

Designing UML Diagrams for Technical Documentation: Continuing the Collaborative Approach to Publishing Class Diagrams

Neil MacKinnon
IBM Toronto Software Laboratory
8200 Warden Avenue
Markham, Ontario L6G 1C7
1-905-413-5883
neilmack@ca.ibm.com

Steve Murphy
IBM Toronto Software Laboratory
8200 Warden Avenue
Markham, Ontario L6G 1C7
1-905-413-3863
srmurphy@ca.ibm.com

ABSTRACT

This paper provides an updated discussion of the authors' ongoing efforts in developing a design framework for UML diagrams in technical documentation.

UML diagrams are a key part of program design. They can enhance understanding of complex programming concepts, and assist in problem analysis and solution design. In a previous paper, "Designing UML diagrams for technical documentation" [1], the authors presented a collaborative process that applies established design principles to UML diagrams, improving diagram presentation and shrinking publication costs and schedules. This paper expands on that work, showing how the established process can now be applied to several different modeling tools, including IBM Rational® XDE and Microsoft Visio. It also provides a detailed description of how to export modeling files to other file formats. The authors show how their process workflow has evolved over the past year as it has been applied by developers, writers, and graphic designers at IBM's Toronto Software Laboratory, and discuss ways in which they will continue to monitor and improve the efficacy of both the process and resulting diagrams.

Categories & Subject Descriptors: H.5.2

[Information Interfaces and Presentation]: User Interfaces – *Screen design; Standardization; Style guides; Theory and methods; Training, help, and documentation*

General Terms: Design, Documentation, Human Factors, Standardization

Keywords: Unified Modeling Language, documentation, graphic design, human factors, visualization, UML diagrams, guidelines

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGDOC'04, October 2004, Memphis, Tennessee, USA.

Copyright 2004 ACM 1-58113-809-1/04/0010...\$5.00.

1. INTRODUCTION

The Media Design Studio (MDS) at the IBM Toronto Lab provides graphic design services for the various software products that are produced at the lab. Included in MDS' responsibilities is preparing various technical diagrams for inclusion in IBM documentation, both print and online. As the Unified Modeling Language (UML) gained in popularity as a design tool, MDS found that an increasing number of submitted UML diagrams were large, unwieldy, and unsuitable for formal publication. A set of guidelines was needed to assist developers and writers in preparing their UML diagrams for publication without having to utilize a graphic designer to re-create the original diagram. This set of guidelines would help reduce time and expense for documentation production schedules.

1.1 Common design problems in UML diagrams

The MDS Diagram database tracks every diagram submission at the IBM Toronto Lab. MDS has found that certain problems are continually repeated in UML diagrams submitted for publication. Due to the limitations of delivery method for the documentation, whether through HTML files (that is, IBM InfoCenters and online help systems), PDF files, or hardcopy manuals, MDS has been continually required to reformat diagrams to fit the medium. Common problems include:

- Diagrams too large to fit on a printed page or standard on-screen viewing area
- Unnecessary English text in diagrams, resulting in increased translation costs
- Disorganized, jumbled diagram design

The IBM Toronto Lab UML diagram workgroup was formed to address the various issues associated with the development and presentation of UML diagrams.

1.2 Guideline Goals

The UML diagram workgroup encourages writers and developers who submit UML diagrams for publication to implement our guidelines in order to:

- Create clear, well-organized diagrams
- Act as a quality-assurance checklist
- Save time and expense
- Minimize the role of the graphic designer
- Limit translation costs

In our previous paper (SIGDOC Conference Proceedings 2003, p. 105) , we described the process (posted on an IBM internal Web site) that the UML diagram workgroup created to help developers, writers, and graphic designers implement a logical class diagram workflow. Based on user feedback and improved diagrams, we achieved our primary goal.

This paper provides an updated discussion of the authors' ongoing efforts in developing a design framework for UML diagrams in technical documentation. We present a current overview of the process, and describe changes and refinements implemented over the past year.

2. NEW TOOLS, NEW DOCUMENTATION

In 2003, IBM acquired Rational Software, a company specializing in software development tools based on object-oriented design and UML. As part of the acquisition, IBM sought to integrate and adapt Rational methodology to its own software development, meaning the UML diagram workgroup needed to test Rational XDE, which is also part of IBM WebSphere® Studio Application Developer's Integrated Development Environment.

The most common tools currently used for designing UML diagrams at IBM are Rational Rose® and Rational XDE. Other design tools, such as Microsoft® Visio®, were also tested, as was RoseGraph, a free tool for converting MDL files to common image formats. The UML diagram workgroup evaluated each of these tools over the past year.

2.1 Rational Rose

In our previous workflow for class diagrams created using Rational Rose, we recommended that developers or writers use Adobe Acrobat Distiller or PDF Writer printer drivers to save UML diagrams to a manageable vector-based file format. However, not everyone has access to those printer drivers since it requires a license for Adobe Acrobat.

Another method for exporting a workable graphic file from Rose is using RoseGraph. RoseGraph enables you to export Rational Rose diagrams to multiple file formats, and can be downloaded from www.rationalrose.com/addins/rosegraph.htm. After evaluating this tool, the UML diagram workgroup decided that RoseGraph offers a more convenient file-export solution for the workflow than Acrobat, and so the following steps were added to the UML diagram process.

***Note:** Before you can complete these steps, you must install RoseGraph on your system. You can download a copy from the Web site listed previously.*

To export the diagrams in your MDL files to another format, launch Rational Rose, and complete these steps:

1. *To launch RoseGraph, double-click the RoseGraph.exe file. (If Rational Rose isn't running when you double-click the RoseGraph.exe file, you will get a message asking if you want to start a new instance of Rose. Click **Yes**.)*
2. *In the **Main** tab of the RoseGraph window, decide on the type of diagram you want to export, and then select whether you want to export all of that type of diagram, or selected diagrams, or none of the diagrams. (See Figure 1.)*

*Remember that in this example, you are exporting a single class diagram. If you want to export one type of diagram but your MDL file contains other types of diagrams, make sure you select **None** for the types of diagrams that you do not want in your export.*

3. *Click the tab for the type of diagram that you want to export. In the example, a single class diagram is being exported, so you will click the **Class Diagrams** tab.*
4. *In the navigation tree on the left side of the window, select the diagrams that you want to export, and then click the arrow to move the file into the **Selected** window (see Figure 2).*

Remember that RoseGraph will save the converted file to a folder that corresponds to the Rose view that contains the original diagram. For example, the Sample Class file selected in the image will be saved to a folder named Logical View

5. *When you have moved all the diagrams that you want to export into the **Selected** window, click **Action > Export**. The Export Options window opens.*
6. *Select the graphic format for your diagrams, and enter the root directory name where you want to save the converted files (see Figure 3). Click **OK**. RoseGraph converts the selected files to the new format. It saves the new files to folders that have the same name as the original view.*

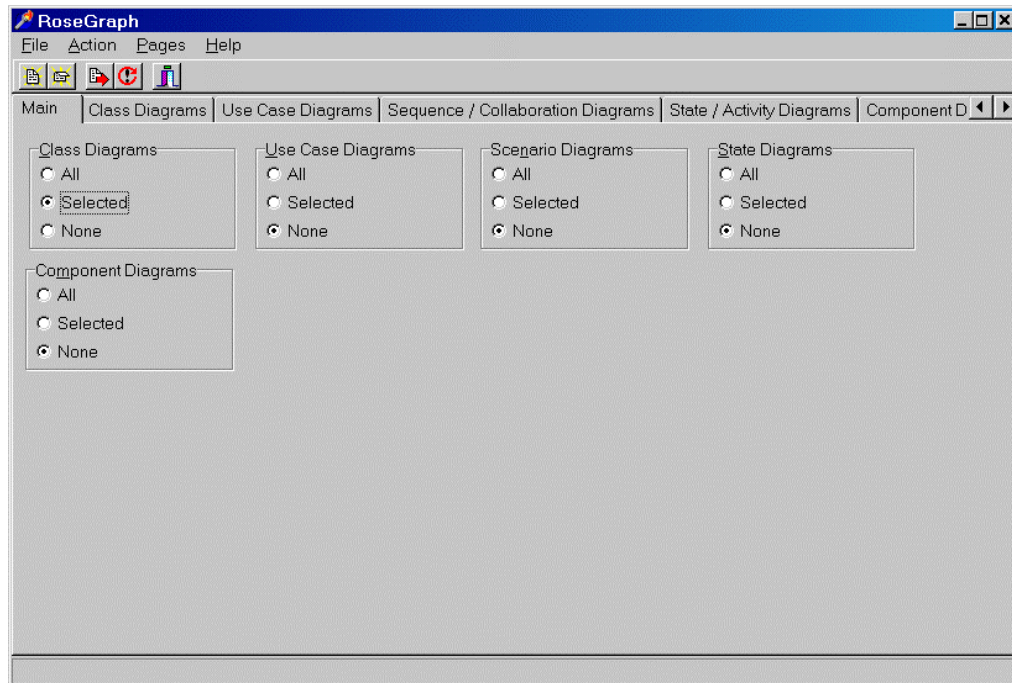


Figure 1: RoseGraph Main tab

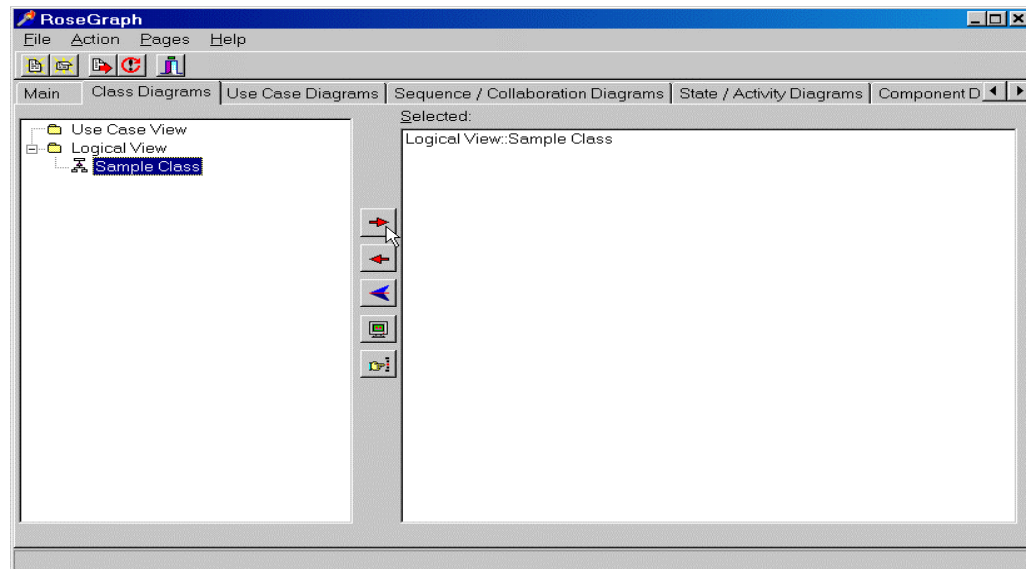


Figure 2: RoseGraph Class Diagrams tab

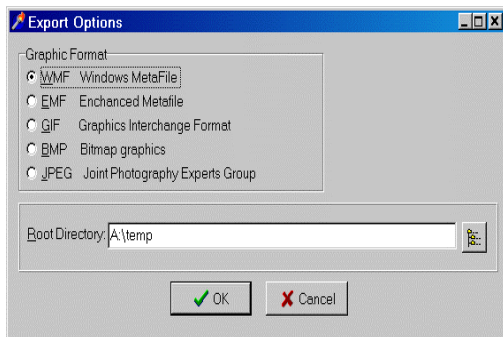


Figure 3: RoseGraph Export Options window

2.2 Rational XDE

XDE Modeler is similar to Rose Modeler, but can also serve as a plug-in to WebSphere Studio Application Developer. Both tools produce models based on UML and share the same default visual element properties. Because both tools produce similar output, the UML diagram workgroup assumed that the class diagram workflow could be easily adapted to include XDE.

However, in demonstrations of the tool and best practices for diagramming, a number of issues arose:

2.2.1 J2EE-specific artifacts

When you create a diagram using Rational XDE, J2EE-specific artifacts (see Figure 4) are included. (The current version of the

UML Diagram Process Web site does not describe a way to revise these images.) These artifacts can be large and obtrusive. The authors determined that the artifacts are essential to the meaning of many diagrams, but were difficult to edit. The workgroup discovered that Rational XDE cannot scale the images proportionately without causing distortions. Since clarity is a key goal of the process, the decision was made to leave the artifacts unchanged.



Figure 4: Rational XDE J2EE artifacts

2.2.2 XDE and RoseGraph compatibility

Testing revealed that while RoseGraph is able to export files from Rational Rose, it will not operate with Rational XDE. The UML diagram workgroup continues to investigate other approaches for exporting graphic files from XDE.

2.2.3 XDE vector file export

Ongoing conversations with Rational development offer no guidance regarding future vector file capability. XDE Modeler 2003 does not have the capability to save or export graphic files. Support for WMF (Windows Metafile), EMF (Enhanced Metafile), PDF (Portable Document File), and SVG (Scalable Vector Graphic), described in the class diagram workflow for Rational Rose, is not currently available for Rational XDE.

2.2.4 XDE and Rose compatibility

Rational XDE can open Rational Rose's file format (MDL), but Rose cannot open XDE's native file format (MDX). You cannot use the Rose workflow as a guide for importing the XDE modeling file and then exporting a graphic file. Although XDE can export XMI files (a form of XML), Rose does not have the capability to read those files.

2.2.5 Graphics export solution

The UML diagram workgroup was able to implement a workaround solution for exporting graphics files from XDE. Borrowing from one of the Rose workflow steps, the workgroup adapted the Rose PDF export process to XDE. Following are the step-by-step instructions incorporated in the process:

1. Finalize the diagram's layout in XDE, then click **File > Print**. Select **PDF Writer** from the printer name menu (see Figure 5). Click **OK**.

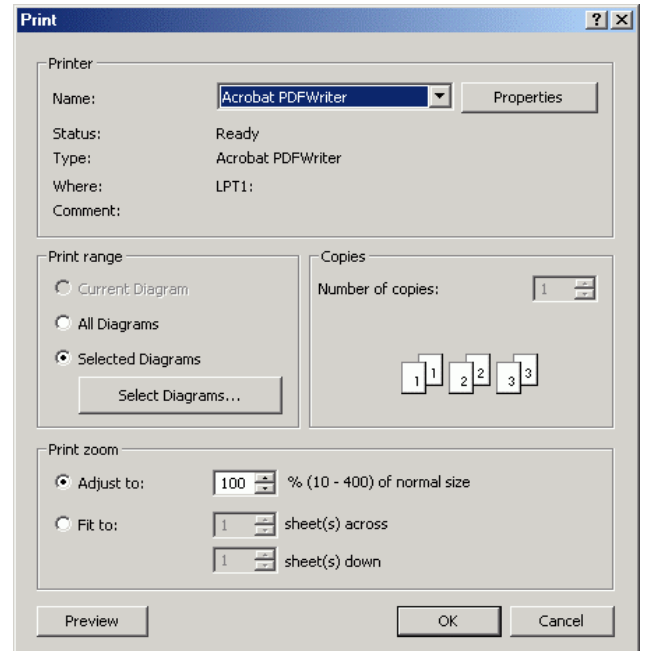


Figure 5: Rational XDE file export options

Note: The Acrobat PDF Writer is named Adobe PDF in Acrobat Version 6.

2. In the Save PDF File As window, name the file and pick a location where you want to save it (see Figure 6). The PDF file is created.

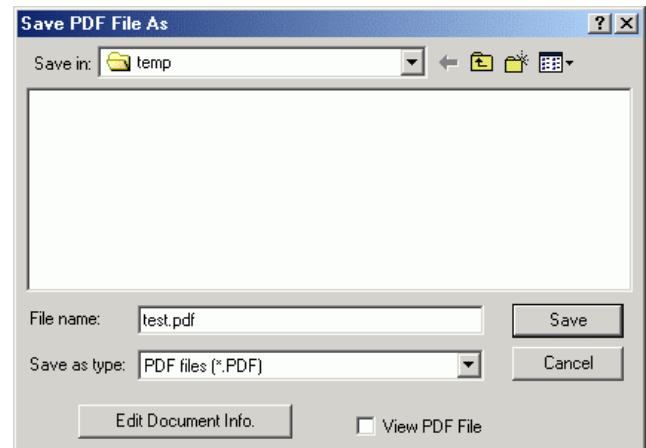


Figure 6: File location window

(In testing, the authors found that the PDF file saved directly in XDE was not completely compatible with graphic tools that Media Design uses, such as Adobe® Illustrator® or CorelDRAW®, because of the requirement to parse graphical elements from one application to another. The authors found that if writers have the full version of Adobe Acrobat®, they can continue following the instructions and produce a workable file format for the graphic designer. If they do not have Acrobat, they should forward the PDF file to

their graphic designer who will be able to produce a workable graphic file.)

3. Open the PDF file in Adobe Acrobat. Go to **File > Export > PostScript or EPS**. When the Export PostScript or EPS window appears, pick the following settings (see Figure 7):

File Format: EPS without Preview

PostScript Option: Language Level 1

Binary radio button

Font Inclusion: None

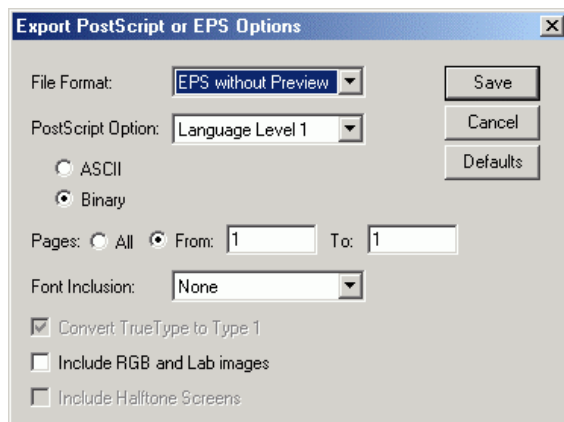


Figure 7: Rational XDE EPS file options

4. Click **Save** and name the file, or keep the current name and pick a location where you want to save it. Click **OK**, and you now have a workable EPS file.

2.3 Microsoft Visio

Because Microsoft Visio is a popular program, and based on user requests, the UML diagram workgroup updated the class diagram process to include a brief section on saving Visio diagrams to the appropriate file format. The following steps were added to the process:

1. After you have finalized the diagram layout, click **File > Save**. The Save window opens.
2. Select **.wmf** under Save As Type. Give the file a name and pick a location where you want to save it (see Figure 8). You have now created a workable WMF file.

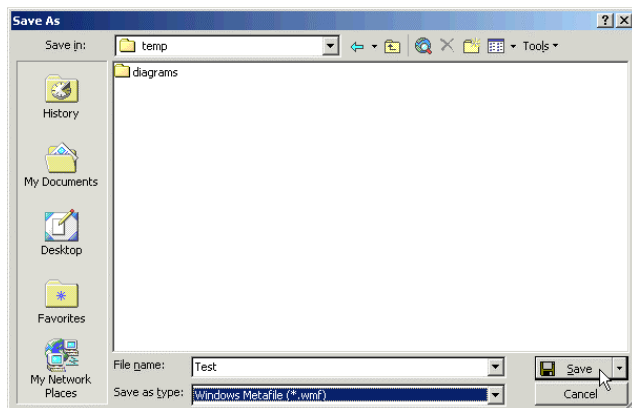


Figure 8: Microsoft Visio file export options

3. ADOPTION OF THE PROCESS AT THE IBM TORONTO LAB

The UML diagram workgroup was able to measure the success of the class diagram process three distinct ways:

- Anecdotal feedback
- Statistics derived from the MDS Diagram Request database
- Quality of submitted diagrams

3.1 Anecdotal feedback

The Web site launch was announced to the Toronto Information Development community by an e-mail notice. However, such a notice can sometimes meet with limited success (mass distribution e-mails are can easily be overlooked). Supplementing the e-mail notice, the workgroup members announced the new process and Web site to their respective teams, and received extremely positive reviews of the both the process and the timeliness of its creation. Teams were encouraged to communicate the existence of the Web site to any development teams that regularly require class diagrams in their documentation, and again the response was overwhelmingly positive.

3.2 MDS statistics

MDS was able to track information about diagrams by monitoring their Diagram Request database.

For all of 2003, and up to and including April, 2004, 135 UML diagrams were submitted. 77 of these 135 diagrams (57%) were submitted in a vector format (WMF, PDF, VSD [Visio]). All of these diagrams are easily editable when opened in a graphic application. In previous years, UML diagrams were in virtually all submitted in JPEG, GIF or BMP format, and needed to be recreated by a graphic designer.

Note: The Web site was officially launched in April, 2003. Thirty-nine UML diagrams were submitted in GIF format in the first three months of 2003, prior to the launch of the process Web site. Since the Web site was launched, 96 UML diagrams have been submitted, and of those, 77 have been submitted in a vector format, for a success rate of over 80%.

3.3 Quality of submitted diagrams

The best method to for measuring the success of the collaborative process for creating class diagrams is to examine the quality of the diagrams now being submitted to MDS. The following actual examples illustrate the type of improvements being realized.

Figure 9 shows an example of a diagram with a notation symbol included in place of unnecessary English text. This diagram will not need translated, saving time, effort, and costs.

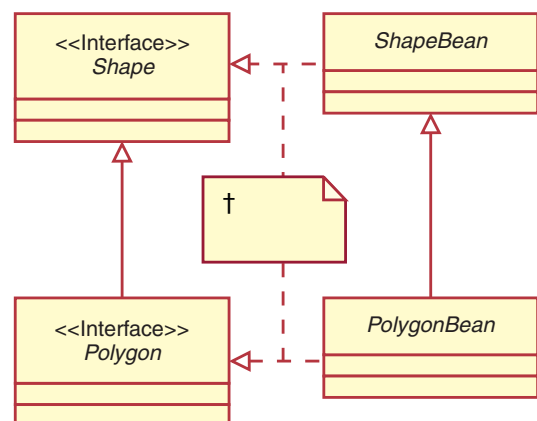


Figure 9: Figure with notation symbol

Figure 10 shows a Visio diagram submitted by a technical writer. The graphic designer grouped the elements in order to open the file and save to the WMF format. Although this is a large diagram, it conforms to the 540 pixel maximum established in the class diagram process, which means that users will not need to scroll when viewing the diagram in on-line help or in an IBM InfoCenter.

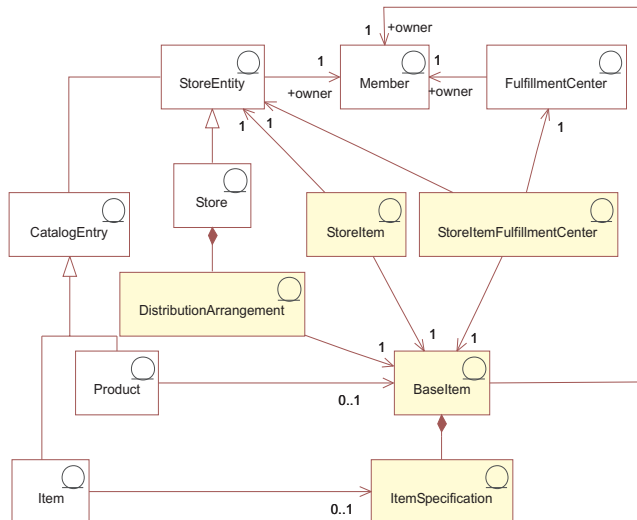


Figure 10: Diagram sized to 540 pixel maximum

Figure 11 shows an example of simplified content. The clean layout – rectangles aligned, right-angled connecting lines, distinct rectangle groups -- helps illustrate the concept, clearly showing the flow of information.

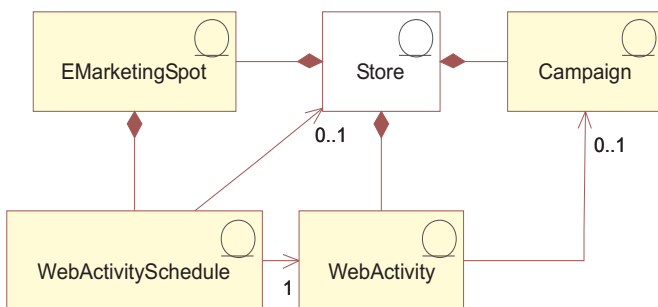


Figure 11: Diagram with simplified content

Figure 12 shows an example of a large diagram created according to visual design standards. Note the layout, alignment, organization, and connecting lines.

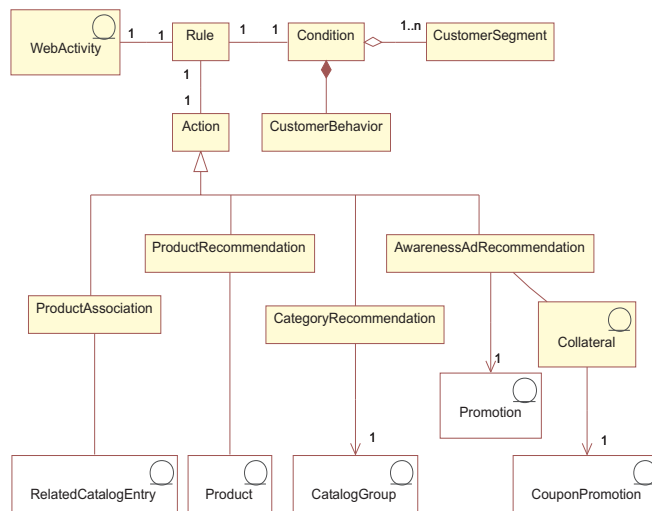


Figure 12: Diagram with appropriate visual design

Figure 13 shows a UML state diagram created using Visio. This diagram is included here because it shows that, even though the process does not currently contain any guidelines for state diagrams, writers and developers are applying the guidelines from the class diagram workflow to other types of UML diagrams. This is an unexpected but most welcome result. **Note:** There is a small parsing error in this diagram in that some text overlaps the oval elements. The authors have included the unedited diagram to give an accurate impression of the quality of diagrams submitted to the Media Design request database.

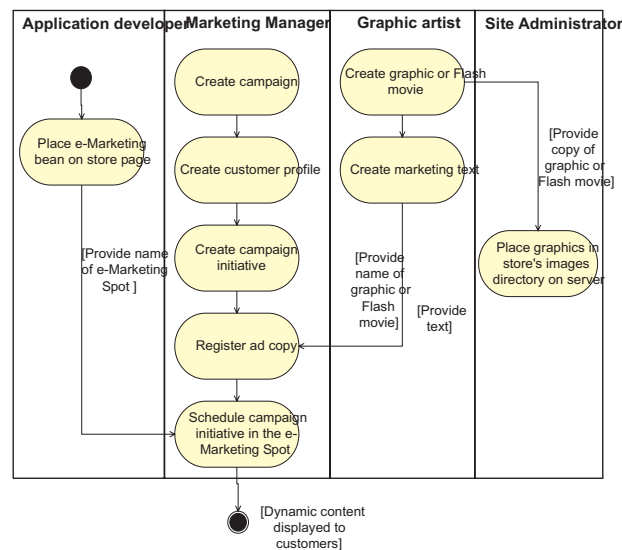


Figure 13: UML state diagram

4. CLARIFICATIONS AND UPDATES

Although the class diagram guidelines are comprehensive, providing step-by-step instructions for all three major stakeholders in creating UML diagrams for publication (developer, writer, and graphic designer), some areas of the process proved unclear or open to interpretation. Although, as noted, the process is now being used for most new diagrams,

some of the examples provided above show that some instructions are being followed more closely than others. Some stakeholders are misinterpreting – or ignoring – some key instructions. Notice that in the figures above, there are examples of misaligned boxes, diagonal lines, unequal weight given to sibling groups, and inconsistent box sizes.

To close this apparent gap in understanding, the workgroup created another example of a well-designed diagram layout (figures 14 and 15). This example includes a superimposed grid showing appropriate diagram alignment. This reinforcement of the grid design concept should assist stakeholders in understanding the applicable process steps.

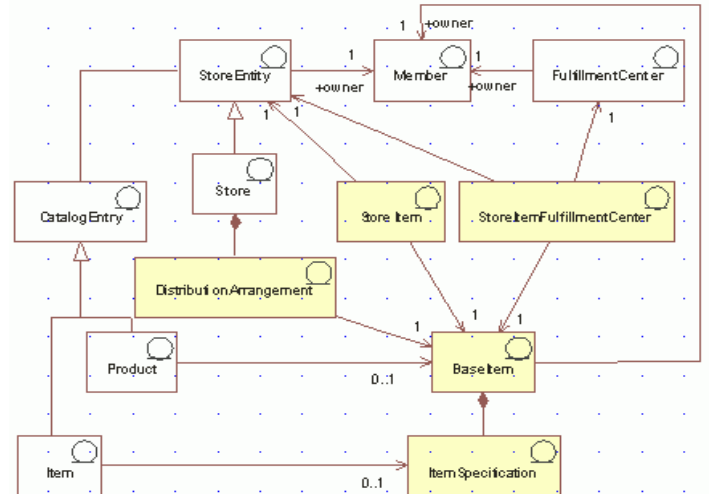


Figure 14: "Before" diagram

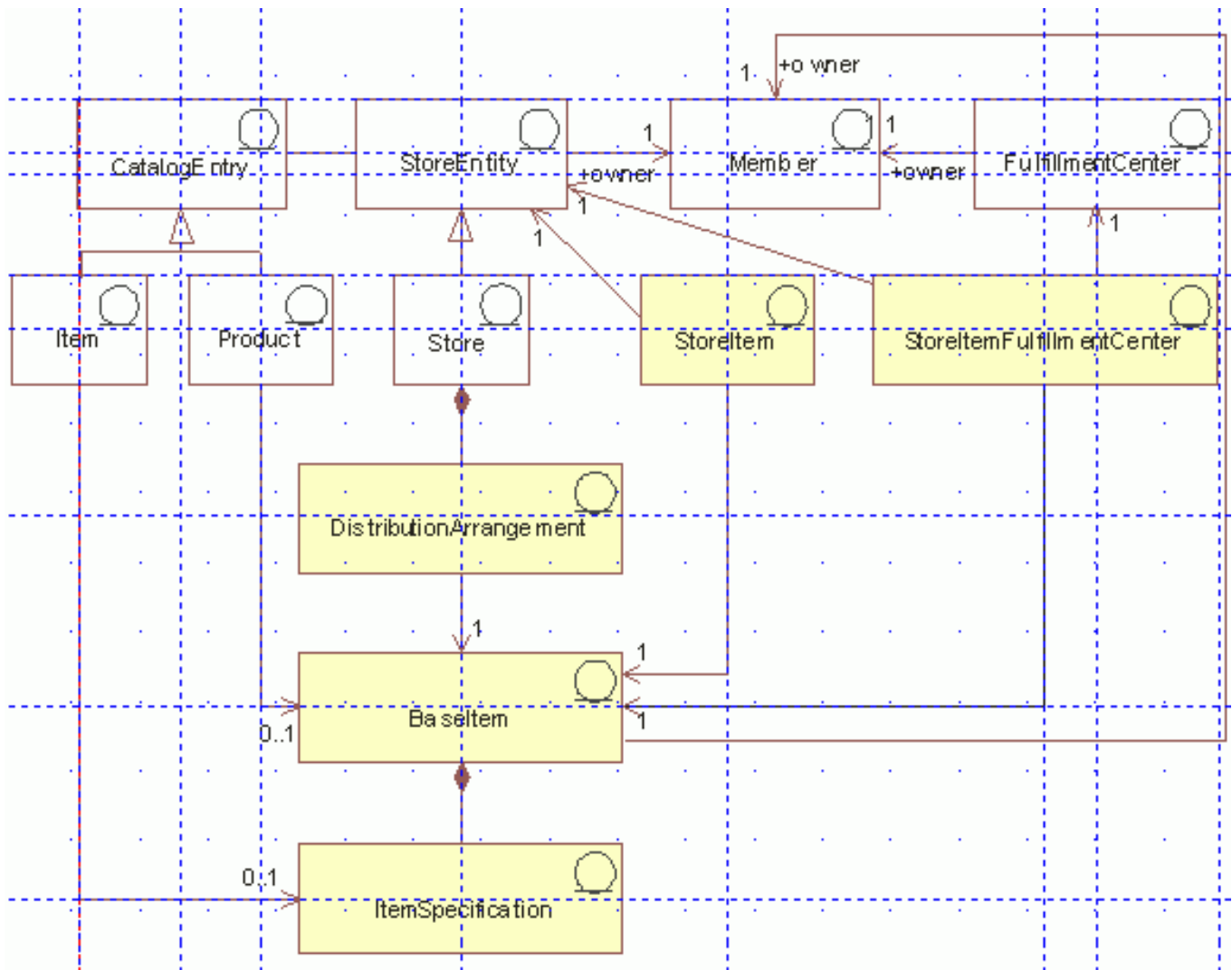


Figure 15: "After" example showing superimposed grid layout (enlarged).

5. CONCLUSION

The UML diagram workgroup continues to solicit feedback on the class diagram process. The effort to establish a collaborative process for creating class diagrams has been welcomed by the development and ID communities at the IBM Toronto Lab, and has yielded significant results in its first year of implementation. The workgroup is now seeking to apply the principles developed for the class diagram process to other types of diagrams in hopes of achieving further success. The positive results achieved by the class diagram process – shorter schedules, saved effort, and reduced costs – indicate the benefits of continued work in this area.

6. REFERENCES

- [1] MacKinnon, N. and Murphy, S. Designing UML Diagrams for Technical Documentation. *SIGDOC Conference Proceedings 2003*, p. 105.

7. TRADEMARKS

IBM, Rational, Rational Rose and WebSphere are registered trademarks of International Business Machines Corporation in the United States, other countries, or both..

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.