Richtlinie zum „Software-Sprint“

Alarmdisplay - Einsatzdaten zur Unterstützung von Hilfsorganisationen

Schlussbericht

Zuwendungsempfänger:

Andreas Brain

DLR Projektträger Berlin
Eingegangen am:
01. Okt. 2020
Eingangsnr.: 4985/B9

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S30 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Das Projekt Alarmdisplay soll Hilfsorganisationen wie z. B. der Feuerwehr als sogenanntes Informationssystem dienen. Dabei werden eingehende Alarmmeldungen verarbeitet und in einer gut zu erfassenden Art und Weise dargestellt (z. B. auf einem Bildschirm in der Fahrzeughalle). Dieser Bedarf entstand aus der persönlichen Erfahrung des Autors bei der Freiwilligen Feuerwehr. Zwar gibt es bereits solche Systeme auf dem Markt, jedoch sind diese in der Regel mit laufenden Lizenzkosten verbunden. Ebenso kann die Beschaffung eines PCs mit Microsoft Windows erforderlich werden, der durchgehend eingeschaltet bleiben muss. Diese Höhe von initialen und laufenden Kosten steht für kleine Feuerwehren mit unregelmäßigem Einsatzaufkommen oft nicht im Verhältnis zum erwarteten Nutzen. Das Open-Source-Projekt AlarmWorkflow[1] setzt ebenfalls auf Windows auf und eine Weiterentwicklung war in den letzten Jahren kaum vorhanden.

Das Ziel des Projektes war es, ein frei verfügbares und betriebssystemunabhängiges Informationssystem zu schaffen, das auf Kleinstrechnern wie dem Raspberry Pi lauffähig ist. Damit sollten sowohl die Beschaffungskosten als auch die laufenden Kosten gesenkt werden. Zusätzlich sollte das System über Schnittstellen verfügen, über die es mit bestehender oder noch zu entwickelnder Software verknüpft werden kann.

Das Projekt wurde von Anfang an in zwei Komponenten unterteilt. Eine Komponente zur Erfassung, Verarbeitung und Weiterleitung von Alarmmeldungen (Alarmzentrale), sowie eine Komponente zur Darstellung der Daten zu einem Einsatz (Alarmanzeige). Die beiden Komponenten sollten auf einander abgestimmt sein, sich aber nicht gegenseitig bedingen. Somit sollte es auch möglich sein, nur eine der beiden Komponenten zu verwenden und mit eigener Software zu kombinieren.

Folgende Meilensteine wurden definiert:
- Aufsetzen der Projektstruktur und Build-Automatisierung
- Verwaltungsoberfläche für Displays und Setup-Prozess
- Ruhemodus mit Datum/Uhrzeit, Terminen und Wetterwarnungen
- Texterkennung (OCR) und Validierung beim Faxempfang
- Extraktion einsatzrelevanter Daten aus Texten nach bekanntem Muster (Fax der ILS Augsburg)
- Alarmanzeige mit den wichtigsten Daten
- Validierung von Adressen und Koordinaten
- Karten für die Alarmanzeige mit Zusatzdaten von OpenStreetMap (z. B. Hydranten)

[1] https://github.com/OpenFireSource/AlarmWorkflow

- Extraktion einsatzrelevanter Daten aus Texten nach einem frei definierbaren Muster
- APIs zur Datenübermittlung inkl. Authentifizierung
- Verwaltungsoberfläche zur Verknüpfung von Dateneingang, Verarbeitungsschritten und Datenausgabe
- Ausdruck von Einsatzdaten und Karten

Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

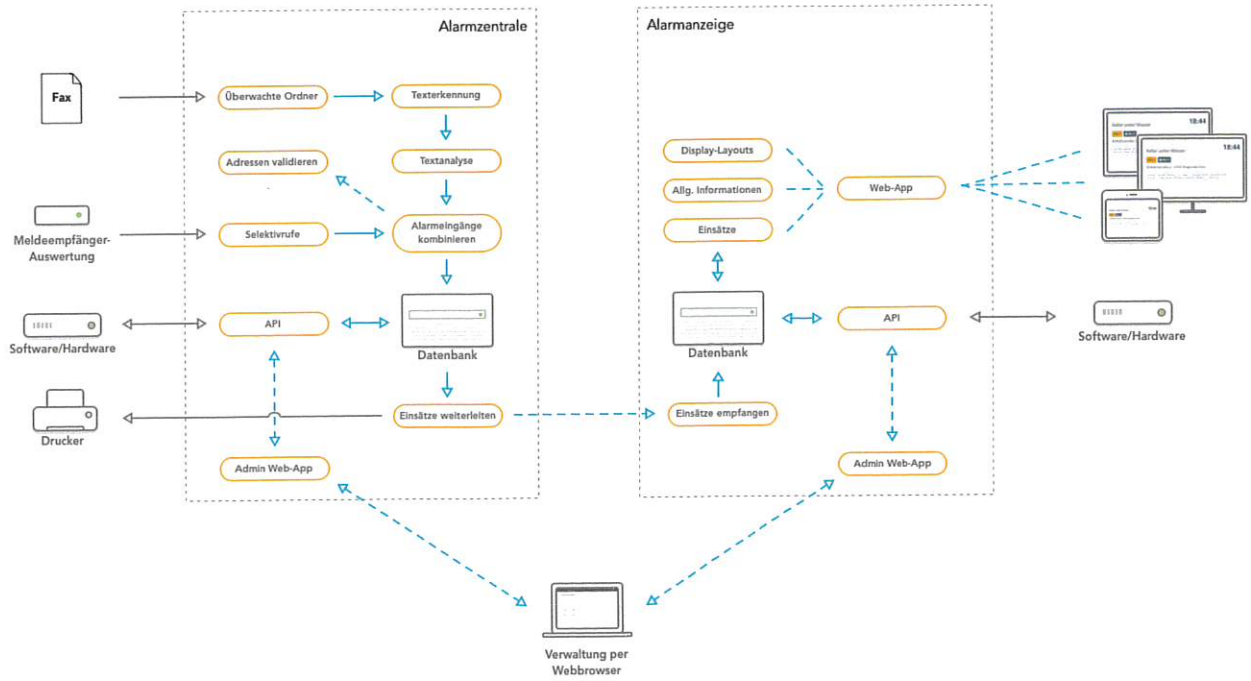
Neben den genannten kleinen Freiwilligen Feuerwehren sind grundsätzlich alle Hilfsorganisationen, die spontan Einsatzkräfte alarmieren müssen, Teil der Zielgruppe. Selbst große Organisationen, bei denen die Kosten für ein proprietäres System keine Rolle spielen würden, profitieren von mehr Transparenz und Interoperabilität. Anpassungen oder Integrationen mit bestehenden Systemen sind keine Frage von Lizenzen oder Verträgen, sondern die Nutzenden werden ermächtigt, diese selbst vorzunehmen oder zu beauftragen. Dies stärkt die Flexibilität und Schlagkraft der Hilfsorganisationen, was der Gesellschaft als Ganzes zugute kommt.

Ausführliche Darstellung der Ergebnisse

Es wurden wie geplant die zwei Komponenten Alarmzentrale und Alarmanzeige geschaffen. Beides sind eigenständige Projekte, die miteinander kompatibel sind und folgen intern dem gleichen Aufbau. Den Kern bildet jeweils ein Node.js-Server, der an eine relationale Datenbank angeschlossen ist. Sämtliche Entitäten der Software sind über eine REST-API per HTTP oder über eine WebSocket-Verbindung verwaltbar.

Ebenso liefert der Server eine grafische Oberfläche zur Verwaltung („Console“) als Webapp aus. Im Fall der Alarmanzeige wird noch eine zusätzliche Webapp für die Anzeige auf den Bildschirmen ausgeliefert. Somit ist jegliche grafische Interaktion über ein beliebiges Gerät mit einem Webbrowser durchführbar. Die Webapps wurden auf Basis von Vue.js geschrieben und sind per WebSocket mit dem Backend verbunden. Das ermöglicht nicht nur einen sehr schnellen Datenaustausch, sondern auch eine laufende Kontrolle ob angeschlossene Systeme nach wie vor verbunden und betriebsbereit sind.

Der Zugang zur API wird über JSON Web Token (JWT) abgesichert, die an einen Benutzeraccount geknüpft sind und eine begrenzte Gültigkeit haben. Um auch Lösungen zu ermöglichen, in denen beispielsweise Mikrocontroller eingesetzt werden, können ebenso dauerhaft gültige API-Tokens generiert werden, die unabhängig von einzelnen Benutzern sind. Derzeit gibt es weder bei den Benutzerkonten noch bei den API-Tokens ein Konzept zur Beschränkung von Berechtigungen, soll aber noch eingearbeitet werden.

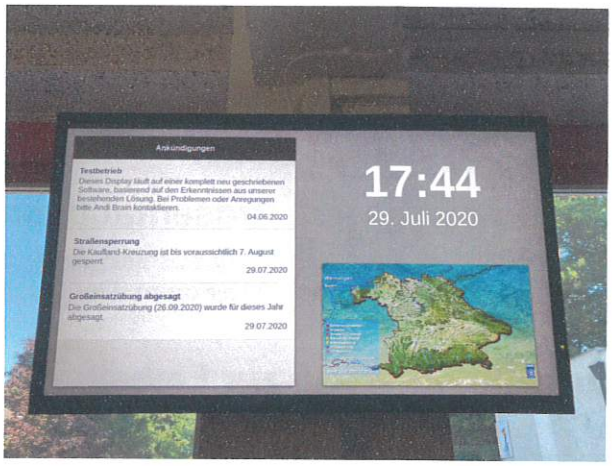


Die Alarmzentrale kann Ordner auf neu eintreffende PDF-Dateien (Alarmfaxe) überwachen und diese auslesen. Dazu kommt die OCR-Software tesseract[2] zum Einsatz, deren Ergebnisse dann nach relevanten Daten zu Einsätzen durchsucht werden. Nachdem jede Leitstelle ein anderes Layout für das Alarmfax verwendet, kann bisher nur das Fax der ILS Augsburg zuverlässig ausgewertet werden. Die Layoutdefinition zur Textanalyse wurde bereits in ein neutrales Format mit regulären Ausdrücken gegossen, sodass die Erweiterung um weitere Layouts keinen Eingriff in den Code erforderlich macht.

Auch von außen können Alarme per API zugeliefert werden, beispielsweise für per Funk empfangene Alarmierungen durch Fünftonfolgen. Mehrere eingehende Alarme werden zu Einsätzen kombiniert, konfigurierte Einsatzmittel werden erkannt und den Einsätzen zugeordnet. Enthaltene Adressen können optional mit Daten von OpenStreetMap abgeglichen werden. Die Einsätze werden an die Alarmanzeige weitergeleitet, weitere Schnittstellen (z. B. für die Zusatzalarmierung) sind geplant.

Die Alarmanzeige stellt den eingegangenen Einsatz möglichst knapp und übersichtlich dar: Einsatzgrund, Stichwort, Zeit seit dem Alarmeingang, Zieladresse und einen Freitext der Leitstelle. Wenn gerade kein Einsatz vorliegt, dient die Anzeige als Informationsübersicht. Bisher sind Datum und Uhrzeit, Unwetterkarten des Deutschen Wetterdienstes und Ankündigungen umgesetzt. Die Anordnung der einzelnen Informationen kann mit einem Drag-and-Drop-Editor in der Console verändert werden. Jedes verbundene Display ist individuell konfigurierbar, auch mehrere wechselnde Ansichten pro Display sind möglich.





Die Darstellung von Terminen wurde zugunsten der universeller einsetzbaren Ankündigungen in die Zukunft verschoben. Im Zeitrahmen der Förderung konnte die Kartendarstellung und der Ausdruck von Einsatzdaten nicht mehr umgesetzt werden, ist aber weiterhin vorgesehen. Eine grafische Oberfläche zur Verknüpfung von Dateneingang, Verarbeitungsschritten und Datenausgabe war vorerst nicht nötig, da die Optionen noch nicht so vielfältig sind.

Neben der ständigen Betreuung durch die Open Knowledge Foundation war auch das UX-Coaching durch *Simply Secure* eine große Hilfe, den Fokus auf die wichtigen Dinge zu richten.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Neben den unter „Beitrag des Projektes zu den Zielen der Förderinitiative“ genannten Vorteilen für die Zielgruppe und die Gesellschaft, ergeben sich alleine aus der Veröffentlichung des Quellcodes langfristig positive Effekte. So besteht eine Basisvariante eines Informationssystems, das Interessierten einen Einstieg bietet und für einige auch dauerhaft ausreichend sein wird. Komplexere Anforderungen können über die Zeit in Kooperation mit den betroffenen Organisationen umgesetzt werden. An FOSS-Lösungen interessierte Städte und Verbände könnten das Projekt finanziell

---

[2] https://tesseract-ocr.github.io/

oder mit Beiträgen zu Code und Dokumentation stärken und so den Einsatz von quelloffener Software in der Daseinsvorsorge vorantreiben.

Die Arbeit an diesem Projekt war eine bereichernde Erfahrung, da alle Aspekte von der Planung und dem UX-Design, über Code und Tests bis hin zur Dokumentation abgedeckt werden mussten. Dennoch ist die Verteilung der Aufgaben auf mehrere Schultern für die Zukunft wünschenswert.

Zunächst sollen weitere Alarmquellen angebunden und ebenso die Weiterleitung von Einsätzen an andere Lösungen ermöglicht oder vereinfacht werden. Ebenso sollen aktuelle Wetterdaten des DWD über das ebenfalls in der 7. Runde geförderte Projekt *Bright Sky*[3] eingebunden werden. Zusammen mit den Nutzenden aus den verschiedenen Leitstellenbereichen soll die Anzahl der bekannten Alarmfaxlayouts erhöht werden. Die direkte Anbindung von Leitstellensystemen ist vermutlich nicht realisierbar, da bestimmte Hersteller mit Geheimhaltungsverträgen arbeiten.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die Arbeiten begannen mit der Alarmanzeige. Da der Server auf Basis von Node.js programmiert wurde, war das Ziel, die Zahl der Abhängigkeiten so gering wie möglich zu halten. In der Vergangenheit kam es immer wieder vor, dass einzelne npm-Module nicht mehr weiterentwickelt oder sogar gelöscht wurden. Daher wurde anfangs kein vollumfängliches Backend Framework verwendet. Mit der Zeit traten aber immer wieder Problemstellungen auf, die selbst bei weniger komplexen Anforderungen nicht innerhalb des Förderzeitraums in Eigenregie zu lösen waren. Dazu zählen insbesondere die Datenbankanbindung und die Benutzerauthentifizierung. Die Alarmzentrale wurde dann von Anfang an auf dem Framework FeathersJS aufgebaut, das die besagten Probleme löste. Zusätzlich vereinfachte es die Synchronisation von Front- und Backend, was es noch attraktiver machte. Daraufhin wurde das Backend für die Alarmanzeige auf Basis von FeathersJS neu geschrieben, ebenso musste die Anbindung des Frontends angepasst werden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Das Projekt wird auf der Projektwebseite[4] vorgestellt. Dort finden sich außerdem Links zur Dokumentation und zum Community-Forum. Der Quellcode wurde auf GitHub[5] veröffentlicht.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Der Projektbeginn war geprägt von der COVID-19-Pandemie und den Gegenmaßnahmen, insbesondere dem sogenannten Lockdown. Dadurch standen keine dedizierten Arbeitsumgebungen wie Co-Working-Spaces zur Verfügung. Diese Situation beeinflusste die Produktivität negativ, sodass die in das Projekt investierten Stunden hinter den Planungen zurück blieben. Im Laufe der Förderung besserte sich die Situation und die Stunden konnten teilweise wieder aufgeholt werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Im Laufe der Förderung wurde sowohl im Landkreis als auch in der Heimatgemeinde des Autors die Einführung eines proprietären Informationssystems vorangetrieben. Dies führte zum Austausch mit verantwortlichen Personen, um ihnen die Entstehung des Projekts Alarmdisplay bekannt zu machen. Dies brachte vor allem neue Erkenntnisse zu potentiellen Nutzenden für einen Pilotbetrieb und weitere Zielsetzungen für neue Funktionen nach der Förderphase.

Augsburg, 30.09.2020

---

[3] https://brightsky.dev/

[4] https://alarmdisplay.org/

[5] https://github.com/alarmdisplay