

A Validation Study of a Requirements Engineering Artefact Model for Big Data Software Development Projects

Darlan Arruda^a, Nazim H. Madhavji and Ibtehal Noorwali
Department of Computer Science, Western University, London, Canada

Keywords: Big Data Applications, Requirements Engineering, Artefact Model, Validation.

Abstract: The elicitation, specification, analysis, prioritisation and management of system requirements for large projects are known to be challenging. It involves a number of diverse issues, such as: different types of stakeholders and their needs, relevant application domains, knowing about product and process technologies, regulatory issues, and applicable standards. The advent of “Big Data” and, in turn, the need for software applications involving Big Data, has further complicated requirements engineering (RE). In part, this is due to the lack of clarity in the RE literature and practices on how to treat Big Data and the “V” characteristics in the development of Big Data applications. Traditionally, researchers in the RE field have created domain models that help in understanding the context of the problem, and in supporting communication and analysis in a project. Analogously, for the emerging field of software applications involving Big Data, we propose an empirically derived RE artefact model. It has been validated for qualities such as: accuracy, completeness, usefulness, and generalisability by ten practitioners from Big Data software development projects in industry. The validation results indicate that the model captures the key RE elements and relationships involved in the development of Big Data software applications. The resultant artefact model is anticipated to help in such activities as: requirements elicitation and specification; definition of specific RE processes; customising and creating a common vision in Big Data RE projects; and creating traceability tools linking the artefacts.

1 INTRODUCTION


Predominantly, the current focus in the field of Big Data software is on data analytics and the development of algorithms and techniques to process and extract value from huge amounts of data (Kumar and Alencar, 2016). In contrast, little research or industry practices focus on software applications and services that utilise the underlying Big Data to enhance the functionality and services provided to the end-users (Madhavji et al., 2015, Arruda and Madhavji, 2018).

While scientific literature (Anderson, 2015; Kumar and Alencar, 2016; Laigner et al., 2018) and economic outlook (Nadkarni and Vesset, 2017; Davenport and Bean, 2018) suggest that the field of Big Data is growing exponentially, there is no recognisable body of knowledge on the development of applications and services that utilise Big Data. Consequently, end-users are potentially missing out on the anticipated benefits of innovative applications

and services that could provide enhanced results, experience, or value. This void is also reflected in the field of Requirements Engineering (RE) where current RE practices (such as elicitation, specification, analysis, etc.) (Sommerville, 2009) do not prescribe how to treat Big Data and the “V” characteristics in the development of Big Data software applications.

The current difficulties in the RE process for Big Data applications is compounded by the lack of suitable domain or artefact models, considered important in RE (Berenbach, 2009). Such models are a means for understanding the various artefacts, activities, and relationships involved in the RE process (Penzenstadler et al., 2013, Nekvi and Madhavji, 2014).

In order to ameliorate the current situation, we created an initial RE artefact model (Arruda and Madhavji, 2017) for Big Data software applications. This model was subsequently assessed for qualities such as accuracy, completeness, usefulness, and

^a  <https://orcid.org/0000-0002-6756-2281>

generalisability by ten practitioners from Big Data software development projects in industry. This validation study is described in this paper including the resultant improved artefact model. Thus, the contribution of this paper, firstly, is the improved artefact model. Further, this paper creates a stronger baseline for the RE artefact model for Big Data application systems upon which can depend new applications development, RE technology development, and further empirical studies.

The remainder of the paper is organised as follows: Section 2 describes relevant background. Section 3 describes the methodology used in the validation study. Section 4 describes the assessment results. Section 5 compares the old and the new versions of the artefact model as well as introduces the post-validation version of the RE artefact model in the context of Big Data Software Developments Projects. Section 6 describes threats to validity and the respective mitigation strategies. Finally, Section 7 concludes the paper.

2 BACKGROUND

In this section, we discuss briefly the background literature on models and present the pre-validation version of the Big Data RE artefact model.

2.1 Models

Modelling is recognised as important in software organisations as exemplified by model-driven development (Selic, 2003). Complementing product modelling, is process modelling of activities along with their inputs and outputs (Humphrey and Kellner, 1989).

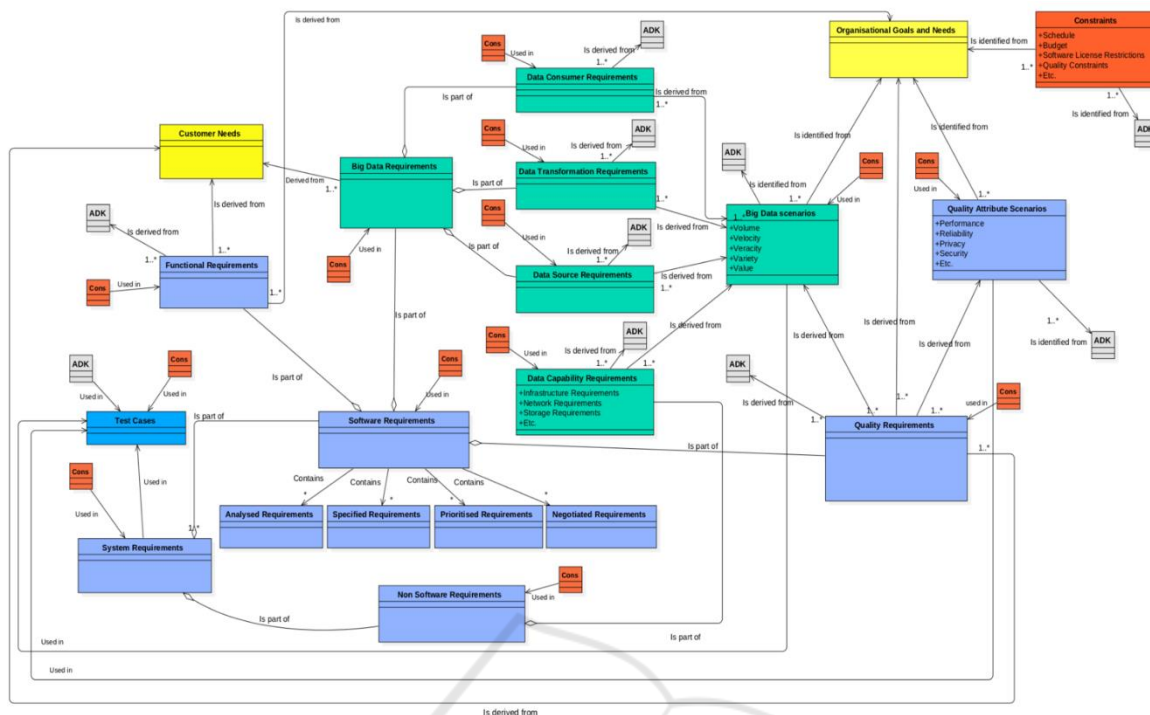
In RE, modelling is helpful for modelling requirements (Horkoff et al., 2017) and understanding the application domain (Berenbach, 2009). At a macro-level, an artefact model depicts project artefacts and relationships such that, it can be used to support requirements elicitation and specification (Mendez et al., 2011), product design, project decisions, and maintenance (Berenbach, 2009).

2.2 Pre-validation Version of the Big Data RE Artefact Model

With reference to Figure 1, the pre-validation RE artefact model for Big Data applications (Arruda and Madhavji, 2017) consists of three types of elements: *artefact*, *association* between two artefacts, and *cardinality*.

This version of the model is composed of 21 elements of which six are Big Data specific elements and numerous relationships. Example elements are (NIST, 2015): Data-Capability Requirements (typically infrastructure related): *the system shall support legacy, large distributed data storage*; Data-Source Requirements (e.g., *the system shall support high-throughput data transmission between data sources and computing clusters*); Data-Transformation Requirements (typically processing related): *the system shall support batch and real-time analytics*; Data-Consumer Requirements (e.g., *the system shall support diverse output file formats for visualisation*).

Example relationships are: (i) *Is-derived-from* relationship: when one or more artefacts can be derived from another artefact (e.g., quality requirements are derived from Big Data scenarios); (ii) *Is-identified-from* relationship: when one or more artefacts (e.g., Big Data Scenarios, Constraints and Concerns, etc.) are identified from another artefact (e.g., organisational goals); (iii) *Is-part-of* relationship represents aggregation: when one or more artefacts are part of one or more larger-class artefacts (e.g., functional requirement is part of software requirements); (iv) *Contains* relationship: when one artefact contains information about another artefact (e.g., software requirements contains analysed requirements); and (v) *Used in* relationship: when one or more artefacts can be used to guide in the definition of one or more other artefacts (e.g., project constraints are used in Big Data scenarios).



Legend: ADK - Application Domain-knowledge | Cons - Constraints

Figure 1: Pre-validation Big Data RE artefact model (Arruda and Madhavji, 2017).

3 RESEARCH METHODOLOGY

This section depicts the qualitative research methodology (Gibbs, 2007) composed of a 4-phase research process depicted in Table 1:

Table 1: Research phases, actions, and outputs.

| Phase | Action | Outputs |
|-------|---|---|
| 1 | Planning: created a data gathering instrument; identified practitioners from Big Data software projects. | Questionnaire (Section 4); list of practitioners. |
| 2 | Assessment of the preliminary RE artefact model by invited participants (convenience sampling). | Feedback from practitioners (Section 4). |
| 3 | Qualitative analysis (thematic coding (Gibbs, 2007)) of feedback (from Phase 2) and data from industry projects (re. artefacts); indexing/categorising text; grouped artefact-types and information as per the RE reference model (Geisberger et al., 2006): (i) business needs, (ii) | Improvements to be made to the model. |

requirements specifications, and (iii) systems specifications (see Section 5).

- Improved the pre-validation model based on output from Phase 3, maintaining the model primitives (Berenbach, 2009): (i) artefacts, (ii) relationships, and (iii) cardinalities.

4 MODEL VALIDATION

In (Shaw, 2003), the author describes several types of validation in software engineering research: (a) by analysis; (b) by experience; (c) by example; (d) by evaluation; (e) by persuasion; and (f) by blatant assertion. Shaw also explains that the validation type needs to be appropriate for the type of research contribution (e.g., validation by experience would be suitable for research results that have been used in practice by someone other than researcher).

For the descriptive model that we describe in this paper, the appropriate validation procedure is 'evaluation': to assess whether the proposed model

satisfactorily describes the phenomena of interest, in our case, development of Big Data software applications.

For the purpose of validation, we created an instrument (questionnaire) for gathering data, composed of 15 questions organised as follows: (i) background questions; (ii) technical validation questions concerned with *completeness* and *accuracy* of the elements and relationships depicted in the proposed model; and (iii) validation questions concerned with *usefulness* and *generalisability* of the proposed model.

The questions in the instrument had multiple-choice responses; used the 5-point Likert scale

(Likert, 1932) (*strongly agree to strongly disagree*); and a few open-ended questions concerning the artefact model. Thirteen practitioners in Big Data software development projects were invited of which ten 10 agreed to participate in the study. Three declined due to business constraints.

4.1 Descriptive Statistics

Table 2 gives descriptive statistics of the participants. The subsections describe the results of the study.

Table 2: Descriptive statistics of the study participants.

| Practitioner | Roles | Application Domains | Experience with Requirements and RE | Experience with Big Data |
|--------------|---|--|-------------------------------------|--------------------------|
| 1 | - Business Analyst - Developer - Researcher | - Marketing - IT/Telecom | Informal | 5+ years |
| 2 | - Requirements Analyst - Developer - Architect | - IT/Telecom | 1- 4 years | 3-5 years |
| 3 | - Business Analyst - Developer | - IT/Telecom | 1- 4 years | 1-2 years |
| 4 | - Manager - Requirements Analyst - Developer - Architect - Consultant - Researcher | - Marketing - IT/Telecom | 11-15 years | 5+ years |
| 5 | - Developer - Architect - Researcher | - Healthcare - IT/Telecom - Defense/Military - Commercial - Cyber security | 1- 4 years | 5+ years |
| 6 | - Requirements Analyst - Manager - Developer | - Government - Transport - Manufacturing | 1- 4 years | 1-2 years |
| 7 | - Developer - Researcher - Consultant | - Geospatial Data Processing | 1- 4 years | 1-2 years |
| 8 | - Requirements Analyst - Developer - Architect - Researcher | - IT/Telecom - Government | 16+ years | 3-5 years |
| 9 | - Developer | - Marketing - IT/Telecom - Geospatial Data Processing | 1-4 years | 3-5 years |
| 10 | - Requirements Analyst - Quality Assurance Engineer | - Transport - Telecom - Mobile | 1-4 years | 1-2 years |

4.2 Results

The following subsections discuss the validation results from specific angles: (i) accuracy and completeness of the model; and (ii) usefulness and generalisability of the model.

4.2.1 Accuracy and Completeness

The questions formulated to assess the *accuracy* and *completeness* of the model were divided into four Likert scale type of questions and one polar (yes-no) question followed by an open text-field.

Table 3 lists the four questions and practitioners’ responses. The responses fall predominantly within the ‘Strongly Agree’ and ‘Agree’ options for all the questions. When asked about the neutral choice made, **practitioner #5** replied that some element names (e.g., data transformation requirements) could change depending on the project. Also, he indicated that not all projects follow the naming proposed by NIST (2015), e.g., the term “*data capability requirements*” could be referred to as “*platform requirements*”. Likewise, **practitioner #10** replied that “*the relationships are okay and represent the way most of the applications are developed, but some other projects could have some different relationship labels*”.

Following the Likert scale questions, we asked: *Do you think any elements are missing from the*

proposed artefact model? Two **practitioners (#2 and #9)** answered “no”; whereas, the remaining eight participants answered “yes”. The suggestions from the “yes” respondents, were as follows:

Practitioner #1 -- non-functional requirements such as privacy and security should be depicted in the model. **Practitioner #3** -- the non-functional requirements related to the process (e.g., documentation quality and template patterns) could be introduced in the model. (We feel that the types and instances of non-functional *requirements* would likely differ from project to project). For example, some projects could have a catalogue of non-functional requirements focused on privacy and security whereas others could have a catalogue of non-functional requirements focused on performance and reliability. Thus, we decided not to include them explicitly in the model (for simplicity reasons). However, they can be considered as *contained inside* the “*Non-functional Requirements Specifications*” artefact, which is represented in the model.

Practitioner #4 -- the “data analytics” type of requirements was missing. We clarified that these types of requirements were indeed represented in the model as “*data transformation requirements*” as classified by NIST (2015). **Practitioners #1, #3, #4 and #10** -- to include elements related to the artefacts for technological requirements for the project, e.g., those elicited concerning the data pipeline: data collection, storage, processing, visualisation, and

Table 3: Results of the accuracy and completeness questions.

| Questions | Likert Items | | | | |
|---|---------------------------------|---|----------------------------|----------------|-------------------|
| | Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly Disagree |
| 1. To what extent do you agree that the schematic model reflects the type of RE artefacts in the development of Big Data applications in industry? | Practitioners 7 and 8 | Practitioners 1, 2, 3, 4, 5, 6, 8, 9 and 10 | | | |
| 2. To what extent do you agree that the names of the artefacts depicted in the proposed artefact model are appropriate? | Practitioners 3, 4, 6, 7, and 9 | Practitioners 1, 2, 8 and 10 | Practitioner 5 | | |
| 3. To what extent do you agree that the labels of the relationships in the artefact-model are appropriate? | Practitioners 3, 4, 6, 7, 9 | Practitioners 1, 2, 5, 8. | Practitioner 10 | | |
| 4. To what extent do you agree that the elements in the artefact model named: data-capability requirements, data-source requirements, data transformation requirements and data-consumer requirements – represent the whole spectrum of the types of Big Data requirements? | Practitioners 7, 8 and 9 | Practitioners 1, 2, 3, 4, 6 and 10 | | Practitioner 5 | |

management. (We agreed with the suggestion, thus adding the technological requirements related entities to the post-validation version of the model).

Practitioners #5 and #8 -- to include a note or a specific element addressing the application type based on the nature of data processing, whether it would be batch or streaming. (The type of application based on the nature of data processing would play an important role in defining the system’s requirements, however, it would not change the types of artefacts in the project. Adding the type of application as an entity would add complexity to the model. Thus, we decided to include an explanatory note linked to the entities denoting “*Big Data Scenarios*” and “*Quality Attributes Scenarios*” since they would cover information regarding the type of application being dealt with in the project).

Finally, **Practitioner #7** -- to better represent the entity denoting “Big Data scenarios”. Specifically, this label could be misleading because the scenarios are domain specific and do not describe only the data specific characteristics. (We agree with this recommendation. Thus, we added an explanatory note linked to the entities named “*Big Data Scenarios*” and “*Quality Attributes scenarios*”).

Improvements made to the model in response to the assessment, as well as the supporting rationale can be seen in Section 5, Table 8.

4.2.2 Usefulness and Generalisability

For assessing the *usefulness* of the artefact model, we asked the following question: *To what extent do you agree that artefact model is useful in practice?* Table 4 depicts that most of the participants ‘agree’ or ‘strongly agree’ that the model is useful in practice.

Table 4: Results of the usefulness question.

| Likert items | Practitioner # |
|----------------------------|----------------|
| Strongly agree | 3, 6, 7 and 8 |
| Agree | 2, 4, 9 and 10 |
| Neither agree nor disagree | 1 and 5 |
| Disagree | - |
| Strongly disagree | - |

Further, we asked the participants to give their opinion on the purposes the artefact-model would be useful for. Some example variety of answers we received are: **Practitioner #2** -- “*to aid in requirements gathering and initial architecture design.*” Also, “*as a guiding template for customer*

and executive level presentations.” **Practitioner #3** - “*with a clear artefact model, it is easier to go through all field/checklist that need to be considered in RE and in architecture design.*” **Practitioner #4** - “*to support the development of specifications for Big Data applications, development of test cases based on the requirements, traceability of requirements through the development cycle*” as well as for “*getting a big picture view of the project and how it fits into the organisation.*” **Practitioner #6** -- “*used as a reference to support the review and elaboration of development processes and policies in companies that work with data-centric applications.*” **Practitioner #7** -- “*This work (the proposed artefact-model) is a first step in providing a solid set of artefacts for supporting practitioners to reason about RE in Big Data Apps.*” **Practitioner #8** -- “*the design of a Big Data application involves a series of requirements artefacts that, in my opinion, are captured by the proposed model.*” Also, “*support the specification, validation, and test of Big Data applications.*” This view is also echoed by **Practitioner #9**. **Practitioner #10** -- “*good start to help in the elicitation process. The requirements analyst could use it to guide in the interviews, focus groups and workshops with stakeholders in order to identify the most important or relevant requirements.*” Finally, **Practitioners #1** and **#5** did not provide any opinion on *usefulness*.

Table 5 depicts a synthesis of categorised reasons and participants based on the analysis of total feedback received on *usefulness*.

Table 5: Reasons for *usefulness* of the model juxtaposed by participant groups.

| Reasons for usefulness of the model | Practitioner # |
|---|-------------------|
| Reason 1: provide a big picture of requirements artefacts used/created in the project. | 4, 6, 7, 9 and 10 |
| Reason 2: aid in requirements elicitation. | 2, 3, 4, 6 and 10 |
| Reason 3: aid in the definition of specific RE processes. | 6, 9, and 10 |
| Reason 4: aid in the specification, validation and testing of Big Data software applications. | 4 and 8 |
| Reason 5: aid in the architecture design; serving as template for executive presentations. | 3, 2 |

When asked: “To what extent do you agree that the artefact model is generic enough to be used in different Big Data software development projects (with few modifications)?”, most of the answers fell within the “agree” (six answers) and “strongly agree” (three answers) options (See Table 6). Thus, there is a consensus amongst the practitioners regarding the applicability of the artefact model in different projects, regardless of their unique characteristics.

Table 6: Results of the generalisability question.

| Likert items | Practitioner # |
|----------------------------|-----------------------|
| Strongly agree | 6, 9, and 10. |
| Agree | 1, 2, 3, 5, 7, and 8. |
| Neither agree nor disagree | 4 |
| Disagree | - |
| Strongly disagree | - |

5 PROPOSED VS. IMPROVED ARTEFACT MODEL

In this section, we present and discuss the improvements made to the pre-validation version of the model in response to the feedback obtained in the validation study as well as present the post-validation version of the RE artefact model.

5.1 Comparative Analysis

Table 7 shows that the model has changed drastically (in the total number of entities) -- doubled -- from 21 to 43 entities and tripled in terms of Big Data specific elements (from 6 to 18). Changes are due to missing elements in the pre-validation model (e.g., technological requirements, external interface requirements, data requirements) or implicit representation in the graphical nodes of the model (e.g., functional specifications contain functional requirements). Also, two new relationship types were added to the post-validation model (e.g., ‘assist-in’ and ‘is-composed-of’). Additions, changes, and removals made to the original model are described in Table 8.

Table 7: Comparative statistics between the pre- and post-validation artefact models.

| | Pre-validation | Post-validation |
|---------------------------------|----------------|-----------------|
| # of Elements | 21 | 43 |
| # of Relationship types | 5 | 6 |
| # of Big Data specific elements | 6 | 18 |

5.2 Post-validation Artefact Model

With reference to Figure 2 (the post-validation artefact model), the entities (artefacts) are grouped into the following three categories extracted from the requirements engineering reference model (Geisberger et al., 2006):

- (1) *Business needs artefacts*: These specify customer and strategic requirements, including product and business goals of the system under development (Geisberger et al., 2006). In the post-validation model, seven entities fall in this group of artefacts (see pink-coloured nodes in Figure 2).
- (2) *Requirements specification artefacts*: These contain functional and non-functional requirements. They are analysed and modelled from the customer and user perspectives and derived from (and justified by) the business needs (Geisberger et al., 2006). In the post-validation model, 15 entities fall in this group of artefacts (see green-coloured nodes in Figure 2).
- (3) *Systems specification artefacts*: These contain a definition of the functional system concept; the required behaviour and its integration into the overall system and environment. It defines constraints on the design and realisation of the system (Geisberger et al., 2006). In the post-validation model, 21 entities fall in this group of artefacts (see blue-coloured nodes in Figure 2).

The Big Data specific entities are: Big Data Requirements Specifications; Data Processing; Requirements Specifications; Data Consumer Requirements specifications; Data Source Requirements Specifications; Data Requirements Specifications; Big Data Scenarios; Technological Requirements Specifications; and their contained artefacts (e.g., Data requirements specifications contain data requirements and data modelling and linking details).

Table 8: Model changes.

| Entities Added | Rationale |
|---|---|
| <i>'Technological Requirements Specifications'</i> | Big Data technologies play a critical role in storing, processing, and managing data. Early decisions in technology selection can help simplify development and aid in the definition of system's architecture. |
| <i>'System Architecture, Design Components, and Abstractions'</i> | These artefacts are influenced by requirements specifications and so their depiction in the artefact model renders the model more explicit. |
| <i>'External interface specifications'</i> | External interface specifications denote that the Big data system will communicate with external components. |
| <i>'Data Requirements'</i> | Data Requirements are an inherent part of any Big Data system. |
| <i>'Business Case' along with its sub artefacts (Goals, Models, Plan, Customer and Stakeholders Needs, and Project definition).</i> | These are critical elements of any development project. |
| Relationships Added | Rationale |
| <i>'Assist-in'</i> | This relationship was added to represent the situation when one or more artefacts assist in the creation of one or more other artefacts (e.g., system requirements in the creation of system architecture). |
| <i>'Is-composed-of'</i> | This relationship denotes the 'grouping' of artefacts (e.g., requirements specifications composed of functional and non-functional requirements). |
| Relationships Removed | Rationale |
| <i>'Is-identified-from'</i> | In improving the artefact model, this type of relationship was no longer needed. |
| Labels changed | Rationale |
| <i>'Data Capability Requirements'</i> (is changed to) <i>'Infrastructure Requirements'</i> <i>'Data Transformation Requirements'</i> (is changed to) <i>'Data Processing Requirements'</i> | These labels (promoted by NIST (2015)) were changed based on recommendations from the practitioners. |

These entities are depicted in the post-validation version of the artefact model in a rectangular shape with bold (darker) borders and integrated with the traditional entities (such as "System's Requirements Specifications", "System Architecture, Design Components, and Abstractions", etc.) by the six types of relationships depicted in the model and described in Section 2 and Table 7 of this paper.

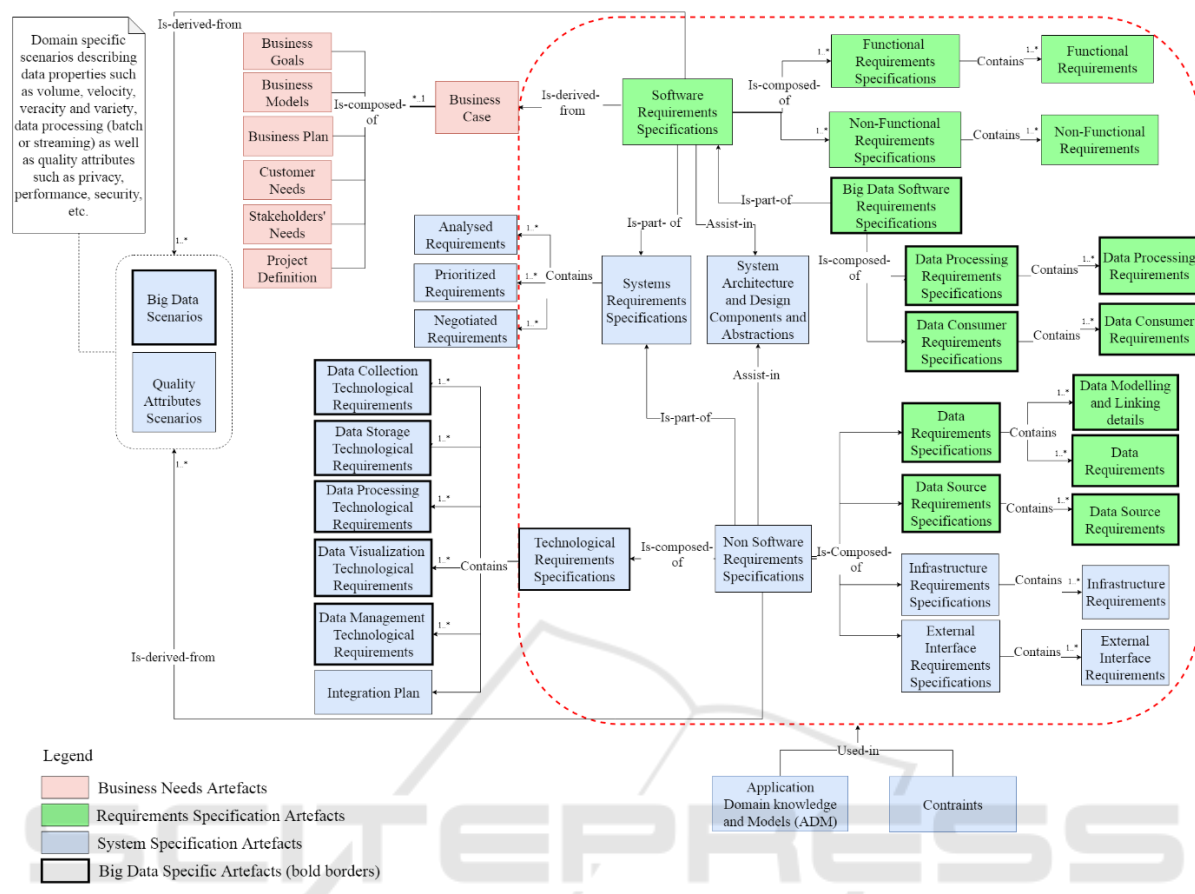
6 THREATS TO VALIDITY

We use Runeson and Host's (2009) guidelines to discuss the threats to validity and our approaches to mitigate them.

Construct validity is concerned with the extent the studied constructs represent their real-life meanings

(Runeson and Host, 2009). Given the large number of artefacts and relationships (i.e., constructs) in the artefact model, construct validity takes heightened importance. One threat to construct validity is in the model assessment. It is possible that the participants misunderstood our intent. To mitigate this threat, we provided the artefact model along with a definition of the elements and relationships in the instrument. We also briefed the model individually prior to assessment and were available for clarification during the study. There were no clarification incidents.

Threats to *internal validity* are concerned with confounding factors that may have influenced causal relationships in the study. Because our study does not involve causal relationships, this threat does not arise in the study.



This model depicts nodes of three types: (i) Business Needs, (ii) Requirements Specification, and (iii) Systems Specification. The rectangles with heavy border-lines are Big Data elements. The two blue rectangles labelled “ADM” and “Constraints” at the bottom of the figure are “Used in” every rectangle encapsulated inside the red boundary. They have been factored out to simplify the diagram.

Figure 2: Post-validation Big Data RE artefact model.

Reliability is concerned with whether the study can be repeated by other researchers and lead to the same results. This threat does exist for several reasons. For example, the participants’ background and experience would likely differ in another study and hence may induce variation in the results. Also, the questions in the instrument may be interpreted with variability. This threat was mitigated by using guidelines for instrument creation (Berdie et al., 1986). Also, the instrument was reviewed by all the three authors independently and any differences were resolved in consensus meetings over several iterations to ensure clarity and correctness. Another threat to reliability can result from the researcher’s subjective interpretation of the gathered data, leading to a biased artefact model. We addressed this threat by ensuring that all the artefact model elements are rooted in the scientific literature and data from actual

Big Data projects. Also, we used thematic coding, an established process for qualitative research.

External validity is concerned with generalisability of the artefact model. Of course, with ten participants, we cannot claim strong generalisability across a large body of Big Data software development projects. However, the varied sources from which we have constructed the model (i.e., literature, expert opinion, and Big Data projects) give a first solid basis for applicability of the model in other projects. Regardless, the user is recommended to exercise caution when using the model in a real-life project.

Selection bias is a possible threat in this study due to the use of convenience sampling for selecting participants. Such bias can skew the resultant artefact model. However, “practitioner knowledge and experience” from diverse real-world Big Data projects helps to mitigate this threat.

Experience bias exists towards early period (1-4 years) of the participants in Big data systems in industry. This threat remains at the early stage of the field of Big Data, but we hope that participants in the future studies on the artefact-model will have gained further experience to minimise this type of bias.

Finally, threats to *conclusion validity* are concerned with whether conclusions are traceable to the findings (Runeson and Host, 2009). This threat is considered contained since all the conclusions presented are shown to have been rooted in specific sections of this paper.

7 CONCLUSIONS

Whereas much attention has been given to analytics concerning Big Data, little amount of attention has been invested in the development of software applications and services leveraging Big Data. This situation is also reflected in the field of RE where domain models, processes, methods, techniques and tools have not yet embraced Big Data in a significant way. To ameliorate this situation, in 2017, we had created a preliminary RE artefact model to aid the development of Big Data software applications (Arruda and Madhavji, 2017).

In this paper, we describe how we have taken the early result to the next level by having the model validated by ten third-party practitioners from diverse Big Data software development projects. Specifically, the model was validated on its qualities such as: accuracy, completeness, usefulness, and generalisability (see Section 4). This paper gives details of the validation study, such as descriptive statistics of the study participants and application domains in industry (see Subsection 4.1); data gathered and analysed (see Subsection 4.2); and the resultant, improved, artefact model (see Figure 2, Section 5).

The validation results indicate that the model captures the key RE artefacts and relationships of a Big Data software development project, currently lacking in the literature. The validation results also confirm consensus amongst the study participants regarding the usefulness and applicability of the model in practice (see Table 5, section 4).

This research is not terminal. Further possibilities include: (1) enhancement of the model embracing new application domains, such as IoT (internet of things); (2) empirical studies of the application of the model in Big Data projects to further assess the model's adaptability and generalisability; and (3) cost

analysis of adopting the artefact model in industry projects.

ACKNOWLEDGEMENTS

This research is supported, in part, by grants from CNPq, The National Council of Technological and Scientific Development – Brazil (process: 200218/2015-8) and NSERC, Natural Science and Engineering Research Council of Canada. We are very thankful to the industry practitioners that took part in the validation study. Finally, we are grateful to the reviewers for their feedback, which has improved the paper.

REFERENCES

- Anderson, K.M., 2015. Embrace the Challenges: Software Engineering in a Big Data World. In *Proc. BIGDSE 2015 1st Int. Work. Big Data Softw. Eng.* 19–25.
- Arruda, D., Madhavji, N.H., 2018. State of Requirements Engineering Research in the Context of Big Data Applications. In *Proc. Requir. Eng. Found. Softw. Qual.* Springer Nature.
- Arruda, D., Madhavji, N. H., 2017. Towards a Requirements Engineering Artefact Model in the context of Big Data Software Development Projects. In *IEEE International Conference on Big Data*, pp. 2232–2237.
- Berdie, Douglas R., Anderson, John F., Nibuhr, Marsha A. 1986. Questionnaires: Design and Use. Scarecrow Press, Metuchen, NJ, 2nd edition.
- Berenbach, B., Paulish, D.J., Kazmeier, J., Rudorfer, A. 2009. *Soft. & Systems Requirements Eng. In Practice.* McGraw-Hill, New York.
- Davenport. T.H., Bean, R. 2019. *Big Companies Are Embracing Analytics, But Most Still Don't Have a Data-Driven Culture (Online)*. Available at: <https://hbr.org/2018/02/big-companies-are-embracing-analytics-but-most-still-dont-have-a-data-driven-culture>. Accessed March 20th, 2019.
- Dalpiatz, F., Ferrari, A., Franch, X., Palomares, C., 2018. Natural Language Processing for Requirements Engineering The Best Is Yet to Come. *IEEE Softw.* 35, 115–119.
- Dipti Kumar, V., Alencar, P., 2016. Software engineering for Big Data projects: Domains, methodologies and gaps. In *Proc. - 2016 IEEE Int. Conf. Big Data*, 2886–2895.
- Geisberger, E., Broy, M., Berenbach, B., Kazmeier, J., Paulish, D., Rudorfer, A., 2006. *Requirements Engineering Reference Model (REM)*.
- Gibbs, G.R. 2007. Analyzing qualitative data, Qualitative research kit, SAGE Publications, Ltd, London, England.

- Horkoff, J., Aydemir, F.B., Cardoso E., Li, T., Mate, A., Paja, E., Salnitri, M., Piras, L., Mylopoulos, J., Giorgini, P. 2017. Goal-oriented requirements engineering: an extended systematic mapping study. *Requirements Engineering*. 24, 133-160.
- Humphrey, W.S., Kellner, M.I., 1989. Software Process Modeling: Principles of Entity Process Models Software Process. In *Proc. 11th International Conference on Software Engineering*, Pittsburgh/USA, 331-342.
- Lamsweerde, A.V. 2009. *Requirements Engineering: from system goals to UML models to software specifications*. Wiley. England, 1st edition.
- Laigner, R.N., Kalinowski, M., Lifschitz, S., Monteiro, R.S., Oliveira, D. De, 2018. A Systematic Mapping of Software Engineering Approaches to Develop Big Data Systems. In *Proc. 44th Euromicro Conference on Software Engineering and Advanced Applications*, 447-453.
- Likert, R., 1932. A technique for the measurement of attitudes. *Arch Psychol*, 5-55.
- Madhavji, N.H., Miransky, A., Kontogiannis, K., 2015. Big Picture of Big Data Software Engineering: With Example Research Challenges. In *Proc. BIGDSE 2015 - 1st Int. Work. Big Data Softw. Eng.* 11-14.
- Méndez Fernández, D., Lochmann, K., Penzenstadler, B., Wagner, S., 2011. A Case Study on the Application of an Artefact-Based Requirements Engineering Approach. In *Proc. of 15th Annual Conference on Evaluation & Assessment in Software Engineering* 104-113.
- Méndez Fernández, D., Penzenstadler, B., 2015. Artefact-based requirements engineering: the AMDiRE approach. *Requir. Eng.* 20, 405-434.
- Nadkarni, A., Vesset, D. *Worldwide Big Data Technology and Services Forecast, 2016-2020* (Online). Available at <https://www.marketresearch.com/IDC-v2477/Worldwide-Big-Data-Technology-Services-10510864/>. Accessed March 20th, 2019.
- Nekvi, M.R.I., Madhavji, N.H., 2014. Impediments to Regulatory Compliance of Requirements in Contractual Systems Engineering Projects. *ACM Trans. Manag. Inf. Syst.* 5, 1-35.
- NIST Big Data Public Working Group: *Use Cases and Requirements Subgroup*, 2015. NIST Big Data Interoperability Framework: Volume 3, Use Cases and General Requirements 3, 260.
- Penzenstadler, B., Fernandez, D.M., Eckhardt, J., 2013. Understanding the impact of artefact-based RE - Design of a replication study. In *Proc. Int. Symp. Empir. Softw. Eng.* 267-270.
- Runeson, P., Host, M., 2009. Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* 14, 131-164.
- Selic, B., 2003. The pragmatics of model-driven development. *IEEE Softw.* 20, 19-25.
- Shaw, M., 2003. Writing Good Software Engineering Research Papers. In *Proc. 25th International Conference on Software Engineering*, 726-736.
- Sommerville, I. 2009. *Software Engineering*. Pearson, Boston, Massachusetts, 9th edition.