

Agile Modeling with Mind Map and UML

Kenji HIRANABE

Abstract

Requirement Gathering or Gathering User Stories (in Agile context) is always a challenging phase in software development. There are no standard processes or notations defined, In addition, communication and facilitation skills are the primary factor that makes the activity effective. In this paper, I propose a new method for exploring “User Wish” using Mind Maps and then model it using UML.

1 What is a Mind Map and Why?

A Mind Map is a unique graphical technique of taking notes as well as visualizing thoughts with a radiant structure [Buzan03].



Figure 1 XP Mind Map [Beck05]

Around a central image, radial branches (BOIs, basic ordering ideas) grow with key words along them, and associative sub-branches grow recursively. In Figure 1, “XP” is the central image and “Values”, “Principles” and “Practices” are the BOIs.

A Mind Map is logically a simple tree structure of key words, but has a radiant shape with colors, pictures, and drawings. Human brain is much better at recognizing shapes or patterns than words or numbers, and it does better than computers do. One example is when you meet an old high-school friend accidentally in a crowded station and recognize her in a second.

Roger Sperry¹ found that the cerebral cortex has two hemispheres, and the cortex tend to divide the major intellectual functions between them. The research showed that the right side appeared dominant in rhythm, spatial awareness, gestalt (wholeness), imagination, dreaming, color and dimension, whereas the left side appeared dominant in words, logic, numbers, sequence, linearity, analysis and lists.

By stimulating the right hemisphere (shapes, colors) as well as the left (logical key words), Mind

Maps have greater power to evoke human memories than ordinary linear notes do.

Here are the five properties of mind maps.

- Keyword Oriented – The structural elements of mind maps are not sentences but key words which construct a map of ideas as a whole.
- Loose Syntax and Semantics – Association is the only relationship between linked key words. It is looser than other graphical language such as UML.
- Fast and Easy – Because of the above two properties, you can write memo by mind maps very fast. Actually, it’s faster than natural language so you can use mind maps as shorthand of minutes, interviews and notes.
- High Level View – You can overview a whole mind map at a glance. Linear memo gets vertically longer down to pages as the content grows, but a mind map gets spatially dense. A mind map’s shape has self-similarity and uses 2D paper space more effectively to show the whole (gestalt).
- Evocative – You can remember the situation when you took the note later because of the right-cortex-oriented information included in the mind map. Mind Maps evoke the context of the scene.

2 Using Mind Maps in IT scenes

There are lots of situations where you can utilize mind maps effectively in your software development activities. The followings are my experiences.

- Minutes/Agenda of meetings – I often prepare an agenda for the meeting by a mind map containing time, place, attendants and agenda preset. During the meeting, I get my laptop PC connected to the overhead projector and I take notes of the conversation in real time. I call this a “meeting log” mind map. The preset mind map has open BOIs of “Conclusion” and “ToDo” so that the facilitator can bring up the final settlement to fill the space and not to leave the conclusion and action vague and undecided. Here’s an example of a minute template.

¹ A Nobel Prize winner in physiology, 1981

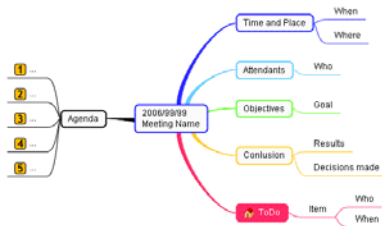


Figure 2 Meeting Agenda Template

- Presentation – In place of a PowerPoint presentation, you can use a mind map as a presentation visual. Because a mind map has explicit shape of logical structure and high level view of the whole presentation, the audience can keep up with your talk and not be lost in the discussion.
- Notes of seminars and books – You can write a book review and seminar memo by a mind map.
- Brainstorming – A team can brainstorm using mind maps. Here’s a picture of a mind map brainstorming, using Alistair’s Keep/Problem/Try reflection format (Figure 3).



Figure 3 Retrospective by Mind Map (Alistair’s Keep/Problem/Try)

- Conversation notes – Use a mind map to take notes of user interaction. It’s fast, evocative and flexible. In the following sections, I introduce a Mind Map as a user interaction memo to capture vague user wish.

3 User Stories and Mind Map

One of the core values of Agile is “interaction.” The key mind shift is from “writing (documentation)” to “talking (conversation).”[Cohn05] In XP, “Story Cards” are ticket/memento of user interaction. By restricting written information in a tangible small card, story cards naturally guides you shift writing to talking and promote conversation with the customer. Story cards are written by the customer by hand, so you can evoke the memory of the conversation by looking at the card. In that sense, story card is a memento of the conversation. At the same time, if you have a question about the story, you can remove the card from the wall and go talk with the customer with the card. So it also is a ticket to another conversation.

Mind maps also have strong evocativeness capturing the context of the conversation. The shape,

color and other properties help the participants of the conversation remember the situation. In that sense, a mind map can be a context embedded memo like story cards.

4 Exploring User wish by Mind Map

A user story is a small piece of user wish. I propose to use a mind maps in order to capture user wish as a whole. A mind map is as evocative as story cards, and better to see the whole at high level.



Figure 4 User Wish Interview Mind Map template

Figure 5 is a template mind map prepared for a user interaction to capture user wish. I prepare BOI’s in advance as questions to ask the users. The questions are:

1. Who will be happy by this system and why?
I always ask this question first. This will capture the system’s stakeholders, their values, their current problems, context and their expectations. These are the essential success factors and risk drivers behind requirements you cannot miss.
2. Who use the system?
This question captures users of the system.
3. When do they use the system?
This captures the system’s story or use case candidates.
4. What other systems communicate with the system?
This question captures other systems out side the system.
5. What information do you want to manage by the system?
This question can gather a good collection of “entities”, objects of domain models. The users concerns and domain keywords can be found in this BOI.
6. Homework
This is not a question. This is a place holder or a parking lot for items you can find answers in the session.

I used this format to capture a city library system. Appendix is the result of my one-hour interview session with a real librarian. This mind map is created concurrently by me in the interview. I got my lap top connected to an overhead projector showing the questions and answers in the mind map, editing it in real time talking with her.

Note that there are items in the “homework” BOI, which includes actions to the question without a good answer in the session and the schedule of the

next meeting. Mind maps have a very flexible format to accommodate these accidental topics.

5 Converting Mind Map to UML

After exploring the whole picture of what the user wants, there are two ways to proceed. One is to gather user stories from the user as in usual agile planning games. The other is to create an agile model of use case models and domain entity models. In this paper, I'll show the latter path.

On the user wish mind map, I add icons of actors, use cases and classes to their candidate. The prepared BOI questions are designed to capture them. In Appendix you can find a UML use case diagram and a class diagram generated from the mind map.

Using mind maps in conjunction with UML means dividing requirements exploring thinking into two modes. The first mode is "Requirements Gathering (User Wish Exploration)" with mind maps. It captures and shares vague ideas and keywords at high level view quickly. The second mode is "Requirements Modeling" with UML. In this mode, you model domain objects and use cases from the keywords gathered in the first mode. I found that "gathering mode" is a divergent thinking, whereas "modeling mode" is a convergent thinking.

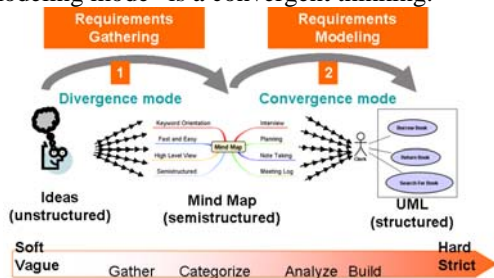


Figure 5 Requirements by Mind Map and UML

6 Tool: "JUDE" – Mind Map and UML editor

We have been developing a UML editor called "JUDE". Recently we combined a new Mind Map feature with the tool so you can generate UML diagrams via Mind Maps.

All the diagrams in the Appendix are drawn by the tool.

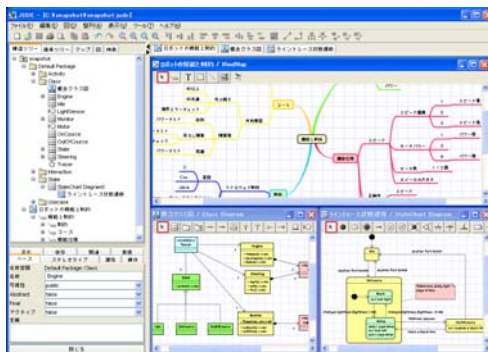


Figure 6 JUDE – UML and Mind Map tool²

7 About the Author

Kenji HIRANABE is the CEO of Change Vision, Inc. He is one of the creators of JUDE, and has translated the books "Lean Software Development", "XP Installed", "Agile Project Management", and other XP/Agile books into Japanese. Kenji thinks of software development as a form of communication game, and is always searching for better ways that makes it more productive, collaborative, and fun.

8 References

[Buzan03] Tony Buzan, "The Mind Map Book", 2003 BBC Active



[Beck05] Kent Beck, "Extreme Programming Explained 2nd ", 2005 Addison-Wesley

[Larman03] Craig Larman, "Agile and Iterative Development", 2003 Prentice Hall

[Cockburn01] Alistair Cockburn, "Agile Software Development", 2001 Addison-Wesley

[Ambler02] Scott Ambler, "Agile Modeling", 2002 John Wiley & Sons

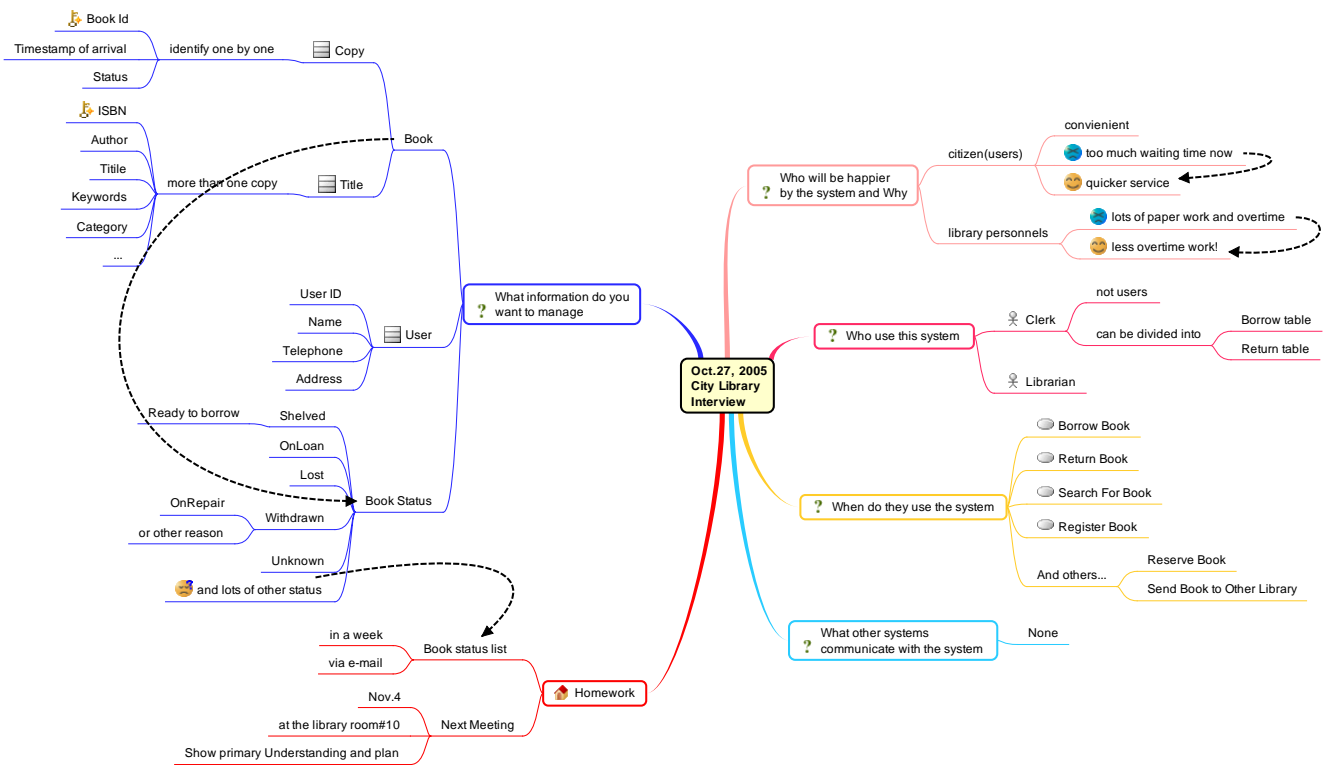
[Cohn04] Mike Cohn, "User Stories Applied", 2004 Addison-Wesley

² <http://jude.change-vision.com/> for more information

Appendix

Mind Map and UML of City Library System

Mind Map of User Interaction



Use Case Diagram Class Diagram

