CHAPTER 1: THE PROBLEM

1.1 Rationale

A recommender system is a system that generates suggestions of items according to user needs. Recommender System is essential in filtering unnecessary information during the recommendation process so that decision-making becomes faster and more accurate [1][2]. Knowledge-based recommender system is one form of recommender system. This approach helps through provide recommendations based on product functional requirements requested [1]. The advantage of a knowledge-based recommender system are as follows: 1) it can provide recommendation results are not based on the preferences of other users, but through a question-and-answer dialog (noise-free recommendations), and 2) recommendations that are more reliable [3].

One of the results of development in the knowledge-based recommender system is the conversational recommender system(CRS). CRS supports a consistent dialogue between the user and the system so that the recommended product suits the user's needs [4]. In the development of CRS there are several challenges in its development are as follows: 1) CRS must be able to interact efficiently, and 2) users can express user requirement easily [5].

Sadeghian, et al [4] developed CRS based on the product's functional requirements. Product functional requirements are products that are seen from their usability. For example, someone needs a smartphone to play games. This CRS was developed to make it easier for users who do not understand the technical features (novice users) to choose products that suit their needs.

In this study, we use ontology as the basis of knowledge. There are three hierarchies used in the knowledge base: a. the functional requirements hierarchy (Funcreq), b. products hierarchy, and c. specifications hierarchy [6].

The Funcreq hierarchy represents the functional requirements used for the question-generating process. During the generating question process, we use semantic reasoning to find candidate nodes on the ontology, the semantic reasoning process will go through the classes that match the user input, each search result will get candidate nodes. Candidate nodes are set functional requirement that the user potentially likes [1]. However, when the system randomly selects

functional requirements from candidate nodes, the questions asked may not match the user's preferences. Thus, the user and system interaction becomes inefficient because the system will ask questions repeatedly. We overcome this problem by involving a learning mechanism to give weight to each candidate node. Therefore, some candidate nodes with the highest weight will go first to the user.

In this study, we use Singular Value Decomposition (SVD) to predict the functional requirements of users who have interacted with the system by using data from user interaction history. We also tested several machine learning algorithms and found SVD as the most stable algorithm for predicting functional requirements. Then, the output of SVD algorithm is a prediction matrix.

Product functional requirements are products that are seen from their usability. For example, someone needs a smartphone that can be used to play games [1][2]. This functional requirements-based CRS product can help users who do not understand technical features to express their needs [1]. CRS based on product functional requirements is effective for products that have many features and are multi-functional.

Generating questions is a process in CRS whose job is to generate questions, when the system interacts between users, the system will ask questions, these questions are the result of generating questions, this process involves a process of semantic reasoning to go through the hierarchy of Functional Requirements (Funcreq) [1]. In the ontology, the results of the semantic reasoning process are candidate nodes, four of the candidate nodes are chosen randomly, the results of this selection will be asked by the user [1]. We develop CRS based on functional product requirements. The process of generating questions in this CRS uses semantic reasoning on ontology [3]. develop CRS using Conversational Path Reasoning. Generating questions in this CRS uses semantic reasoning and ontology as base knowledge.

1.2 Theoretical Framework

Collaborative filtering is an approach in recommender systems. This approach will recommend items, the items are selected based on the prediction results of the rating vs user matrix. The items that have the highest weight will be recommended to users [4]. Many algorithms are used to predict this rating, such as Singular Value Decomposition (SVD), K-nearest neighbors (KNN), Non-negative matrix factorization (NMF), etc.

1.3 Conceptual Framework/Paradigm

We built CRS to help users who are not familiar with technical features to get products (smartphones) that suit user needs. Involving a learning mechanism in selecting candidate nodes can make the interaction more efficient, and SVD method can weight candidate nodes with good accuracy and running time.

1.4 Problem Statement

Several studies have developed CRS based on functional needs [5][1][6][7]. In these studies, the system selects candidate nodes using semantic reasoning on the ontology (Funcreq) of candidate nodes that are randomly selected to be asked to the user. However, the questions asked by the system may not match the user's preferences. This will cause the user's question and answer interaction with the system to be repeated and make the interaction inefficient.

1.5 Hypothesis

An efficient CRS is CRS that has few iterations, fast system response and can recommend products according to user preferences [1][5][6][7][8], so that a good generating question process is needed. A good generating question process is the questions that are generated according to the user's preferences.

Premise 1 : Good question-generation can determine efficient CRS [1][8][5][6][7]

Premise 2 : Learning methods in collaborative filtering can determine the weights in the candidate nodes [4].

Hypothesis :

1. Involving a learning mechanism in selecting candidate nodes can make the interaction more efficient.
2. The SVD method can weight candidate nodes with good accuracy and running time.

1.6 Assumption

CRS based on functional requirements combined with semantic reasoning and SVD for the question-generating process can increase the efficiency of the number of iterations when the user interacts with the system.

1.7 Scope and limitation

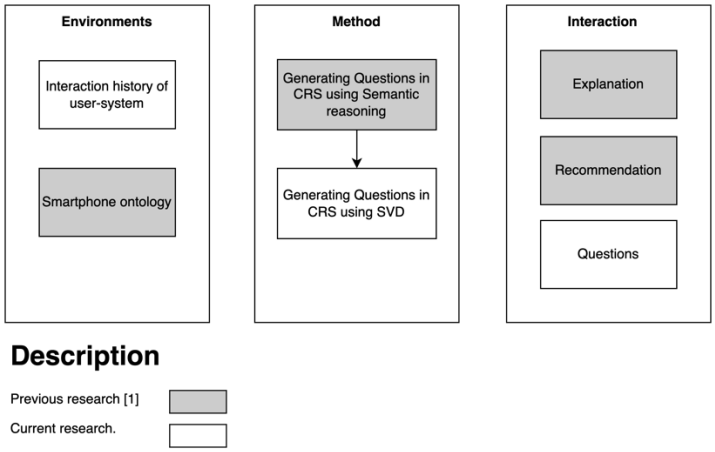

Figure 1.1 Research Scope

In this study, we generate questions on CRS based on product functional requirements. We use user-system interaction history data, and smartphone ontology (previous research) as input. To generate this question, we combine two methods, that are using semantic reasoning (previous research) which is used to obtain candidate nodes, and after obtaining candidate nodes, we weight them using SVD. The result of this generating process is a question that is likely to match the user's preferences. Where the question is one of the interactions in CRS. In addition to the question, there are interactions in CRS, that are recommendation, and explanation (an explanation of why the product is recommended to users), but both interactions have been carried out in previous research.

a. We use the functional requirements ontology from previous research.
b. We do not include new users or users who have never interacted
c. We are only limited to users who have interacted with users.
d. This research focuses on the process of generating questions.
e. Do not make changes to the domain knowledge on the ontology.

f. Ontology and semantic reasoning process using previous research.
g. We do not make changes to the recommended product process.

1.8 Importance of Study

Building CRS based on the functional requirements of the product to help users who do not understand the technical features or difficulties in disclosing technical features to get products that match user preferences, by using the SVD method and semantic reasoning to generating questions, it is hoped that CRS will be more efficient in interacting users with the system.