



netidee

PROJEKTE

Resilionator

Zwischenbericht | Call 15 | Projekt ID 5172

Lizenz: CC-BY

Inhalt

1	Einleitung.....	3
2	Status der Arbeitspakete.....	3
2.1	Arbeitspaket 1 - < <i>Detailplanung und Formales am Projektstart</i> >	3
2.2	Arbeitspaket 2 - < <i>Frontend Prototyp</i> >	3
2.3	Arbeitspaket 3 - < <i>Netzaugmentierung</i> >	4
2.4	Arbeitspakete 4-N.....	4
3	Umsetzung Förderauflagen.....	4
4	Zusammenfassung Planaktualisierung	4
5	Öffentlichkeitsarbeit/ Vernetzung.....	5
6	Eigene Projektwebsite.....	5

1 Einleitung

Im Projekt *Resilionator* möchten wir ein einfach zu verwendendes Open Source Tool für die Resilienz in Computernetzen bereitstellen. Wir haben nun knapp vier Monate Projektlaufzeit hinter uns und möchten das Projekt bis Ende August abschließen.

Resilionator richtet sich sowohl an Wissenschaftler, die Open Source Community, aber gerade auch die Industrie und alltägliche Nutzer ohne großes Vorwissen im Bereich der Netzwerke. Aus wissenschaftlicher Sicht können neue Ideen und Algorithmen eingebunden werden, aus Open Source Sicht die funktionalen Komponenten verbessert werden, und aus industrieller Sicht können Schwachstellen erkannt und verbessert, sowie Feedback an die Community gegeben werden. Das Ziel für normale Nutzer ist es, diesen durch visuelles Feedback einen Einblick in Computernetzwerke und Routingalgorithmen zu gewähren.

Durch die Verlängerung der Projektlaufzeit von Ende Juni auf Ende August können wir auf alle Aspekte detaillierter eingehen.

2 Status der Arbeitspakete

2.1 Arbeitspaket 1 - < *Detailplanung und Formales am Projektstart* >

Das erste Arbeitspaket beschäftigte sich hauptsächlich mit der formellen Planung des Projekts, wie die Erstellung eines Projektplans, Formulierung der Arbeitspakete, der Inbetriebnahme der Projektwebseite und das Verfassen des ersten Blogeintrags.

Dieses Arbeitspaket wurde erfolgreich abgeschlossen, es gab keine Planabweichungen.

2.2 Arbeitspaket 2 - < *Frontend Prototyp* >

In unseren Augen war das zweite Arbeitspaket eines der wichtigsten, denn hier wurden die Entscheidungen über viele Projektgrundlagen getroffen. Angefangen von den Projektspezifikationen (Aussehen und Funktionalität), der Auswahl einer geeigneten Programmiersprache, Frameworks und Bibliotheken bis hin zur Erstellung des ersten Prototypen und einer anschließenden Aussendung von Einladungen zum Feedback.

Eine der wohl wichtigsten Entscheidungen war es die mächtige Python-Library NetworkX in unserem Backend zu benutzen. Diese Entscheidung prägte dann auch die Auswahl der Programmiersprache nämlich Python. Zwar kann man mit Python auch unter anderem Webapps entwickeln, jedoch entschieden wir uns hier für den klassischen Weg einer grafischen Desktop Applikation, welche mit Tkinter der Standard GUI Library von Python umgesetzt wurde. Der Grund hierfür ist, dass eine Desktop GUI Applikation die größte Flexibilität mit kaum Einschränkungen bietet. Python Frameworks für Webapps sind zwar mittlerweile ausgereifter jedoch noch immer mit teilweise größeren Einschränkungen verbunden. Dadurch, dass unsere Applikation "portable" ist d.h. eine Ausführung ist ohne Installation möglich, hat auch der Nutzer kaum Mehraufwand.

Dieses Arbeitspaket wurde erfolgreich abgeschlossen, durch die Verlängerung der Arbeitszeit von Ende April zu Ende Mai konnten wir detailliert auf alles eingehen.

2.3 Arbeitspaket 3 - <Netzaugmentierung>

Das dritte Arbeitspaket beschrieben wir allererst die formalen Spezifikationen des Backends. Der zweite Schritt war dann, diese Spezifikationen umzusetzen. Wichtig bei diesem Arbeitspaket war eine klare Trennung von Front und Backend. Das Backend ist bei uns für alle Berechnungen zuständig und schickt die Resultate zurück an das Frontend. Das Frontend nimmt Benutzereingaben entgegen, schickt diese dann an das Backend, wartet auf die Resultate und bildet diese grafisch oder textuell ab. Ein großer Punkt bei der Umsetzung des Backends, aber auch Frontends, war die Effizienz. Alle von uns verwendeten Funktionen von NetworkX sind am neuesten Stand der Forschung und deswegen so performant wie es aktuell nur möglich ist z.B. bei der Netzaugmentierung. Im Frontend haben wir eine eigene Garbagecollection eingebaut, welche dafür sorgt, dass GUI Elemente die nicht mehr benötigt werden sofort gelöscht werden bzw. falls möglich wiederverwertet werden. Dies führt zu einer enormen Verbesserung der Ressourcennutzung des Programms. Schlussendlich haben wir in diesem Arbeitspaket auch die ersten Feedbacks bekommen und auch schon die ersten Verbesserungen durchgeführt. Eine oft genannter Kritikpunkt war, dass es zu umständlich sei Knoten und Kanten immer über das Menü hinzuzufügen. Deswegen haben wir eine "Quickaccess" Leiste implementiert, welche es den Nutzer ermöglicht dies sofort aus der Hauptansicht zu erledigen. Zusätzlich ist es auch möglich das Layout des Graphen zu ändern, denn dies war auch ein großes Anliegen unserer Tester.

Dieses Arbeitspaket wurde erfolgreich abgeschlossen, durch die Verlängerung der Arbeitszeit von Ende April zu Ende Mai konnten wir detailliert auf alles eingehen.

2.4 Arbeitspakete 4-N

Die weiteren Arbeitspakete werden wir als nächstes angehen. Diese können wir im Vergleich zur Ursprungsplanung detaillierter vertiefen und planen auch eine Visualisierung des Routings. Je nach Zeit könnte aus der Dokumentation auch eine kleine wissenschaftliche Veröffentlichung entstehen, welche der Verbreitung und Popularisierung des Projekts nutzen würde.

3 Umsetzung Förderauflagen

<Keine Förderauflagen>

4 Zusammenfassung Planaktualisierung

Alle Anpassungen des Plan-Excels kurz zusammengefasst

Im ursprünglichen Antrag wurden 7.810€ bewilligt, im ersten Plan waren jedoch nur 4.359€ vorgesehen, da der eingestellte Mitarbeiter einen niedrigeren Stundensatz hatte. Nach Rücksprache möchten wir diesen nun länger beschäftigen, sodass das Projekt sich um 2 Monate verlängert, dabei jedoch immer noch leicht unter den angegebenen Kosten bleibt (falls sich noch minimale weitere Ausgaben ergeben).

Dabei möchten wir die weiteren Arbeitspakete vertiefen, sowie eine Visualisierung des Routings einbauen, und sofern es sich noch zeitlich ausgeht, auch die Dokumentation zu einer kleinen wissenschaftlichen Arbeit ausbauen, um das Projekt innerhalb der wissenschaftlichen Community zu popularisieren.

5 Öffentlichkeitsarbeit/ Vernetzung

Bei unserem Projekt “Resilionator” handelt es sich um ein Open-Source Projekt, deshalb ist uns die Meinung von außerhalb sehr wichtig. Die Einholung von Feedback sahen wir als ersten Schritt in diese Richtung. Wir haben zum Testen Personen aus verschiedenen Kreisen herangezogen, um Resilionator für eine breitere Nutzergruppe attraktiv zu gestalten. Unter den Testern befanden sich Professoren aus der Informatik, Studenten aus mehreren, auch nicht technischen, Studiengängen, Nutzer aus der IT-Branche, aber auch Nutzer welche wenig bis keinen Bezug zu Netzwerken, Computern oder der Informatik allgemein hatten.

In den weiteren Verlauf werden wir eine nächste Feedbackrunde abzuhalten, in der wir neue Nutzer, aber auch Teile der vorherigen Nutzer, einladen. Hier erhoffen wir uns neue Einblicke in das Nutzererlebnis nach der Implementierung der Verbesserungen aus der ersten Feedbackrunde.

6 Eigene Projektwebsite

<https://gitlab.cs.univie.ac.at/ct-projects/resilionator>

Die Förderung durch netidee ist auf der Webseite vermerkt und es wird zur offiziellen Webseite <https://www.netidee.at/resilionator> verlinkt.

Auf der eigenen Projektwebseite ist der aktuelle Code (Stand Ende Mai) erhältlich.