Requirements Specification using Fact-oriented Modeling: a Case Study and Generalization

Gabor Melli

Affiliation: Prediction Works, Inc.

Email: gmelli@predictionworks.com

Jerre McQuinn

Affiliation: Microsoft Corporation

Email: jerrem@microsoft.com

Requirements Specification using Fact-oriented Modeling: a Case Study and Generalization

Abstract. We present a case study of the application of fact-oriented modeling to the capture and management of requirement specifications for the introduction of an information technology solution within Microsoft. The delivered solution involves automation and centralization of information about relationships between Microsoft product offerings. The methodology contributed to the project's fast turn-around time and high quality deliverable largely due to the clarity, completeness and traceability of business concepts and individual specification statements. We conclude with a generalization of the methodology used.

1 Introduction

Fact-oriented Modeling[1] [3], [4], [7] is a technique that assists with the conceptual modeling of an IT Solution. The approach however has not yet been fully incorporated into software requirement specification standards [8], [9], [10], [12], [13], [14], [2]. With the introduction of such standards as Structured Business Vocabulary and Rules (SBVR) [5], [7] it is now possible to consistently employ Fact-oriented Modeling in the delivery of enterprise solutions. In this paper we intend to illustrate, via a case study, how a Solution Requirements Specification Methodology can integrate Fact-oriented Modeling with a classic requirements specification approach.

Fact-oriented Modeling depends upon a controlled vocabulary of Business Concepts which can be used by business and IT stakeholders to communicate in a common language, leaving little room for ambiguity. Many Microsoft legacy systems have physical data structures that do not reflect the business concepts and relationships that they support. Fact-oriented Modeling changes this paradigm by requiring that Business Concepts and the allowed actions and relationships between them are specified as Business Rules *before* the functional specification begins.

In Microsoft IT, we are adopting a Structured Requirements Management (STREAM) [1] approach to documenting Specification Items through the use of Fact-oriented Modeling and by including the explicit identification of Business Concepts, Business Facts, and Business Rules in RuleSpeak® [6] notation. The authors, following STREAM guidelines, recently employed Fact-oriented Modeling to the specification of a new IT application, named PReM, to centrally manage product relationships. We represented the solution's Business Requirements by means of Specification Items that are atomic, itemized, prioritized, and written in a Structured Language that is understandable to both Business Stakeholders and IT Stakeholders.

Perceived benefits from the use of the approach included:

- Specification Statement Clarity
  - Reduced guesswork in picking business concepts, facts, rules or requirements from paragraphs of text.
  - Encouraged the graphical depiction of the relationship of Business Concepts.
  - Increased the accuracy of ensuing analyses (functional, technical, and test case analyses).
  - Promoted faster production of functional and technical specifications, even when the project is outsourced and off-shored. [15].

- Specification Statement Traceability
  - Enabled faster and more accurate determination of coverage by functional and technical specifications and test cases.
  - Sped up issue resolution because facts were interconnected with the logical data model and the physical data model.

In section 2 of this paper, we present PReM as an illustrative case study of a successful integration of Fact-oriented Modeling into requirements specification. In section 3, we then generalize, using Fact-oriented Modeling to illustrate the Solution Requirements Specification Methodology (SRS) process itself. Sections 4 and 5 then describe the benefits and make concluding remarks.

---

[1] Key terms are defined in the factmodels.com/PReM1/v060930 and factmodels.com/SRS1/v060930 repositories.

2 Case Study

This section presents a case study of the Product Relationship Management (PReM) application development, begun in the fall of 2007, to provide relationships between Microsoft products in computer-consumable format. Prior to this project, the relationships were either solely in legal documents or were maintained in an assortment of consuming systems.

The project used a classic waterfall Software Development Lifecycle Methodology (SDLC), described below, but the collection of *Specification Statement*s would have equally well enabled an agile Scrum Methodology.

2.1 Vision-Scope Phase

The PReM Vision-Scope phase documented the business opportunity, constraints, solution alternatives, costs and benefits. It identified the scope of a recommended solution and a roadmap to realize the solution over multiple releases.

PReM set out to address the opportunity to better manage interrelationships between Microsoft products to help customers make optimum purchase decisions. For example, customers need to know that if they have Office Standard Edition L&SA they may "step up" to Office Professional Plus (Figure 1). Currently these product relationships are documented in policy documents, such as the Product Use Rights[2]. The PReM vision is to systematically deliver a centralized master service for all Product Relationships that impact customer decisions.

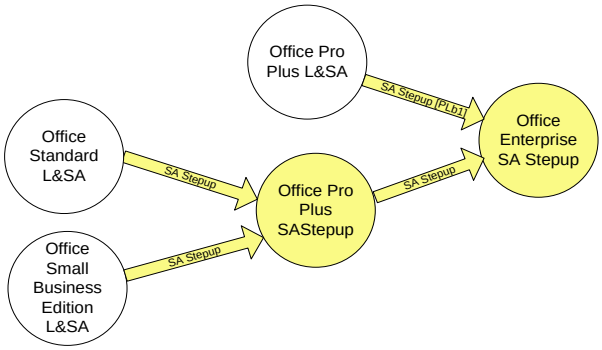

Figure 1 Fact Model for Office SA Step-Up Relationships.

We began with Information Analysis of policy documents, of master product data, and of experts' knowledge. We quickly realized that the relationship instances were too numerous to be accurately created and maintained manually. In a typical case, each of 150 product part numbers (SKUs) relates to each of 150 different SKUs to generate 22,500 individual relationship combinations. There are hundreds of such cases and so we concluded the need to automate the relationship management using descriptive attribution for each SKU.

Further analysis gave insight to abstractions that were not rendered in the initial fact model. For example, we immediately saw that we needed the concept of Product Collection and Product Relationship Rule and then we realized that a Product Collection can be an Antecedent Product Collection or a Consequent Product Collection in a Product Relationship Rule. A snippet of the fact model is shown in Figure 2.

---

2 http://www.microsoftvolumelicensing.com/userights/DocumentSearch.aspx?Mode=3&DocumentTypeId=1

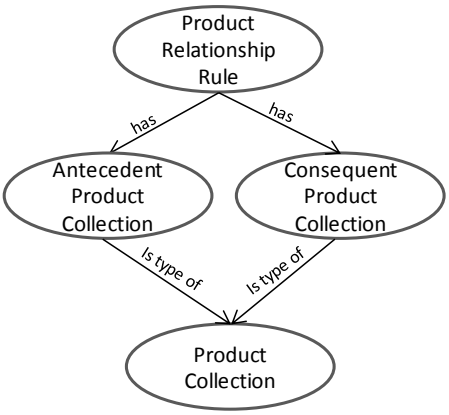

Figure 2 Fact Model Snippet

Since the volume of relationships requires automation, and because relationships undergo changes with the on-going launch and retirement of products, we specified a system which implements Business Rules of the following canonical form:

If a customer has/wants to install/order an offering from product collection *C* then, according to relationship type *T*, the customer is required/entitled/recommended to install/order/have acquired/align to offering(s) from product collections (A_1 AND A_2 AND …) OR … OR (…AND A_{n-1} AND A_n)

We refer to collection A as the as the "antecedent collection" and collection C as the "consequent product collection." Below is a sample rule instance:

If a customer wants to order an offering "Office Professional Plus SA Step-Up" then, according to relationship type "SA Stepup", the customer is required to have acquired an offering from product collection "Office Standard" or "Office Small Business Edition".

For example, members of one Antecedent Product Collection include:

021-05441	Office Standard Arabic Lic/SA Pack MVL
021-06785	Office Standard Brazilian Lic/SA Pack MVL
021-07816	Office Standard Bulgarian Lic/SA Pack MVL

These are related via the SA StepUp rule to members of the Consequent Product Collection which includes:

269-07501	Office Professional Plus Single Language SA StepUp MVL from Office Std

Step-Up rules are shown as a Fact Model in Figure 2 where each circle represents a Product Collection. It illustrates that Office Pro Plus SA-Step-Up collection is a Consequent Product Collection for Office Standard L&SA and is also an Antecedent Product Collection for Office Enterprise SA-Step-Up under the SA-Stepup Product Relationship Rule. A few similarly detailed Fact Models served to illustrate the concepts and their abstraction to business stakeholders.

In the Vision-Scope phase we began to capture Business Concepts and Business Facts. It was during the Requirements Phase that we made a methodical and diligent commitment to expressing the solution as Specification Items.

2.2 Requirements Phase

The requirements phase completes the Information Analysis, and performs Use Case Analysis and Business Process Analysis to produce a Business Requirements Document (BRD). We documented Business Concepts, and rendered the facts textually as well as graphically.

For example the "Product Collection" concept was presented in the BRD as:

> A PReM Product Collection is a collection of Products.
>
> **AKA**: Product Collection, Collection
>
> **Context**:
>
> - BR-121 (Fact) A Product Collection can have: Identifier, Name, Long Name, Effective Start Date, Effective End Date.
> - BR-122 (Fact) A Product Collection can contain zero or more Products.
> - BR-123 (Fact) A Product Collection can act in the role of a Consequent Collection.
> - BR-124 (Fact) A Product Collection can act in the role of an Antecedent Collection.
> - BR-108 (Rule) A Product Collection must be either an Attribute-based Collection or a Composite Collection.

The numbers were added near the end of the requirements analysis, once they were added into a master spreadsheet.

Use Case Analysis followed from a fact model (Figure 3) which relates actors to Business Concepts discovered during Information Analysis. This analysis discovered new concepts and facts and Business Rules, expressed as RuleSpeak®.

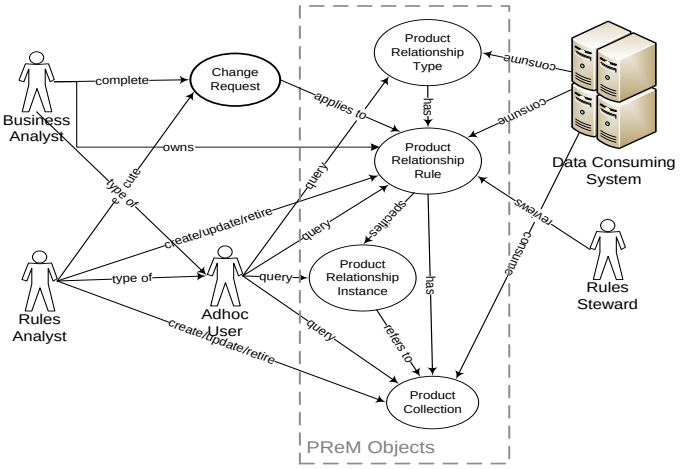

Figure 3 Use-based Fact Model

Each role in the Use Case Analysis was described in a fact-oriented manner. Users found it easier to understand this fact model when we used person-icons, rather than bubbles for role concepts. The description of roles, such as the Rules Analyst, was critical to adoption of new responsibilities in operational teams. For example, the Rules Analyst role of the PReM solution was described as follows:

A PReM Rules Analyst is a person who manages the Product Relationships that are mastered in the PReM System.

AKA: Rules Analyst, Product Relationship Rules Analyst

Context:

- Can validate that rules are complete, non-redundant, and non-conflicting.
- Can facilitate term/rule validation sessions with stakeholders.
- Can maintain traceability to business rules
- Can liaise with Data Governance and with IT staff.
- Can resolves cross domain conflicts relating to business rules.
- Must have the following skills
  - Deep knowledge of product launch processes
  - Understanding of data governance and exception management
- BR-3 (Fact) A Rules Analyst can receive and execute Change Requests.
- BR-14 (Fact) A Rules Analyst can modify a PReM Object.
- BR-16 (Rule) Only a Rules Analyst can modify a PReM Object.
- BR-184 (Fact) A Rules Analyst can update a PReM Object from Draft status to Pending Approval or Cancelled.
- BR-185 (Fact) A Rules Analyst can update a PReM Object from Pending Approval status to Canceled.

See: PReM Business Analyst

We performed Business Process Analysis to specify the management of the new Product Collections and Product Relationship Rules. The "Define Relationship process included steps for a Rules Analyst to receive a request, encode a Product Collection, encode a Product Relationship Rule, and commit them both to production. Through Business Process Analysis we discovered more concepts, facts, rules and requirements. Examples include:

Rule-108: A Product Collection must be either an attribute-based collection or a composite collection, composed of other collections.

Requirement-66: Solution shall provide an interactive Attribute-based Collection creation capability for a Rules Analyst to choose attributes, and then select a list of values available for that attribute.

Finally, we compiled all of the facts, rules and requirements into an Excel spreadsheet and gave each item a numeric identifier, alignment with process step, identification of type (Fact, Rule, Requirement), and a priority. This table became the master of the information. (Figure 4).

The management of the spreadsheet impelled one of the authors (Melli) to develop an Access database. Requirements materialized for new capabilities of the Access database which in turn led to the application of Fact-oriented Modeling to a generalized Solution Requirements Specification Methodology, described in section 3.

ID	Process	Step	Type	Priority	Requirement Definition
108	Define Relationship	1.1	Rule	1	A Product Collection must be either an Attribute-based Collection or a Composite Collection.
123	Define Relationship	1.2	Fact	1	A Product Collection can act in the role of a Consequent Collection.
124	Define Relationship	1.2	Fact	1	A Product Collection can act in the role of an Antecedent Collection.
66	Define Relationship	1.1	Reqt	1	Solution shall provide an interactive Attribute-based Collection creation capability for Rules Analyst to choose attribute, and then select a list of values available for that attribute.

Figure 4 Sample of the specification items master list within the BRD.

2.3 Design Phase

When the requirements phase completed, the direct involvement of the authors turned from a primary role to a supporting role. The project was designed and built by an offshore-outsourced vendor solicited by means of a Request for Proposals (RFP) package. [15] Bidders were supplied with the BRD and it appeared that the fact modeling technique improved the quality of the proposals we received.

The design phase of the project resulted in a Functional Specification Document that included a Logical Data Model (LDM) and a Technical Specification Document that included a Physical Data Model (PDM). The Business Concepts translated directly into entities in the LDM and the PDM. (Figure 5)

2.4 Build Phase

The build phase results in the delivery of working code. The developers referenced the fact model in this phase.

2.5 Testing Phase

An independent separate team wrote User Acceptance Test (UAT) cases. Fact models assisted UAT authors and testers in rapidly grasping the relationship of Business Concepts to each other. UAT was executed in a shorter time, with significantly fewer bugs than projects of similar size and complexity. We maintained test cases in Microsoft Visual Studio with Specification Traceability from the BRD. (Figure 5)

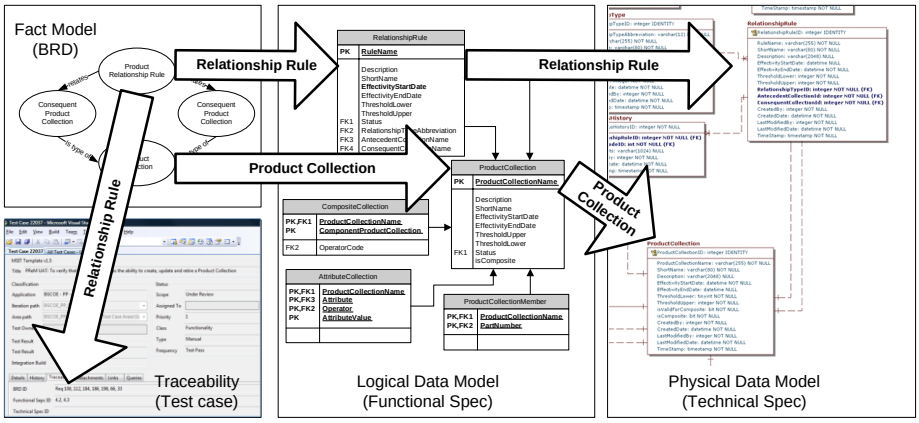

Figure 5 Fact-oriented Modeling enabled Business Concepts to persist through system analysis, design and test cases without guesswork and without loss in translation.

3 Methodology

This section presents a generalization of the solution requirements specification methodology (which we refer to as SRS) that was employed in implementation of the PReM project described in the use case. The aim of this section is to help the reader apply some or all of the methodology to their needs.

To illustrate how the methodology could be applied to another endeavor we present the SRS methodology using a structure similar to the Business Requirements Document structure for the PReM project. A solution is described along two main subject areas: how agents interact with the business concepts (Use Case Analysis) and how business concepts relate to each other (Information Analysis). Each subject area includes a brief descriptive narrative, a fact model, and a description of the concepts included in the fact model.

3.1 Use Case Analysis

This section describes the agents in fact modeling, and the tasks they perform. This approach is consistent with the Business Analysis Body of Knowledge with one or more Business Analysts acting in different roles. [10] A Business Process Analyst produces "as-is" and "to-be" processes flowcharts to identify process improvements to be automated. From Business Process Analysis they specify Business Requirement which may be further described via Use Cases.

Sometimes, Business Process Analysis must be augmented by Information Analysis. A Business Information Analyst might be specialized at performing database queries or experienced at policy interpretation.

The fact model in Figure 6, followed by definitions, illustrates the requirements and rules discovery activities.

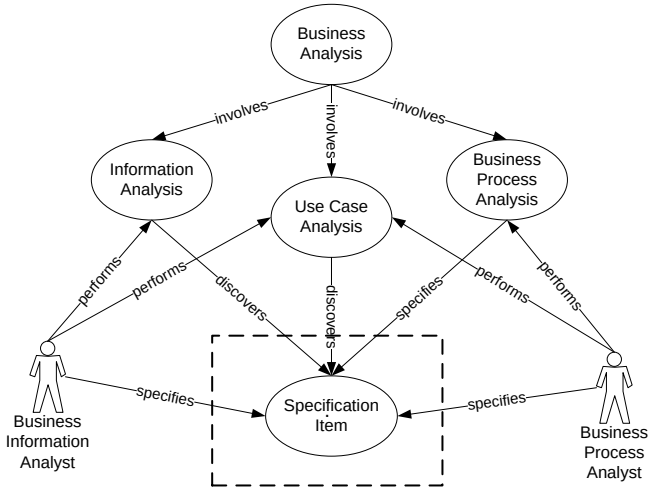

Figure 6 A fact model focused on the users of the methodology and what they can affect. The node enclosed in the rectangle is described in section 3.3.

Name	***Definition***
Business Analyst	A person who analyzes business needs to identify business problems and propose solutions. **Example(s):** • A person who produces Business Requirement Statements **Fact(s):** • A Business Analyst can Create, Delete, Modify a Business Concept.
Business Information Analyst	A Business Analyst who focuses on focuses on Business Information Analysis to derive Specification Items. **Example(s):** • An analyst who reviews contractual documentation for policy compliance. • An analyst who performs SQL queries of transactional data.
Business Opportunity	An opportunity for improving a business process or achieving a business goal. **Example(s):** • Business partners need to understand product relationships so as to recommend better solutions to customers.
Business Process Analysis	An analysis that decomposes relevant processes into discrete steps to produce a better understanding of the functions performed. **Example(s):** • A Rules Analyst receives a request to encode a new Business Rule.

Business Process Analyst	A Business Analyst who focuses on processes and use cases to derive Specification Items. **Example(s):** • A Six Sigma Green Belt who produces process diagrams to assist in process improvement.
Information Analysis	An analysis that requires the decomposition of complex knowledge contained in policies, documents, and data into smaller parts to identify patterns which can be encoded as rules. **Example(s):** • Customers who acquired a License with maintenance can renew the maintenance when their agreement expires.
Use Case	A description of a system's behavior as it responds to a request that originates from outside of that system. **Example(s):** • Use Case 1 - A data steward needs approve a change to a data object.
Use Case Analysis	An analysis that defines user interaction with processes and systems to produce a better understanding of the required system components.

3.2 Information Analysis

This section describes the Information Analysis we performed to document the SRS methodology. We include a fact model (Figure 7) of the information objects and we supply definitions for the concepts in the model[3]. Although for PReM we did not identify different kinds of requirements, the figure shows how requirements might be classified into Business Requirements, User Requirements, etc. Classification of requirements varies somewhat between standards [10], [12]. [14].

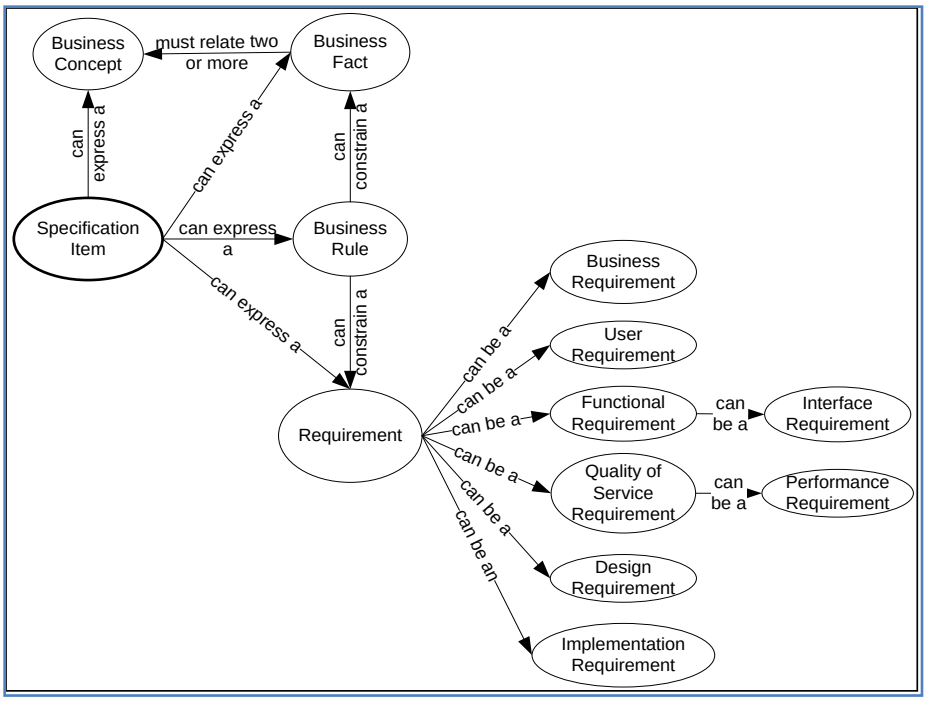

Figure 7 Concept Model of the SRS Methodology

[3] The master repository for the methodology described in this paper can be found at http://www.factmodels.com/SRS1/v080630/SrsMgr1_SrsDb.accdb.zip.

The following table describes the concepts in the figure above.

Name	Definition
Business Concept	A Concept that is under the jurisdiction of a Business Group. **Fact(s):** • Can have Subject Areas.
Business Fact	A Fact that is under the jurisdiction of the Business Group. **Example(s):** • A Product Collection can act in the role of an Antecedent Collection. **Fact(s):** • Can be a Business Fact. • Can express a Business Fact. **Rule(s):** • Must relate to more Business Concepts. • Must constrain a Business Fact
Business Requirement	A condition or capability needed by a Business Group to achieve a Business Objective. **Example(s):** • A Product Collection is a retired Product Collection when its Effectivity End Date is in the past. **Fact(s):** • A Relationship Type can be a Business Requirement. • A Requirement can be a Business Requirement.
Business Rule	A Rule that is under the jurisdiction of a Business Group. **Example(s):** • Only a Rules Analyst may create a Product Relationship Rule. • Counter-Example: Business Regulations (e.g. Sarbanes-Oxley) are not business rules; but business rules can be put in place to comply with them. • Counter-Example: Laws of physics. • Counter-Example: Adopted external standards (e.g. Semantics of Business Vocabulary and Business Rules). **Fact(s):** • Can be an Operative Business Rule that conveys obligation. • Can be a Structural Business Rule that conveys necessity. • Can be styled as Prefixed Rule Keyword Style or Embedded (mixfix) Rule Keyword Style • A Specification Item can express a Business Rule.
Design Requirement	A Requirement that specifies the design of a system. **Example(s):** • Application needs .NET framework 3.0 on Web Server. **Fact(s):** • A Requirement can be a Design Requirement..
Functional Requirement	A Requirement that specifies a specific behavior or operation that a System must be able to perform. **Example(s):** • Solution shall retain committed PReM Objects indefinitely. • Solution shall provide the ability to associate notes to a project plan. • Solution must allow the user to enter free text to the project plan notes, up to 255 characters in length. **Fact(s):** • Can be an Interface Requirement.. • A Requirement can be a Functional Requirement.

Implementation Requirement	An Implementation Requirement is a Requirement of the Solution's Implementation Phase (transition from the Current State to the desired Future State). **Context:** • Implementation requirements describe capabilities that the solution must have in order to facilitate transition from the current state of the enterprise to the desired future state, but that will not be needed once that transition is complete. (IIBA - BABOK) **Example(s):** • "Solution shall provide User Interface that operates on a Windows XP or Windows Vista client." **Fact(s):** • A Requirement can be an Implementation Requirement.
Interface Requirement	A Functional Requirement that specifies how a Solution must interact with some external System. **Example(s):** • "Solution shall deliver the following data to consuming systems: Product Collections (ID, Name, Long Name, Effective Start Date, Effective End Date, Items)." • "Solution shall deliver the PReM data to the data warehouse no later than midnight PDT on the last day of the calendar month." **Fact(s):** • A Functional Requirement can be an Interface Requirement.
Performance Requirement	A Quality of Service Requirement that constrains the maximum period of time between two Events. **Example(s):** • "Solution shall provide response time to user action on User Interface within 5 seconds." **Fact(s):** • A Quality of Service Requirement can be a Performance Requirement.
Quality of Service Requirement	A Requirement that describes environmental conditions under which the Solution must remain effective. **Example(s):** • "Ordinary system availability shall be 24 hours/day x 7 days/week with the exception of scheduled downtime." **Fact(s):** • A Requirement can be a Quality of Service Requirement. • A Quality of Service Requirement can be a Performance Requirement..

Requirement	Something that is required from a Business Solution. **Context:** • "A requirement is a condition or capability needed by a user to solve a problem to achieve an objective which has to be documented." (IEEE 830-1998) **Example(s):** • Solution shall provide the ability to add notes to a project plan. (Functional Requirement) **Fact(s):** • Can be a Functional Requirement. • Can be a Design Requirement. • Can be an Implementation Requirement. • Can be a Quality of Service Requirement. • Can be a Business Requirement. • Can be a User Requirement. • A Specification Item can express a Requirement. • A Business Rule can constrain a Requirement.
User Requirement	A Requirement that specifies how a Stakeholder will interact with the Solution. **Example(s):** • Solution shall enable a Rules Analyst to interactively create, update or retire a Relationship Rule. • Solution shall provide an interactive Attribute-based Collection creation capability for a Rules Analyst to choose attributes, and then select a list of values available for that attribute. **Fact(s):** • A Requirement can be a User Requirement.

3.3 Specification Item

The central concept of the methodology is a Specification Item which provides an atomic, itemized and prioritized aspect of a Solution that is expressed in a Structured Language and is under the jurisdiction of the Business. We present it as a fact model in Figure 8 (followed by its definition).

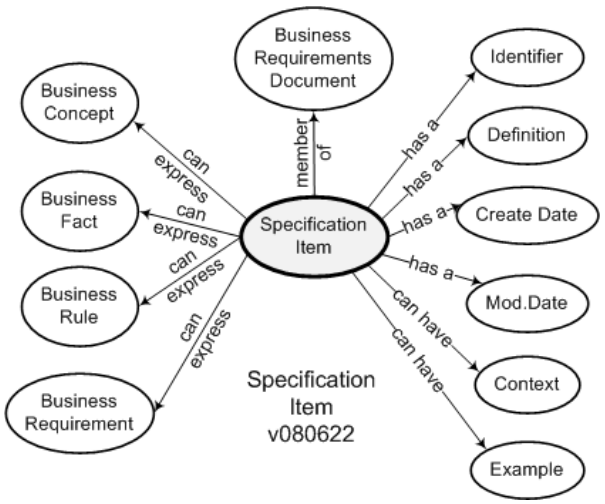

Figure 8 Fact Model of a Specification Item

Name	*Definition*
Specification Item	An atomic, itemized and prioritized aspect of a Solution that is expressed in a Structured Language and is under the jurisdiction of the Business. **Fact(s):** • Can express a Requirement. • Can have a Subject Area. • Can have an Identifier. • Can have a Priority. • Can express a Business Concept. • Can express a Business Rule. • Can express a Business Fact. • Can have a Specification Statement. • Can have a Context Statement. • A UAT Test Case can validate a Specification Item. **Rule(s):** • Must have one and only one Identifier. • Must have one and only one Last Modified Date. • Must have one and only one Definition Statement. • Must have one and only one Created Date.

4 Benefits Analysis

4.1 Summary of Objects

This section presents a quantitative analysis of the number of information objects and relations that were generated as a result of the requirement specifications. While we have limited comparative metrics, the table below shows a rough comparison with another project of similar size and complexity, managed by one of the authors (McQuinn) [11].

The comparison project employed less-rigorous fact modeling and experienced greater change, measured by BRD iterations and requirements change after approval (in bold outline). It is the authors' feeling that greater rigor contributed substantially to better quality and faster delivery. This is corroborated by benefits listed in the following section.

Item	PReM	Comparison project
Concepts	33	22
Facts	40	23
Rules	36	29
Requirements	70	114
Processes	3	4
Process Steps	18	18
BRD pages	78	66
BRD revisions	**6**	**13**
Percent Requirements change after approval	**2%**	**17%**
User Acceptance Test bugs	**7**	**Not applicable**

4.2 Benefits

Anecdotal benefits were collected from the BRD consumers, including the systems analysts, designers, test engineers, and from the authors' own observations. We were finishing this paper as the PReM system went into production, and we were surprised to continue finding benefits that resulted from a clear and well-organized BRD. Nearly all users pointed to the fact model diagrams as helpful in their initial understanding of the requirements.

- Fact modeling kept us focused on Business Concepts, reducing the temptation to dive into logical or physical modeling in the requirements phase.
- Fact modeling provided a framework and rigor to identify and focus on Business Rules in context, rather than adding them after the fact to fill a separate BRD chapter.
- The BRD was included in the vendor solicitation RFP. The selected vendor stated that the fact models helped in more quickly preparing a more accurate proposal.
- Systems analysts remarked on the readability and clarity of the BRD. The PReM analyst produced only 24 formal questions compared to more than 100 for other BRDs of similar size and complexity.
- The BRD and functional specification were each completed in 1 month, compared to typically 6 or more weeks each. The functional and technical specification proceeded in parallel, partly because of strong fact models to guide both.
- The India-based vendor reported that throughout the design and build phase they continually referred to the Specification Items by number, both increasing accuracy and decreasing turnaround on questions. The communication would otherwise have been belabored by a 12.5 hour time difference.
- Fact modeling standardized names for critical Business Concepts and those names persisted through systems analysis, design, test, and user documentation. The consistency aided in communication and searchability of all documentation.

4.3 Barriers to adoption

Perceived barriers to adopting the approach include:

- The effort required to integrate the approach into an organization's existing Software Development Lifecycle Methodology (SDLC) may require modification of the format or usual content expected in the requirements documentation.
- It may be more difficult for the business stakeholders to grasp the big picture. They may miss the business narrative if it is not presented in story format.
- There may be some initial resistance by the business to the development and use of a controlled vocabulary for Business Concepts and precise and sufficiently abstract Business Rules.

5 Conclusion

Fact-based modeling was successfully applied to the delivery of a new IT solution at Microsoft. The approach followed sped-up all activities during all project phases and resulted in a higher quality (more accurate and consistent) solution.

Looking ahead we foresee long-term opportunity in:

1. Evangelizing within Microsoft IT in order to deliver future versions of the solution in response to changing business requirements,
2. Improving the accuracy and speed with which a Rules Analyst can speak with a Business Analyst to elicit Business Rules.

3. Establishing a repository of fact model patterns to enable re-use. Topic areas to assist enterprise solutions would include: Product, Customer, Pricing, Contracts, and Governance.
4. The alignment of fact modeling with other industry standards (UML diagramming) and to elicit non-rule-based requirements.
5. Automating the capture and management of specification items, including automatic assignment of identifiers; validation that every rule has a fact; validation that every concept is defined, and automated addition of hyperlinks from an MS-Word-based document to the requirements specification repository.

6 References

1. Petr Choteborsky, and Rik Gerrits: Business Rules Management without a Rule Engine: Does it make sense? Conference Proceedings 10th International Business Rule Forum. (2007)
2. Stijn Goedertier, Christophe Mues, and Jan Vanthienen: (2007). Specifying Process-Aware Access Control Rules in SBVR. Springer Berlin / Heidelberg. Advances in Rule Interchange and Applications. Volume 4824/2007
3. Terry Halpin: Business rules and Object-Role modeling. In Database Programming and Design, vol. 9, no. 10 (Oct. 1996), pp. 66-72.
4. Terry Halpin: (1989). A Logical Analysis of Information Systems: Static aspects of the data-oriented perspective. PhD Thesis.
5. Sjir Nijssen: SBVR ~ Ground Facts and Fact Types in First-Order Logic. In *Business Rules Journal*, Vol. 9, No. 1 (January 2008)
6. Ronald G. Ross: Principles of the Business Rule Approach. Addison-Wesley. (2003).
7. Object Management Group: Semantics of Business Vocabulary and Business Rules (2007).
8. Jos Vos: Is There Fact Orientation Life Preceding Requirements? On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops. (2007).
9. W. M. N. Wan-Kadir, and Pericles Loucopoulos: Relating evolving business rules to software design. Journal of Systems Architecture. Volume 50, Issue 7, July 2004, (2004). Pages 367-382
10. International Institute of Business Analysis (IIBA): Business Analysis Body of Knowledge (BABOK ®) v1.6. (2006)
11. Jerre McQuinn: Decision Tables-From Specification to Operation. Conference Proceedings 10th International Business Rule Forum. (2007)
12. IEEE: Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998. (1998)
13. Software Engineering Institute: Capability Maturity Model Integration (CMMI) for Development, Version 1.2. Technical Report CMU/SEI-2006-TR-008. (2006).
14. Karl E. Wiegers: Software Requirements, 2nd Edition. Microsoft Press. ISBN 0-7356-0631-5 (1999)
15. Charalambos L. Iacovou, and Robbie Nakatsu: A risk profile of offshore-outsourced development projects. Communications of the ACM. Volume 51, Issue 6 (June 2008)