

Identifying Transparency Requirements by using Smart Knowledge

Background

The discussion about the need for software transparency is growing, motivated by the increasing number of software systems making part of our lives. Built-in transparency can help to build trust and influence the relationship of the user with the software. It can also facilitate its use and provide the user with a better understanding of the situation he/she is in and why he/she has obtained that output or result, favoring a better User Experience.

Identifying those requirements are an important step to build understandable and transparent software to the end-user. However, this identification is also a challenge for software engineers since transparency requirements are a complex network of interconnected requirements which influence one another: e.g., explainability, usability, informativeness, understandability, accessibility. It is also important to keep the users' expectations and opinions in mind in order to provide quality software.

One way of identifying those needs is through the analysis of textual documents, including user feedback and specification documents.

Aufgabe

In the context of this thesis, an analysis should be carried out to identify concepts related to transparency. This includes the research of related terms and their representation in a data model (e.g., ontology, or others). A second part of this work consists in identifying these requirements in text by using this data model. A software component should be developed, which facilitates the analysis of text documents (e.g., comments in App Stores, specifications documents) and the identification and classification of those requirements. The idea is to facilitate the work of software engineers, providing better information visualization and a smart identification of transparency requirements.

The resulting application should be evaluated by means of an empirical study.

Includes:

- Limiting the points and the scope to be investigated in cooperation with the supervisor
- Investigate the related concepts
- Design study and architecture
- Development of software component
- Evaluation
- Interpretation and discussion of the results

Organisatorisches

Betreuer: M.Sc. Larissa Chazette larissa.chazette@inf.uni-hannover.de, Raum G304

M.Sc. Wasja Brunotte, wasja.brunotte@inf.uni-hannover.de, Raum G307

Prüfer: Prof. Dr. Kurt Schneider,

Beginn: ab sofort möglich

Randbedingungen: Good Programming Skills, Usability, UX