



**TN 22**

## **Systemarchitektur**

Jörn Möltgen & Reiner Borchert

Version 1.0

Stand: 07.05.2003

### **Inhaltsverzeichnis**

|     |  |    |
|-----|--|----|
| 1   | Übersicht.....   | 2  |
| 2   | Komponenten.....   | 2  |
| 2.1 | Graphische Benutzerschnittstelle (GUI).....              | 2  |
| 2.2 | Controller (Programmsteuerung) .....                     | 3  |
| 2.3 | Szenarienmanager.....                                    | 4  |
| 2.4 | Interaktionsmodule – Visualisierung und Bearbeitung..... | 5  |
| 2.5 | Wissensverarbeitung .....                                | 6  |
| 2.6 | Wissensbasen und Ontologien .....                        | 7  |
| 2.7 | Analyse-Module – GIS und Modelle .....                   | 8  |
| 2.8 | Geodatenbanken .....                                     | 8  |
| 2.9 | FLUMAGIS-Datenbank .....                                 | 9  |
| 3   | Beispiel-Szenario .....                                  | 10 |

## **EINLEITUNG**

Der Aufbau eines Softwareproduktes, wie das des FLUMAGIS-Prototypen, ist in einer groben Annäherung schon in einer sehr frühen Phase zu planen. Mit der Systemarchitektur fallen wichtige Entscheidungen zum System, die für die weiteren Spezifizierungen grundlegend sind und die in späteren Projektphasen schwer rückgängig gemacht werden können.

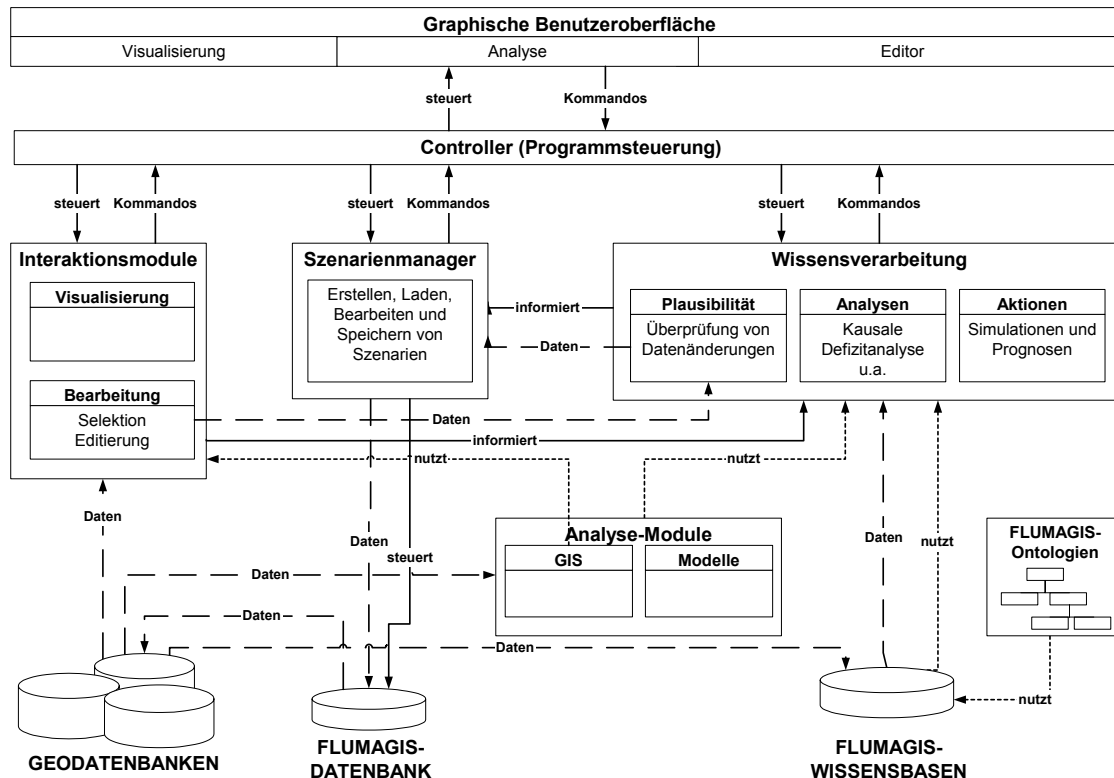
Eine Systemarchitektur sollte folgende Fragen beantworten:

- Aus welchen (Sub-)Komponenten besteht das System?
- Welche Informationsflüsse gibt es zwischen den einzelnen Komponenten?
- Welche Schnittstellen bestehen zwischen den Komponenten?
- Welche Schnittstellen gibt es zu externen Systemen?
- Welche Schnittstellen müssen entwickelt werden?

Basierend auf der Systemarchitektur müssen die Anforderungen an Komponenten, Schnittstellen, Datentypen und Methoden für das Softwaredesign weiter spezifiziert werden. Diese Spezifikationen dienen als Grundlage für die eigentlichen Implementierungsarbeiten. Daher muss die Architektur in der Aufzählung ihrer Elemente vollständig und konsistent sein.

Die Systemarchitektur wird in einer Übersichtsgraphik und erklärenden Texten beschrieben. Die Informationsflüsse zwischen den Komponenten werden in der Graphik als Verbindungslinien zwischen den Komponenten dargestellt, wobei zu jeder Komponente und zu jeder Verknüpfung ein beschreibender Text gehört.

# 1 ÜBERSICHT



## 2 KOMPONENTEN

### 2.1 Graphische Benutzerschnittstelle (GUI)

| Graphische Benutzeroberfläche |         |        |
|-------------------------------|---------|--------|
| Visualisierung                | Analyse | Editor |

Die GUI umfasst alle nach außen sichtbare Programmteile (Oberfläche). Über Menüs, Buttons, Werkzeugleisten, Dialogboxen u.a. kann der Benutzer Programmfunktionen aufrufen. Diese sind inhaltlich gruppiert (Visualisierung, Analyse, Editierfunktionen,...).

Die GUI nimmt die Benutzereingaben (z.B. Menuselektion, Mausklick) entgegen und leitet sie als Kommandos an den *Controller* weiter, der auch für die Steuerung der GUI verantwortlich ist (z.B. Aktivierung und Deaktivierung von Menüpunkten und Buttons).

#### Schnittstellen:

- **GUI → Controller** (Kommandos): Jede Komponente, die eine Programmfunktion auslösen soll, wird mit einem eindeutigen Kommando (z.B. einer Funktionsnummer) verknüpft. Wenn die Komponente vom Benutzer angewendet wird (z.B. durch Mausklick), meldet die GUI dies in Form des Kommandos an den Controller weiter.

- **Controller** → **GUI** (Steuerung): Der Controller überprüft ständig den aktuellen Programmkontext und aktiviert jeweils die aktuell möglichen und sinnvollen Funktionen (bzw. die entsprechenden GUI-Komponenten) und deaktiviert die aktuell nicht zur Verfügung stehenden.

## 2.2 Controller (Programmsteuerung)

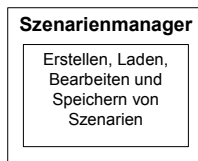
|                                       |
|---------------------------------------|
| <b>Controller (Programmsteuerung)</b> |
|---------------------------------------|

Der Controller interpretiert die Kommandos der Benutzerschnittstelle und ruft die damit verbundenen Programmfunktionen auf, verbindet also die visuellen Elemente des GUI mit Programmfunktionen. Auch von anderen Systemkomponenten kann der Controller Kommandos empfangen und diese wie die von der GUI erzeugten.

### Schnittstellen:

- **GUI** → **Controller** (Kommandos): Jede Komponente, die eine Programmfunktion auslösen soll, wird mit einem eindeutigen **Kommando** (z.B. einer Funktionsnummer) verknüpft. Wenn die Komponente vom Benutzer angewendet wird (z.B. durch Mausklick), meldet die GUI dies in Form des Kommandos an den Controller weiter.
- **Controller** → **GUI** (Steuerung): Der Controller überprüft ständig den aktuellen Programmkontext und aktiviert jeweils die aktuell möglichen und sinnvollen Funktionen (bzw. die entsprechenden GUI-Komponenten) und deaktiviert die aktuell nicht zur Verfügung stehenden.
- **Controller** → **aktive Systemkomponenten** (Interaktionsmodule, Szenarienmanager, Wissensverarbeitung) (Steuerung): Der Controller enthält eine Liste der aufrufbaren Programmfunktionen der aktiven Systemkomponenten. Diese Funktionen sind mit Kommandos verknüpft. Wenn der Controller ein Kommando empfängt, kann er die dazu gehörige Funktion ausfindig machen und diese starten.
- **aktive Systemkomponenten** (Interaktionsmodule, Szenarienmanager, Wissensverarbeitung) → **Controller** (Kommandos): Die aktiven Systemkomponenten können von sich aus Kommandos an den Controller schicken, die dieser wiederum interpretiert und in Funktionsaufrufe umwandelt.

## 2.3 Szenarienmanager

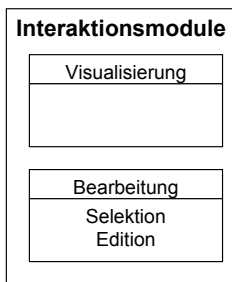


Der Szenarienmanager registriert im Hintergrund alle von der *Wissensverarbeitung* gemeldeten Veränderungen der Geodaten (Editor-Vorgänge, Simulationen) sowie die Ergebnisse von Analysen und Prognosen (Erstellen eines Szenarios). Als Benutzerfunktionen bietet er das Laden, Bearbeiten und Speichern von Szenarien an.

### Schnittstellen:

- **Controller** → **Szenarienmanager** (Steuerung): Der Controller kann drei Funktionen des Szenarienmanagers starten: Laden, Bearbeiten und Speichern eines Szenarios.
- **Szenarienmanager** → **Controller** (Kommandos): Der Szenarienmanager schickt nach dem Laden oder Bearbeiten eines Szenarios eine Nachricht an den Controller, der diese wiederum an alle anderen Komponenten weiterleitet. So wird sicher gestellt, dass das System (z.B. die Visualisierung) insgesamt angemessen auf Änderungen von Objektdaten reagiert.
- **Wissensverarbeitung** → **Szenarienmanager** (Daten und Information): Die Wissensverarbeitung informiert den Szenarienmanager über Änderungen der Objektdaten sowie Analyse- und Prognoseergebnisse. Der Szenarienmanager protokolliert diese Informationen als eine fortlaufende Kette von Ereignissen.
- **Szenarienmanager** → **FLUMAGIS-Datenbank** (Daten): Der Szenarienmanager speichert ein Szenario temporär oder dauerhaft in der FLUMAGIS-Datenbank. Dauerhaft gespeicherte Szenarien bekommen einen Namen, eine optionale Beschreibung und Informationen über Zeitpunkt der Erstellung, der Modifikationen und die beteiligten Nutzer.
- **Szenarienmanager** → **FLUMAGIS-Datenbank** (Steuerung): Der Szenarienmanager steuert die FLUMAGIS-Datenbank, indem er die Daten eines geladenen Szenarios für die Geodatenbanken freigibt und damit verfügbar/zugänglich macht. Die „fiktiven“ Szenariodaten sollen vom System wie „reale“ Daten behandelt werden.

## 2.4 Interaktionsmodule – Visualisierung und Bearbeitung



Die **Visualisierung** stellt die Geodaten dem Maßstab angemessen dar, bietet Möglichkeiten der Ausschnittswahl (Zoom / Pan) sowie der Wahl der Betrachterposition (3D). Sie führt automatisch Maßstabswechsel durch, wenn dies erforderlich ist. Die symbolische Darstellung von Geobjekten wird ebenfalls von der Visualisierung dem Maßstab und dem aktuellen Kontext angepasst; sie kann vom Benutzer konfiguriert werden.

Das **Bearbeitungsmodul** von visuellen Elementen umfasst die Selektionswerkzeuge sowie Werkzeuge zum Editieren von Geobjekten und deren Attributen.

Die **Selektion** kann von Programmfunktionen auf bestimmte Objektklassen eingegrenzt werden. Außerdem kann die Selektion sich auf ganze Bereiche / Abschnitte beziehen. Die Wissensverarbeitung bekommt auf Anfrage eine Liste der selektierten Objekte zugesandt.

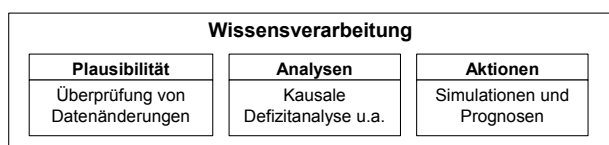
Die **Editierung** von Geobjekten umfasst sowohl Änderungen der Geometrie (Position, Größe) der Objekte wie auch die Editierung der Attribute in Tabellenform. Außerdem können neue Objekte erzeugt und mit Attributwerten versehen wie auch vorhandene Objekte gelöscht werden. All diese Editiermöglichkeiten können jedoch von der Wissensverarbeitung in Abhängigkeit vom Objekttyp eingeschränkt werden, damit nicht erlaubte Operationen unterbleiben.

### Schnittstellen:

- **Controller** → **Interaktionsmodule** (Steuerung): Der Controller kann die Programmfunktionen der Interaktionsmodule aufrufen.
- **Interaktionsmodule (Bearbeitungsmodul)** → **Controller** (Kommandos): Nach erfolgter Selektion oder Editierung sendet das Bearbeitungsmodul eine entsprechende Nachricht an den Controller, der diese wiederum an alle anderen Komponenten weiterleitet (Selektion) oder direkt die Plausibilitätskontrolle der Wissensverarbeitung aufruft. Die Datenänderung wird erst nach der Kontrolle akzeptiert.
- **Selektionsmodul** → **Wissensverarbeitung** (Information): Die Wissensverarbeitung fragt bei Bedarf bei dem Selektionsmodul an, ob Geobjekte selektiert worden sind, und bekommt als Antwort eine Liste der aktuell selektierten Objekte zurückgeliefert.
- **Editierung** → **Plausibilitätsprüfung** (Daten): Auf Anforderung der Wissensverarbeitung liefert die Editierung die veränderten Daten an die Plausibilitätskontrolle.

- **GIS** → **Interaktionsmodule** (Nutzung): Die Interaktionsmodule können bei Bedarf GIS-Funktionen aufrufen / nutzen.
- **Geodatenbanken** → **Interaktionsmodule** (Daten): Zur Visualisierung und Bearbeitung greifen diese Module auf die Geodatenbanken zu.

## 2.5 Wissensverarbeitung



Die **Plausibilitätskontrolle** tritt nach jeder Änderung von Geodaten und Attributen in Aktion. Insbesondere werden Datenänderungen nach einem Editiervorgang kontrolliert; aber auch die sich bei Aktionen und Simulationen ergebenden Datenänderungen werden überwacht.

Die Plausibilitätskontrolle soll offensichtlich unmögliche oder absolut sinnlose Vorgänge verhindern. Ungünstige oder schädliche, aber mögliche Vorgänge soll sie erlauben, aber ggf. mit einer Warnmeldung kommentieren.

Bei akzeptierten Änderungen liefert sie anschließend die veränderten Daten an den Szenarienmanager weiter.

Das **Analysemodul** enthält die Programmfunktionen, mit denen der Istzustand der Geoobjekte analysiert werden kann. Die wichtigste Analyse ist die Defizitanalyse, die ökologische Defizite (Abweichungen vom angestrebten Zustand) durch Vergleich der Objektdaten mit parametrisierten Leitbildern durchführt. Leitbilder sind in den Wissensbasen gespeichert.

Das **Aktionsmodul** kann einerseits auf Anforderung bestimmte **Maßnahmen** durchführen (automatisierte Editiervorgänge, die ebenso wie „manuelle“ die Plausibilitätskontrolle durchlaufen), andererseits aber auch **Simulationen** starten, die basierend auf dem Wissen der Wissensbasen versuchen, für einen festgelegten Zeitrahmen Voraussagen über die künftige Entwicklung nach durchgeführten Maßnahmen oder auf Grund von veränderten Modellen oder Rahmenbedingungen zu treffen. Daraus resultierende Prognosen werden detailliert in einem Informationsfenster (HTML-Browser) ausgegeben.

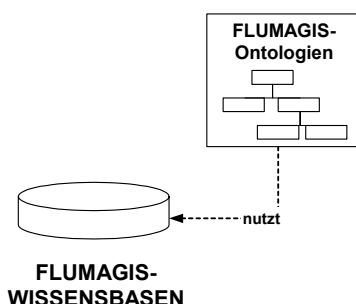
Alle Teile der Wissensverarbeitung informieren den **Szenarienmanager** über Datenänderungen und Analyse- sowie Prognoseergebnisse.

### Schnittstellen:

- **Controller** → **Wissensverarbeitung** (Steuerung): Der Controller kann die Programmfunktionen der Wissensverarbeitung aufrufen (z.B. Analyse oder Simulation starten).

- **Wissensverarbeitung** → **Controller** (Kommandos): Nach erfolgter Aktion, Analyse oder Simulation sendet die Wissensverarbeitung eine entsprechende Nachricht an den Controller, der diese wiederum an alle anderen Komponenten weiterleitet.
- **Selektionsmodul** → **Wissensverarbeitung** (Information): Die Wissensverarbeitung fragt bei Bedarf bei dem Selektionsmodul an, ob Geobjekte selektiert worden sind, und bekommt als Antwort eine Liste der aktuell selektierten Objekte zurückgeliefert.
- **Editierung** → **Plausibilitätsprüfung** (Daten): Auf Anforderung der Wissensverarbeitung liefert die Editierung die veränderten Daten an die Plausibilitätskontrolle.
- **Wissensverarbeitung** → **Szenarienmanager** (Daten und Information): Die Wissensverarbeitung informiert den Szenarienmanager über Änderungen der Objektdaten sowie Analyse- und Prognoseergebnisse. Der Szenarienmanager protokolliert diese Informationen als eine fortlaufende Kette von Ereignissen.
- **Analyse-Module** → **Wissensverarbeitung** (Nutzung): Die Wissensverarbeitung kann bei Bedarf GIS-Funktionen und vorhandene Modelle aufrufen / nutzen.
- **Wissensbasen/Ontologien** → **Wissensverarbeitung** (Daten, Nutzung): Die Wissensbasen machen die Daten der Geodatenbanken für die Wissensverarbeitung nutzbar, indem sie sie ontologisch strukturieren und Relationen herstellen. Auch das Fachwissen liegt in den Wissensbasen vor und kann von der Wissensverarbeitung genutzt werden.

## 2.6 Wissensbasen und Ontologien



Die Ontologien sind eigentlich ein zentraler Bestandteil der Wissensbasen. Sie geben vor, die die Daten (Geodaten und Attribute) strukturiert sind und wie sie zueinander in Beziehung stehen (Relationen). Da die Wissensverarbeitung genau diese Informationen nutzt, benötigt sie die Wissensbasen, um auf die Daten der Geodatenbanken zuzugreifen. Erst die Aufbereitung der Daten durch die Wissensbasen machen die Daten für

die Wissensverarbeitung brauchbar.

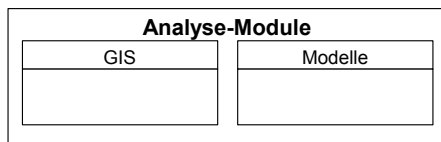
Ein wesentlicher Bestandteil der Wissensbasen ist daher das Modul, das die Daten der Geodatenbanken zu Objekten, Eigenschaften und Relationen im Sinne der Ontologie macht.

Daneben sind auch Fachwissen, Methoden und Leitbilder ontologisch strukturiert und beschrieben.

**Schnittstellen:**

- **Wissensbasen/Ontologien → Wissensverarbeitung** (Daten, Nutzung): Die Wissensbasen machen die Daten der Geodatenbanken für die Wissensverarbeitung nutzbar, indem sie sie ontologisch strukturieren und Relationen herstellen. Auch das Fachwissen liegt in den Wissensbasen vor und kann von der Wissensverarbeitung genutzt werden.
- **Geodatenbanken → Wissensbasen** (Daten): Die Geodatenbanken liefern Geodaten und Attribute als „Rohmaterial“ für die Wissensbasen.

## 2.7 Analyse-Module – GIS und Modelle

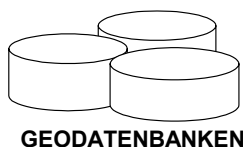


Die Analyse-Module stellen einen Pool von Funktionen bereit, die von anderen Programmkomponenten genutzt werden können. Es handelt sich dabei um Standard-GIS-Funktionen wie auch um fachspezifische Modelle.

**Schnittstellen:**

- **Analyse-Module → Wissensverarbeitung und Interaktionsmodule** (Nutzung): Wissensverarbeitung und Interaktionsmodule können bei Bedarf GIS-Funktionen und vorhandene Modelle aufrufen / nutzen.
- **Geodatenbanken → Analyse-Module** (Daten): Die Geodatenbanken liefern Geodaten und Attribute als „Rohmaterial“ für die Analyse-Module.

## 2.8 Geodatenbanken



Die Geodatenbanken enthalten alle Daten des Ist-Zustands (Geodaten, Objekt-Attribute). Sie werden zu Beginn des Projektes mit den Daten des zu bearbeitenden Flussgebiets bestückt und sind, solange diese Daten aktuell sind, schreibgeschützt. Eine ggf. notwendige Aktualisierung führt der Systemadministrator durch.

Systemadministrator durch.

Die Geodatenbanken sind mit der FLUMAGIS-Datenbank verbunden. Diese kann die Daten eines oder mehrerer Szenarien freigeben. Diese „virtuellen“ Daten überschreiben für das System die Istdaten der Geodatenbank. Das System geht dann von der Realität der fiktiven Daten aus.

### Schnittstellen:

- **Geodatenbanken** → **Interaktionsmodule** (Daten): Zur Visualisierung und Bearbeitung greifen diese Module auf die Geodatenbanken zu.
- **Geodatenbanken** → **Wissensbasen** (Daten): Die Geodatenbanken liefern Geodaten und Attribute als „Rohmaterial“ für die Wissensbasen.
- **Geodatenbanken** → **Analyse-Module** (Daten): Die Geodatenbanken liefern Geodaten und Attribute als „Rohmaterial“ für die Analyse-Module.
- **FLUMAGIS-Datenbank** → **Geodatenbanken** (Daten): Die FLUMAGIS-Datenbank informiert die Geodatenbank über gegenüber dem Istzustand veränderten Daten (Szenarien).

## 2.9 FLUMAGIS-Datenbank



**FLUMAGIS-DATENBANK**

Die FLUMAGIS-Datenbank enthält (virtuell) veränderte Daten, die durch Editierung oder Simulation entstanden sind. Die Daten sind als Szenarien organisiert; aus einem Szenario geladene Daten überschreiben die Istdaten der Geodatenbanken. Szenarien werden vom Szenarienmanager in der FLUMAGIS-Datenbank abgelegt. Ein im Entstehen begriffenes Szenario wird zunächst temporär gespeichert und kann vom Benutzer dauerhaft gesichert werden.

### Schnittstellen:

- **FLUMAGIS-Datenbank** → **Geodatenbanken** (Daten): Die FLUMAGIS-Datenbank informiert die Geodatenbank über gegenüber dem Istzustand veränderten Daten (Szenarien).
- **Szenarienmanager** → **FLUMAGIS-Datenbank** (Daten): Der Szenarienmanager speichert ein Szenario temporär oder dauerhaft in der FLUMAGIS-Datenbank. Dauerhaft gespeicherte Szenarien bekommen einen Namen, eine optionale Beschreibung und Informationen über Zeitpunkt der Erstellung, der Modifikationen und die beteiligten Nutzer.
- **Szenarienmanager** → **FLUMAGIS-Datenbank** (Steuerung): Der Szenarienmanager steuert die FLUMAGIS-Datenbank, indem er die Daten eines geladenen Szenarios für die Geodatenbanken freigibt und damit verfügbar/zugänglich macht. Die „fiktiven“ Szenariodaten sollen vom System wie „reale“ Daten behandelt werden.

### 3 BEISPIEL-SZENARIO

Zur Verdeutlichung der Informations- und Datenflüsse wird hier ein kleines Szenario entworfen:

Ein Planer will den Oberlauf des Gewässers „Bb“ auf strukturelle Defizite überprüfen und ggf. Maßnahmen zu deren Beseitigung ergreifen.

Arbeitsschritte:

1. Der Planer **zoomt** den Gewässer-Bereich heraus, den er untersuchen will (Controller → Interaktionsmodule: Visualisierung).
2. Die zu untersuchenden Abschnitte werden **selektiert** (Controller → Interaktionsmodule: Selektion).
3. Die Funktion **Defizitanalyse** wird gestartet (Controller → Wissensverarbeitung: Analysen). Es erscheint eine Dialogbox, in der der Planer einige Vorgaben machen kann:
  - Er wählt das zu verwendende **Leitbild** aus, in diesem Fall das Leitbild zur Gewässer-Strukturgüte. (Leitbilder enthalten Informationen über die zu benutzenden Daten sowie über mögliche Defizittypen.)
  - Falls noch kein Bereich selektiert wurde, kann der Planer dies jetzt nachholen (→ Schritt 2.).
4. Nach dem Start der **Analyse** werden die Daten der selektierten Gewässerabschnitte mit den Sollwerten des Leitbildes verglichen. Dabei treten Differenzen zu Tage, für die jeweils spezielle **Defizitobjekte** erzeugt werden. Defizitobjekte enthalten zunächst nur Informationen über den *Defizittyp* und das Objekt, für das das Defizit festgestellt wurde. In diesem Fall wird als Defizit eine Laufeinschränkung im Abschnitt 2 festgestellt.
5. Für alle Defizitobjekte wird eine **Ursachenanalyse** durchgeführt. Hierfür wird im Datenbestand des betroffenen Bereichs nach Geobjekten gesucht, die als Verursacher des Defizits in Frage kommen. In diesem Fall findet die Analyse ein Rohr, durch das das Gewässer geleitet wird.
6. Das Defizit wird **visuell dargestellt**. Das verursachende Rohr wird ebenfalls hervorgehoben. Außerdem wird das Analyseergebnis in einem **HTML-Browser** dargestellt. Alle relevanten Daten sind über Links erreichbar. Außerdem schlägt die Analyse geeignete **Maßnahmen** vor, mit denen das gefundene Defizit beseitigt werden könnte. Der **Szenarien-Manager** registriert das Ergebnis der Analyse im Hintergrund.

7. Der Planer kann über das Browserfenster oder über das Menü/Buttons (Graphische Benutzeroberfläche) die vorgeschlagene Maßnahme „Rohr entfernen“ auswählen. Die Wissensverarbeitung enthält Funktionen zur (virtuellen) Umsetzung der Maßnahme (Controller → Wissensverarbeitung: Aktionen). Die **Plausibilitätskontrolle** überprüft vorher, ob die gewünschte Maßnahme überhaupt möglich ist und ob sie voraussichtlich sinnvoll ist. Unmögliche Maßnahmen werden abgelehnt, voraussichtlich nicht sinnvolle mit einer Warnung kommentiert.
8. Die **Maßnahme** wird, wenn die Plausibilitätskontrolle sie akzeptiert hat, virtuell umgesetzt (Objekt „Rohr“ wird entfernt, Defizit gelöscht). Der **Szenarien-Manager** registriert die Aktion im Hintergrund.
9. Der Planer kann nun die Funktion „Simulation/Prognose“ starten (Controller → Wissensverarbeitung: Aktionen). In einer Dialogbox kann er Vorgaben machen, z.B. über die zu berücksichtigenden Aspekte und den zu simulierenden Zeitrahmen. Die **Simulation** versucht anschließend, die Folgen der Maßnahme abzuschätzen, wenn möglich, als zeitlich differenzierter Ablauf. Das Ergebnis der Simulation wird als **Prognose** bezeichnet. Es wird ebenfalls wenn möglich visualisiert und im HTML-Browser angezeigt. Der **Szenarien-Manager** registriert den Verlauf der Simulation im Hintergrund.
10. Dieses Szenario kann, wenn gewünscht, in der FLUMAGIS-Datenbank abgelegt werden. Dabei werden alle Aktionen, Ergebnisse und Datenänderungen, die im Hintergrund vom **Szenarienmanager** protokolliert wurden, gespeichert und können später entsprechend wieder hergestellt werden.