



Decision Critical Bayesian Networks-Editor

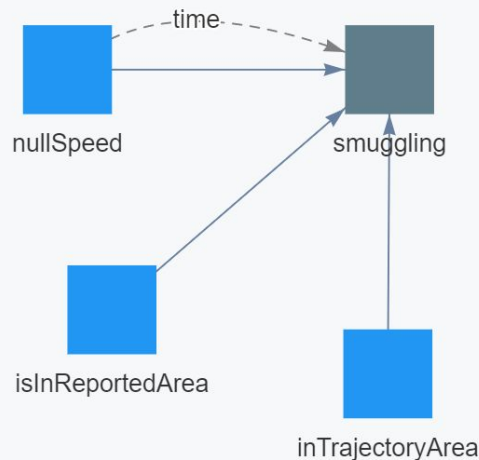
Das Problem

- Überwachung des maritimen Raums
- Viele Schiffe, noch mehr Informationen
 - Manuelle Überwachung nicht möglich
- Expertenwissen zur Einschätzung nötig
- Auswertung in Echtzeit essentiell



Die Lösung

- Schiffsdaten automatisch analysieren
- Von Experten erstellte Analyse verfahren
 - Dynamische Bayessche Netzwerke (DBNs)
 - Graphische Darstellung von bedingten Abhängigkeiten
 - Knoten: Beobachtungen
 - Kanten: Gewichtung einer Beobachtung
 - Dynamisch: Knoten können durch vorherige Zeitschritte bedingt sein
 - Zeitreihe von Beobachtungen
- Integration mit DigLT
 - Daten werden in DCBN ausgewertet
 - Ergebnisse werden angezeigt



Unser Projekt

- Dynamic Bayesian Network-Editor
 - Editor
 - Testumgebung
 - Evidenzformel-Editor (Bsp Evidenzformel : `Ship.speed < 5 && isInArea(Ship.location, area1)`)
- Nutzerverwaltung
 - Superadmin
 - Admin
 - Moderator
- Touchscreen kompatibel

Web GUI

Evidences Editor GUI

Network Editor GUI

Backend

Evidences

Core

Adapter DigLT

Inference Engine
Z.B. AMIDST

IOSB DigLT

Georeferenced
geometries

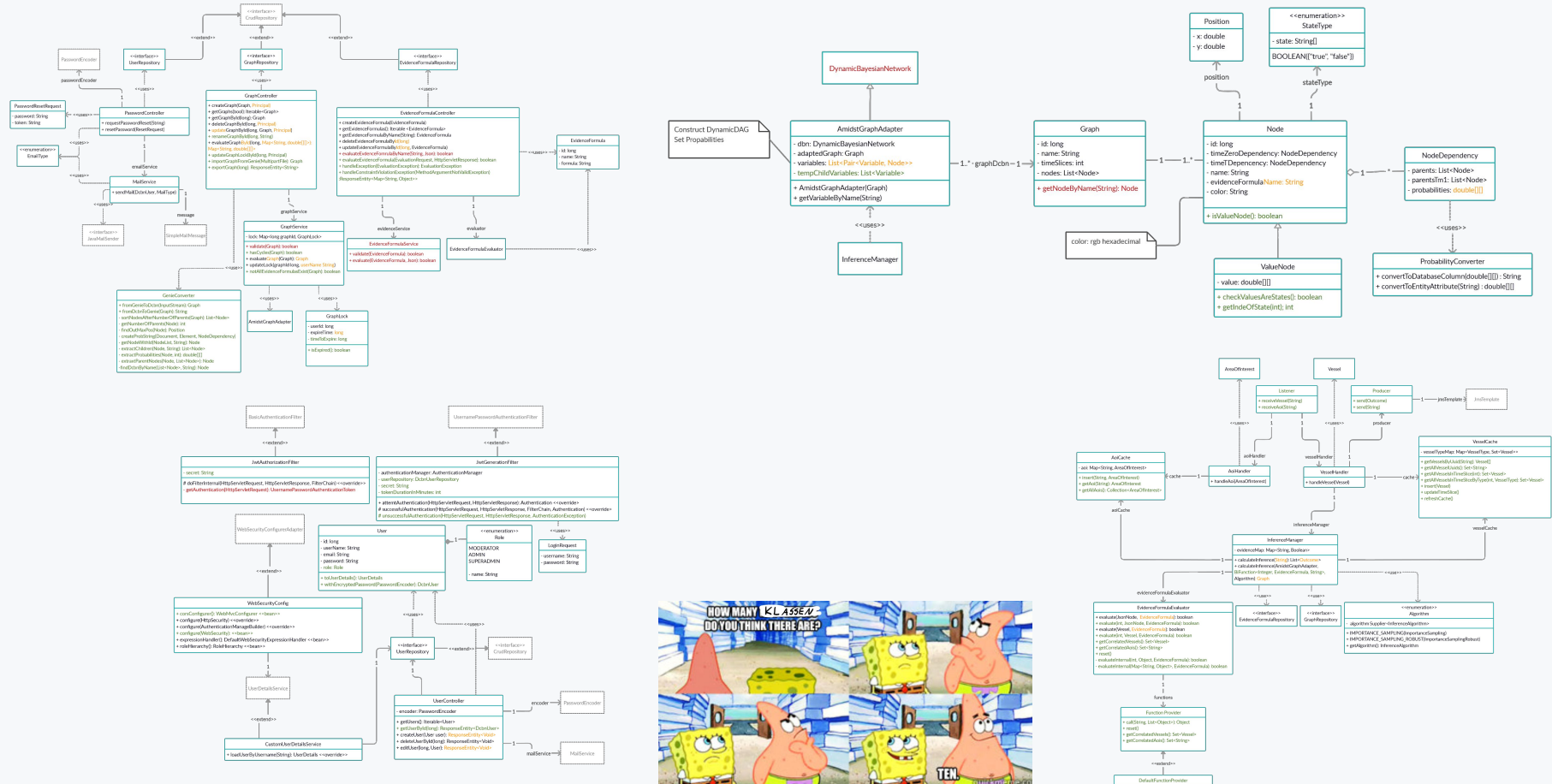
Vessels

Annotations

Alarm

Active MQ

DCBN-Backend / REST-Klassendiagramm



Sprachen

- Backend : Java (Spring Boot)
- Frontend : Vue, TypeScript

Libraries

- AMIDST
- Visjs

Tools

- Git mit GitHub
- GitHub Actions
- IntelliJ IDEA / Visual Studio
- LaTeX, Overleaf
- SonarQube
- Stoplight Studio
- Postman

Statistik

Codezeilen

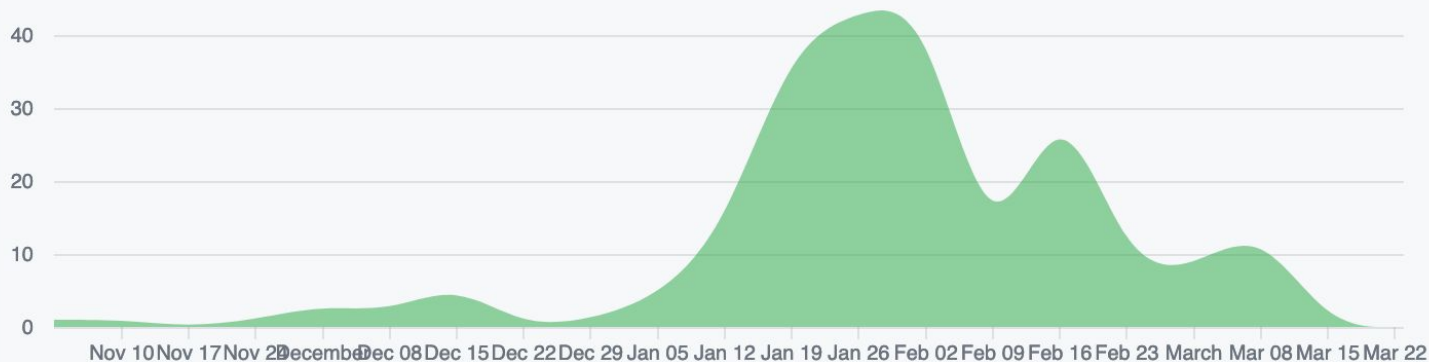
- Java: 4500
- Vue: 3400
- TypeScript: 820

Git

- Commits: 521
- Pull requests: 88

Team

- Entwickler: 5

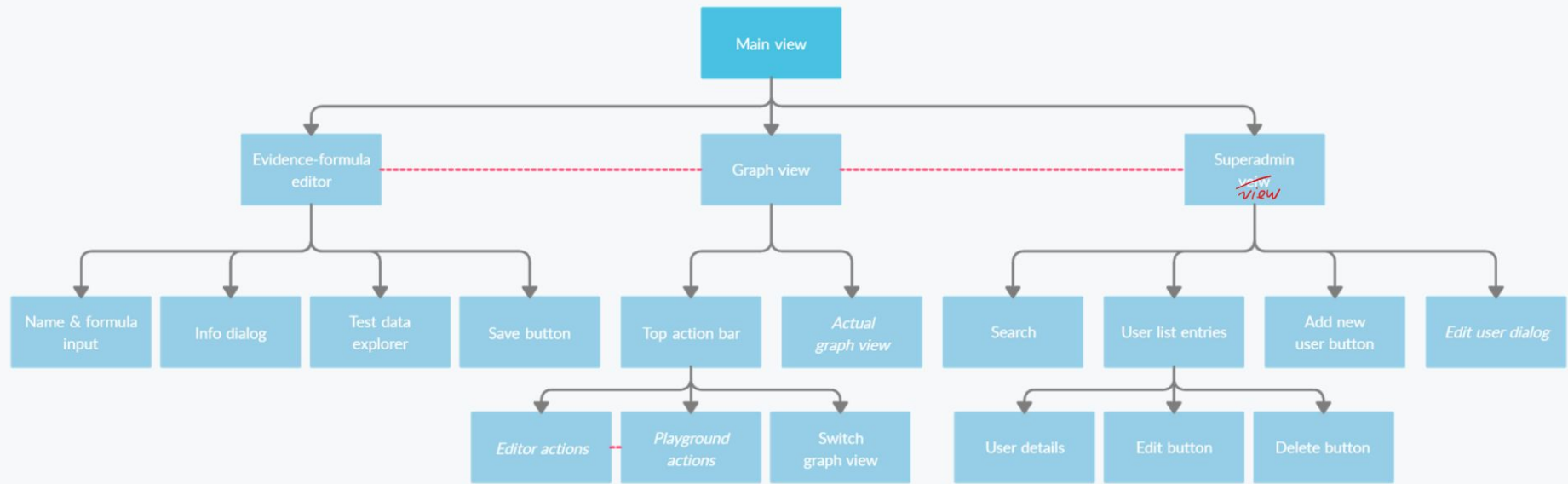


Planungsphase

- 07.11.2020 - 21.11.2020
- Pflichtenheft
- 58 FA
 - 52 “Muss-Kriterien” - alle erfüllt
 - 6 “Kann-Kriterien” - 3 erfüllt
- 3 NFA
 - alle erfüllt
- 18 geplante Testfälle

Entwurfsphase

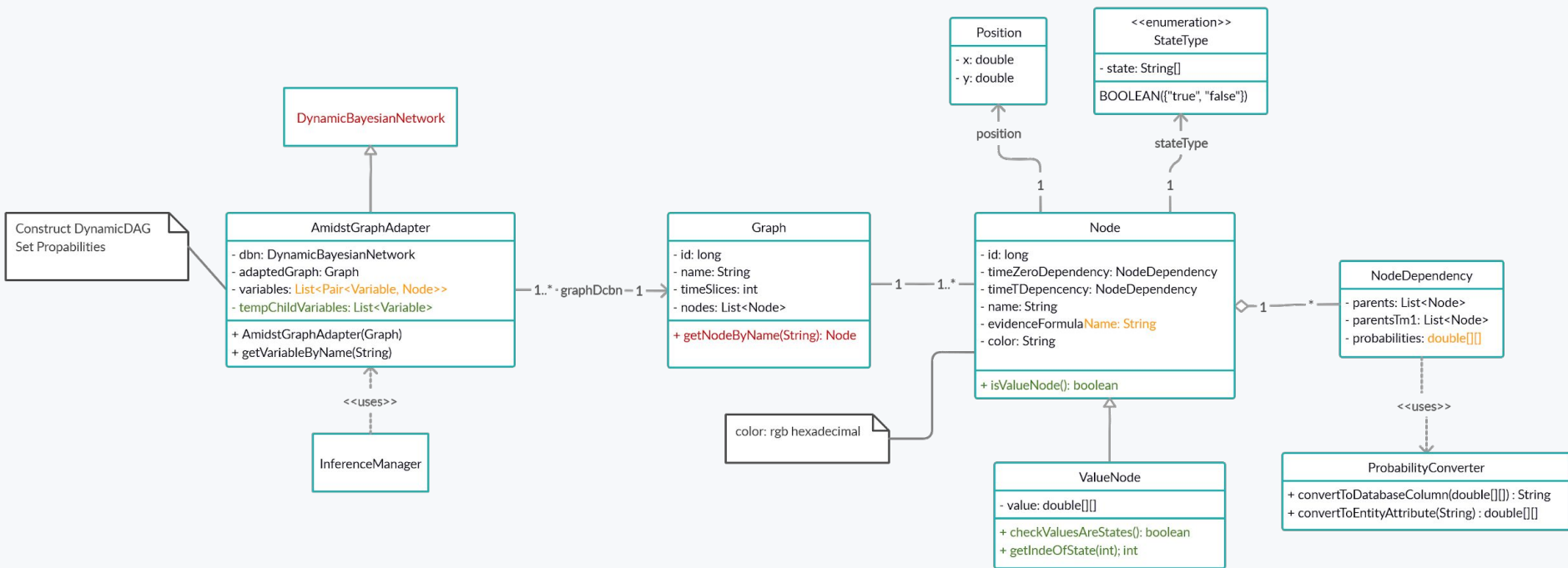
- 25.11.2020 - 19.12.2020
- Frontend
 - Benutzerverwaltung
 - Anmeldeseite
 - Evidenzformel-Erstellung
 - Graph-Editor/Evaluierung
- REST-Schnittstelle
 - Anfragen vom Frontend an das Backend
- Backend
 - Benutzererstellung und -anmeldung
 - DBNs
 - Auswertung von Evidenzformeln
 - Kommunikation mit IOSB Server und der Inferenz-Engine



Frontend-Main View

Implementierungsphase

- 07.01.2020 - 20.02.2020
- Methoden von 23 Klassen angepasst
 - 61 neue Methoden
 - 6 Methoden entfernt
 - 13 Methoden geändert
- Modularität verbessert
- Struktur aus der Entwurfsphase gleich geblieben



Änderungen des Graphs und des Graph Adapters beim Implementieren

Qualitätssicherung

- 05.03.2020 - 19.03.2020
- Unit-Tests, Spring Integration Tests Backend
- 75% Testabdeckung (Getter/Setter und Datentypen nicht getestet)
- Statische Analyse mit SonarQube
- Bugs und Schwachstellen bereinigt

☆ **dcbn** Passed

7 **C**
Bugs

8 **D**
Vulnerabilities

0.0% **E**
Hotspots Reviewed

9 **A**
Code Smells



☆ **dcbn** Passed

0 **A**
Bugs

0 **A**
Vulnerabilities

100% **A**
Hotspots Reviewed

0 **A**
Code Smells

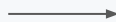
☆ **DCBNFrontend** Passed

2 **B**
Bugs

0 **A**
Vulnerabilities

0.0% **E**
Hotspots Reviewed

6 **A**
Code Smells



☆ **DCBNFrontend** Passed

0 **A**
Bugs

0 **A**
Vulnerabilities

100% **A**
Hotspots Reviewed

0 **A**
Code Smells

SonarQube Statische Analyse vor und nach der Qualitätssicherung

Lernerfahrungen

- Dynamische Bayes'sche Netze
- Planung durch UML-Diagramme
- Spring Boot, Java, TypeScript, Vue
- Umgang mit Git und kollaborativer Versionsverwaltung



Demo

