

Kontextbasiertes E-Learning zur Diagnose und Behebung von Netzwerkfehlern am Beispiel der App ThermoFind

Hendrik Geßner¹ und Kristin Sass¹

Abstract: ThermoFind ist eine App zur Fehlerdiagnose und –behebung in Thermostat-Steuerungs-Netzwerken. Dabei wurde ein kontextbasierter Ansatz gewählt, um passend zu den diagnostizierten Fehlern Lerneinheiten zu ermitteln, die beim Verständnis der Fehler und deren Behebung unterstützen. Dabei wird auf E-Learning gesetzt, um eine Vereinigung von beruflicher Erfahrung und der App bekannten Lerninhalten zu erreichen.

Keywords: context based elearning, network diagnosis, contactJS, micro learning

1 Szenario und Konzeption

Die App ThermoFind ist Rahmen einer Kooperation der Universität Potsdam mit der Firma equeo entstanden. Das Szenario wurde hierbei von equeo entwickelt. Ziel war es, Wartungstechniker bei der Vor-Ort-Fehlerdiagnose und –behebung an Thermostatsteuerungseinheiten und den dabei beteiligten Netzwerken und Netzwerkgeräten zu unterstützen. Der Bedarf hierzu ergab sich aus einer Vielzahl an heterogenen Fehlerquellen, deren schrittweise manuelle Überprüfung zeit- und arbeitsintensiv ist.

ThermoFind wurde zur Unterstützung von Wartungstechnikern konzipiert. Dabei werden je nach Situation passende Lerneinheiten angezeigt, die jeweils auf einen diagnostizierten Fehler hinweisen sowie Lösungsansätze aufzeigen und on demand benötigtes Zusatzwissen vermitteln. Ziel war es, einen Technologie- und Marketingprototypen zu entwickeln. Es ging einerseits um die Überprüfung eines kontextorientierten Ansatzes auf Basis des contactJS-Tool-Stack [MLL15]. Andererseits spielte die Vermarktbarkeit durch equeo eine Rolle.

Der Prototyp konzentriert sich auf eine falsch konfigurierte Zeit an der Thermostatsteuerungseinheit, betrachtet aber auch angrenzende Fehler, insbesondere in der Netzwerkkonfiguration.

¹ Universität Potsdam, Institut für Informatik und Computational Science, August-Bebel-Straße 89, 14482 Potsdam, vorname.nachname@uni-potsdam.de

2 Implementation

ThermoFind wurde als Android-App entwickelt. Dabei kam das Hybrid-Framework Cordova zum Einsatz. Der Quellcode der App ist dabei überwiegend in HTML, CSS und JavaScript geschrieben. Für WiFi- und NTP-Zugriff wurde nativer Android-Java-Code verwendet.

Die Kontexterfassung wurde mit Hilfe von contactJS, einem JavaScript-Port des Context Toolkit [Ak01], umgesetzt. Die App überprüft kontinuierlich alle Kontextquellen. Die ermittelten Kontextrohdaten werden verfeinert und mit einem Regelwerk abgeglichen. Auf Basis dieses Abgleiches wird eine Diagnose gestellt und passende Lerneinheiten ausgewählt. Die so ermittelten Einheiten werden in einer Liste dargestellt. Lerneinheiten enthalten dabei immer eine Übersicht des Netzwerkes, wobei der von der Einheit betrachtete Fehlerort hervorgehoben ist. Unter der Netzwerkübersicht erklärt ein Hilfetext den diagnostizierten Fehler und gibt Lösungsansätze. Optional werden weitere Inhalte, beispielsweise ein Auszug aus dem Gerätehandbuch, eingeblendet.

3 E-Learning-Szenario

Das Ziel von ThermoFind war es, Wartungstechniker bei der Fehlerdiagnose und -behebung zu unterstützen. Somit wird erwartet, dass auf ein technisches Grundverständnis zurückgegriffen werden kann. Es wurde bewusst auf eine Kette von Anweisungen zur Fehlerbehebung verzichtet, die App-Benutzende zu bloßen Marionetten einer Diagnose-Anwendung machen würden. Stattdessen sollte eine Vereinigung von beruflicher Erfahrung und der App bekannten Lerninhalten mit Hilfe von Microlearning erreicht werden. So wurden die Textinhalte der App nicht als Anweisungen, sondern als Vorschläge formuliert. Zudem gibt es abgesehen von Erfolgsmeldungen keine sequenziellen Abhängigkeiten zwischen Lerneinheiten. Techniker müssen also nicht erst einen Diagnoseschritt durchführen, um den nächsten freizuschalten. Jeder Inhalt ist auf etwa einer Displaygröße Umfang reduziert, um eine fokussierte Bearbeitung zu ermöglichen.

4 Evaluation und Fazit

ThermoFind wurde sowohl in Nutzertests als auch über Expertenfeedback evaluiert. In den Nutzertests hat sich gezeigt, dass technisches Vorwissen in Umgang mit Netzwerken erforderlich ist, um die Meldungen der App verstehen und geeignete Maßnahmen ergreifen zu können. Mit Wissen aus einem Informatik-Studium konnten alle Fehler erfolgreich behoben werden. Das Expertenfeedback hat ergeben, dass vorbereitende Inhalte fehlen, die vor dem Eintreffen bei Kunden konsumiert werden können und beispielsweise über erwartete Geräte und Netzwerktopologie aufklären. Sowohl Nutzertests als auch Expertenfeedback haben auf die Notwendigkeit von

Feedback hingewiesen. So muss jederzeit erkennbar sein, ob noch Diagnoseprozesse laufen oder keine Hilfe angeboten werden kann.

Die Prozesse in Thermostat-Steuerungs-Netzwerken sind ausreichend komplex, um technische Unterstützung bei der Diagnose und Wartung zu erfordern. contactJS hat sich hierbei als brauchbarer Technologie-Stack erwiesen. Insbesondere sind Erstellung und Wartung der E-Learning-Inhalte mit Hilfe des bereitgestellten Autorensystems gut handhabbar und auch ohne Programmierkenntnisse möglich. Allerdings ist die Implementierung der Kontexterfassung aufwendig, da die Fehlerdiagnose einsetzspezifisch entwickelt werden muss. Insgesamt sind die zugehörigen Werkzeuge noch in einem Alpha-Stadium und benötigen Weiterentwicklung.

5 Inhalt der Demo

Die Demonstration umfasst eine Vorführung der App in einem Demo-Netzwerk mit Thermostatsteuerungseinheit. Der Aufbau des Netzwerkes ist hierbei in Abb. 1 skizziert.

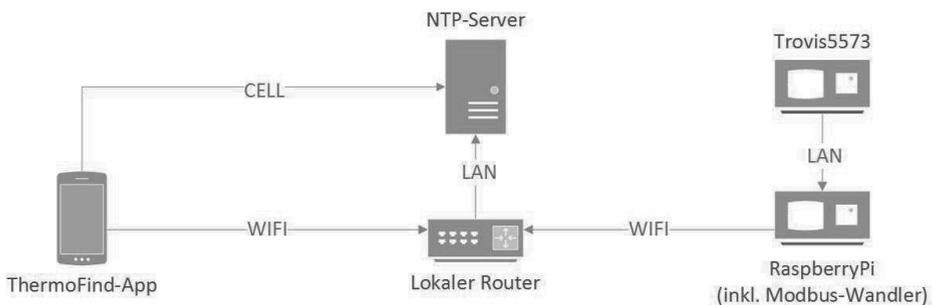


Abb. 1: Demo-Netzwerkaufbau

Innerhalb des Netzwerkes werden verschiedene Probleme provoziert, die anschließend unter Zuhilfenahme der App schrittweise diagnostiziert und anschließend behoben werden. Die Fehlerszenarien orientieren sich hierbei an den Szenarien der Nutzertests. Es ist vorgesehen, Interessenten sich selbst an der Fehlerdiagnose und -behebung versuchen zu lassen.

Literaturverzeichnis

- [Ak01] Dey, A. K.: Understanding and Using Context. Personal and Ubiquitous Computing Journal 5/01, S. 4-7, 2001.
- [MLL15] Moebert, T.; Lemcke, S.; Lucke, U.: A cross-platform context detection framework. Proc. 15th International Conference on Advanced Learning Technologies (ICALT), 2015.