his tasks. For example, Use Case Model definition, template and related tool mentoring information is displayed while the Architect is editing a Use Case Model. The integration is rather simpler in the case of RSA and relies solely on the integration of the Process Advisor View in RSA.

Table 11: Integration with Software Development Process

	Evaluation
--	------------

Borland	4
RSA	2

4.1.11 Market share and financial factors

Borland's Together for Eclipse overall market share across selected number of industries is 27% [4]. Borland's Together for Eclipse license cost is around 2,700 USD.

RSA market share across the same number of industries is 50% [4]. License cost depends on the number of users, and on existence of other valid Rational Software licenses. The full license cost for RSA is around 5,000 USD.

Table 12: market share and financial factors

	Evaluation
Borland	3
RSA	4

4.1.12 Hardware requirement and Platform support

Borland requires CPU Pentium 4 of 1 GHz, 500 MG of RAM, and a 570 MG of Desk Space. Borland's OS support includes Windows, RedHat, SUSE, and Mac OS 10.4 [6]. Borland also requires SVG viewer version 3.0 or newer.

RSA requires Pentium 3 of 500 Mhz, 1 GB of RAM, and 768 MG of Desk Space. RSA supports Windows, RedHat, SUSE, but does not support Mac OS. In addition, RSA support VMWare Environment [5].

Table 13: Hardware and Platform support

	Evaluation
Borland	3
RSA	3

4.2 Overall Evaluation

Table 14 summarizes the overall evaluation of the two candidate tools. The weighted total is calculated by summing all the evaluation multiplied by the weight and divided by the total number of weights.

Table 14: Overall evaluation

No.	Feature Type	Weight	Borland	RSA
1	Core	2	3.4	2.8
2	Core	2	1.6	3.2
3	Core	2	2.4	3.4
4	Core	2	2.0	1.8
5	Core	2	2.2	3.6

6	Core	2	1.4	3.4
7	Core	2	1.2	3.8
8	Core	2	1.2	3.8
9	Soft	1	2	4
10	Soft	1	4	2
11	Soft	1	3	4
12	Soft	1	3	3
Weighted total		20	2.14	3.23

As the table illustrates, RSA is evaluated higher than Borland by 1.09 points out of four, or by 27.25%.

5 Conclusion and Remarks

Our core evaluation criteria are quite specific and focused on the requirement integration capability. The core criteria are supported by a number of soft criteria to ensure a broader spectrum for evaluation, and to cover specific features of interest for our organization.

To ensure objectivity, the features have been evaluated independently for each tool. The overall evaluation is then calculated. The features have been evaluated independently by a number of roles within our organization.

Borland has significant gap in handling the requirement links and integration information during model transformation. This results in a significant loss of requirement links information. In addition, Borland does not support proper synchronization of links between domain elements and requirements. The lack of proper synchronization results in inconsistencies when a domain element or requirement is deleted.

RSA, on the other hand, exhibited an adequate and semantically meaningful requirement integration handling. Requirement links and integration information are not lost during the transformation of models. RSA supports a number of requirement views that allows the architect to manage integration with requirements in RequisitePro without the need to opening the requirement project in RequisitePro, with the exception discussions as the Architect using RSA has to open RequisitePro Repository in order to participate in requirements discussions.

Both tools have fallen short in terms of proper support for impact analysis, as well as proper support for user configurations of requirement integration preferences. Moreover, the candidate tools do not support proper requirement link synchronization across different models.

RSA scored higher than Borland by 27.25%. Based on our experiment and evaluation criteria, we recommend our organization to use RSA.

6 References

- [1] Balasubramaniam Ramesh and Matthias Jarke, "Toward reference models for requirements traceability". IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 27, NO. 1, JANUARY 2001.
- [2] O. Gotel and A. Finkelstein, "An Analysis of the Requirements Traceability Problem". Proceedings of First Int'l Conf. Requirements Eng., pp. 94±101, 1994.
- [3] Alan W. Brown, Simon K. Johnston, Grant Larsen, and Jim Palistrant, "SOA Development Using the IBM Rational Software Development Platform: A Practical Guide"
- [4] Gartner's Magic Quadrant Report, "Magic Quadrant for OOA&D Tools, 2H06 to 1H07". Document ID Number: G00140111. Gartner's issuance date: 30 May, 2006.
- [5]RSA hardware and software Requirements, IBM Rational Software, viewed April 15th, 2008. <<http://www-306.ibm.com/software/awdtools/architect/swarchitect/index.html> >
- [6] Borland's Together for Eclipse Hardware and Software requirements, Borland Together Integrated and Agile Design solutions. Viewed April 17th, 2008. Last Updated September 7th, 2005. <<http://techpubs.borland.com/together/tec2006/en/readme.html>>
- [7] Borland's History information. Viewed April 17th, 2008. <<http://en.wikipedia.org/wiki/Borland>>
- [8]Redmond Magazine, Last viewed on April 22nd, 2008. Last updated on May, 2007. "IBM to acquire Rational Software". < <http://redmondmag.com/news/article.asp?EditorialsID=5632>>
- [9] Choong Koon Fong, "Quick Start Guide to MDA, A Primer to Model-Driven Architecture Using Borland® Together® Technologies". September, 2006.
- [10] Bran Selic, "The paradigm of Model-Driven Development". IEEE computer society, 0740-745 9/2003.
- [11] Borland Software Corporation, "Borland together for Eclipses, RequisitePro integration Guide". August, 2006.
- [12] IBM Developer's work resource center, Article number 2742, "*Model Rational RequisitePro requirements in Rational Software Architect*". Last viewed on April 22nd, 2008. <<http://www.developers.net/ibmshowcase/view/2742>>

About the Author

Omar Badreddin is a PhD student at University of Ottawa, and a Research Visitor at IBM Center for Advanced Studies and Research. Badreddin conducts research in software modeling, model transformations, and model driven development. He is a regular contributor to IBM RedBooks Publications and an ITSO instructor.

This work is part of the academic course “5112 Software Engineering” under supervision of Dr. Daniel Amyot.

For additional information and related resources, including the VM image with all required software installed and configured, please send your request to **Omar_Badreddin@ca.ibm.com**