

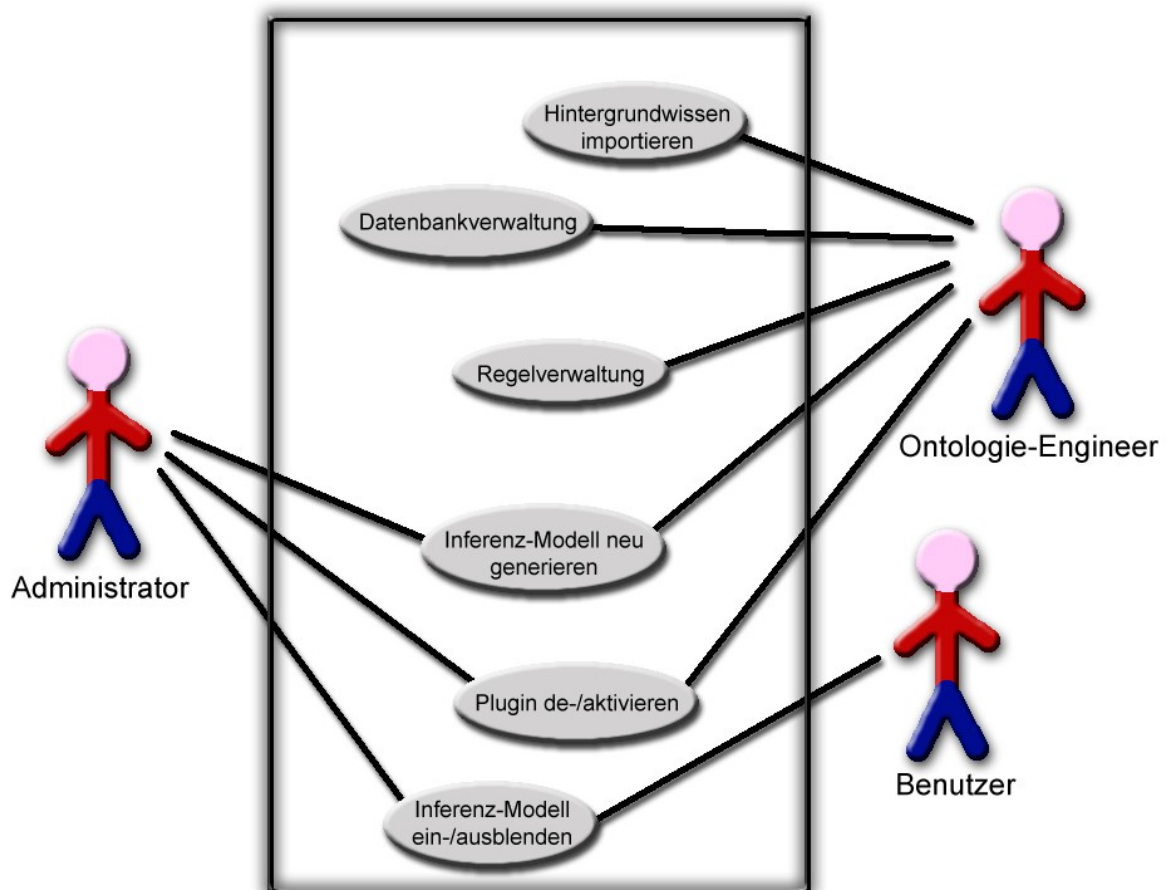
Endabgabe Entwurfsbeschreibung (Update)

1. Allgemeines

Ziel dieses Projektes ist es, eine Erweiterung für das Ontowiki-System zu entwickeln. Es handelt sich dabei um ein Plugin. Die Hauptaufgabe des Plugins liegt darin, anhand von bestimmten, vordefinierten Regeln, implizite Daten aus bereits vorhandenen expliziten Daten zu gewinnen.

Die Regeln lassen sich über eine Regelverwaltungs-Komponente zu einzelnen Property's erstellen. Das Abarbeiten der Regeln resultiert in der Generierung einer weiteren Wissensbasis im Ontowiki, welche dann zusätzlich inferierte Daten beinhaltet. Diese Daten werden in Form von URIs dargestellt und sollen später durch eine zusätzliche Markierung deutlich von expliziten Daten unterscheidbar sein.

2. Produktübersicht

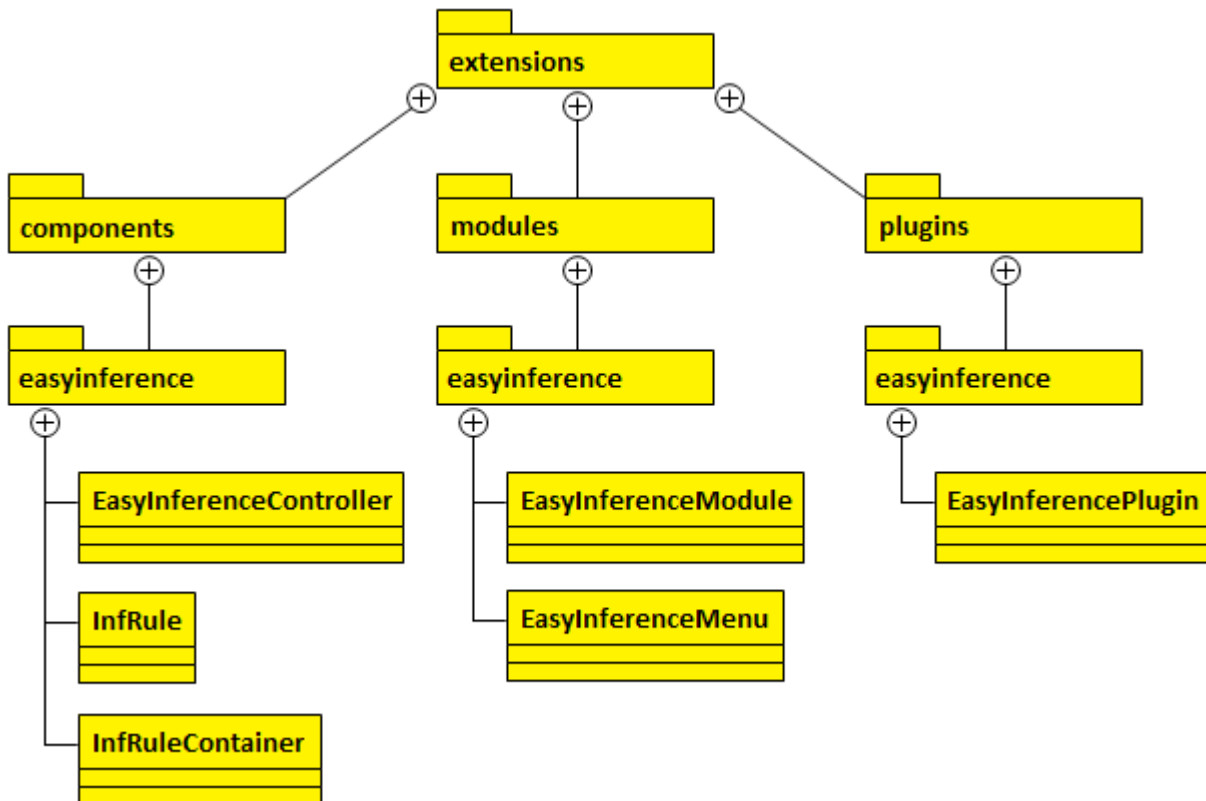


Für das EasyInference-Plugin sind drei Arten von Nutzer relevant. Der Ontologie-Engineer, der Administrator und der normale Benutzer. Hierbei spielt jedoch der Administrator die Hauptrolle, der Ontologie-Engineer nur dann eine übergeordnete Rolle, wenn es zu Problemen mit der Regelwissensbasis

kommt. Der normale Benutzer kann die durch das Plugin erzeugten Inferenzen nutzen und hat somit eine erweiterte Suchmöglichkeit, falls das Inferenzmodell aktiviert ist.

Dem Administrator stehen alle Funktionalitäten des Plugins zur Verfügung, er verwaltet die Regeln, die Datenbank und er ist dafür verantwortlich, dass die Ontologie gültig bleibt. Er kann das Inferenzmodell neu generieren und das Plugin de- bzw. aktivieren.

3. Grundsätzliche Entwurfsprinzipien des Gesamtsystems



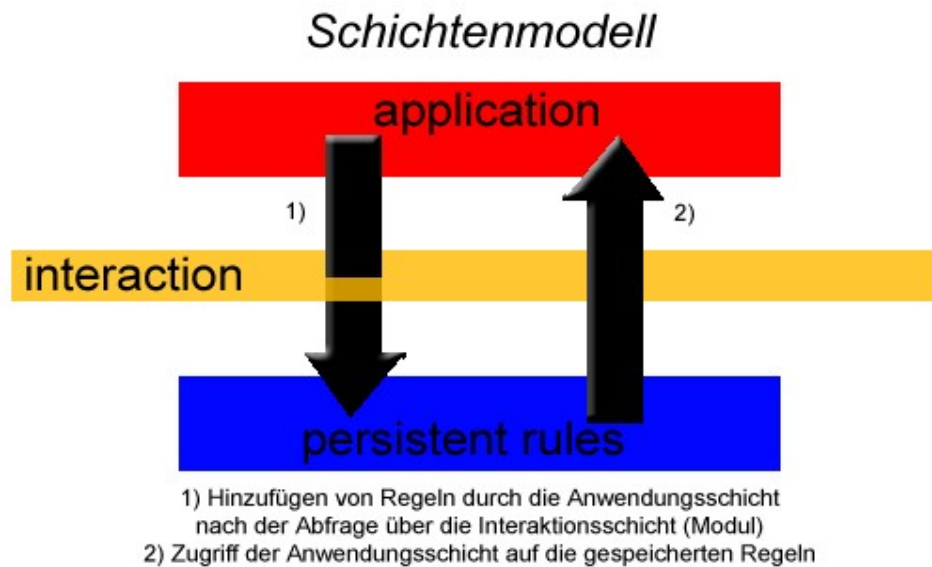
Das EasyInference-Plugin ist eine Extension zu Ontowiki. Da Ontowiki strikte Regeln besitzt welche Art von Extension welche Funktionalität besitzt wird das EasyInference-Plugin in 3 Teile aufgeteilt. Es besteht aus einer Ontowiki-Komponente, einem -Modul und einem -Plugin.

Komponenten, Module und Plugins werden in unterschiedlichen Ordnern gespeichert. Jeder dieser Ordner besitzt einen easyinference-Unterverzeichnis, in dem die jeweiligen Klassen des EasyInference-Plugins zu finden sind.

Die Komponente übernimmt dabei die Aufgabe des Controllers in der MVC-Architektur (Model-View-Controller). Dieser Controller besitzt Actions, die das Inferenzmodell erstellen und manipulieren können. Das Modul übernimmt die Aufgabe, dem Benutzer die Möglichkeit zu geben, Regeln auf Properties und Klassen direkt über das User-Interface anzuwenden. Es zeigt ein kleines Fenster, in dem ein Benutzer die Regeln verwalten kann. Dazu zählt das Anwenden von vorgegebenen Regeln, aber auch das Rückgängigmachen dieser Anwendungen. Regel können einzeln auf jede beliebige Property angewendet

werden. Außerdem stellt das Ontowiki Modul ein Menü zur Steuerung der Regelverwaltung. Auf Events von Ontowiki reagiert das Plugin. Es dient dabei einerseits zur Änderung der Darstellung von Inferenzen, die sich dadurch von "normalen" Daten (explizit) unterscheiden, zum anderen werden Änderungen am Ursprungsmodell überwacht, sodass diese Änderungen auf das Inferenzmodell übertragen werden. Somit wird eine fortwährende Synchronisation des Inferenz-Modells angestrebt.

Das Schichtenmodell

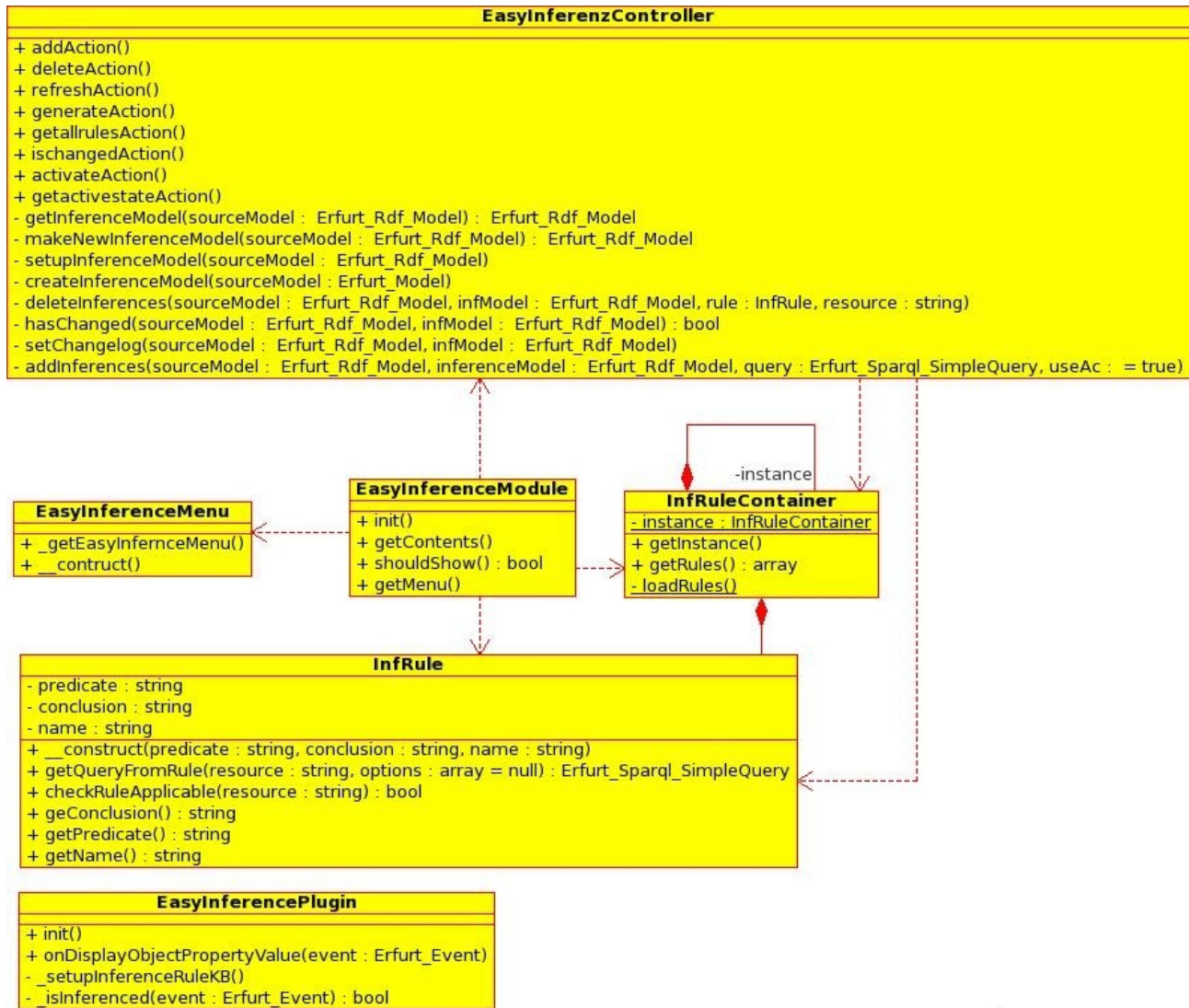


Das EasyInference-Plugin lässt sich in drei Schichten gliedern. Den Hauptteil des Plugins bildet die Anwendungsschicht, die sowohl den Ablauf steuert, als auch Verarbeitungen durchführt. Um das Plugin anpassungsfähig zu halten, werden Regeln dauerhaft gespeichert. Sie bilden eine wichtige Grundlage des Plugins. Die Kommunikation zwischen diesen beiden Schichten läuft teilweise über die Interaktionsschicht, über die Regeln ausgewählt und hinzugefügt werden können. Die Interaktionsschicht wird durch ein Regel-Modul repräsentiert.

4. Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete

Statisches Modell

UML Diagramm:



InfRule

Die Klasse InfRule repräsentiert die Regeln, durch die die Inferenzen erzeugt werden. Sie besitzt die Attribute *predicate*, *conclusion* und *name* wobei *predicate* die Bedingung ist, wann eine Regel benutzt wird und *conclusion* die Schlussfolgerung, die aus der Regel geschlossen wird, die Regeln werden mit *name* bezeichnet. Auf diese Attribute kann mit *getPredicate()*, *getConclusion()* und *getName()* zugegriffen werden. Dem Konstruktor werden diese Daten als String übergeben, woraufhin sie geparkt werden. Die Methode *getQueryFromRule()* erzeugt eine Sparql-Query mit der die inferenten Daten aus dem Modell abgefragt werden. *checkRuleApplicable* prüft, ob eine Regel auf eine Ressource angewendet werden kann.

InfRuleContainer

Die Klasse InfRuleContainer ist eine Singleton-Klasse, die sämtliche Regeln beinhaltet, die mit getRules() abgerufen werden können. Die Instanz der Klasse wird mit der statischen Methode getInstance() abgerufen. Mit Hilfe der statischen Methode loadRules() werden die Regeln geladen und in ein statisches array gespeichert.

EasyInferenzController

Diese Klasse stellt eine Komponente in OntoWiki dar. Sie beinhaltet verschiedene Actions mit denen das Inferenz-Modell manipuliert werden kann. Die addAction()-Methode fügt dem Inferenz-Model eine neue Regel hinzu. Die generateAction()-Methode generiert Inferenzen die aus einer Regel hervorgehen. Die Methode deleteAction() entfernt eine Regel und alle Inferenzen, die aus ihr hervorgegangen sind. Die refreshAction()-Methode generiert alle Inferenzen komplett neu. Sollte bei diesen Aktionen das Inferenzmodell noch nicht existieren, so wird es mit Hilfe der Methode createInferenceModel() erstellt. Mit Hilfe der getactivestateAction()- und activateAction()-Methode kann ein Inferenzmodell deaktiviert und aktiviert sowie der aktuelle Status abgefragt werden. Die getallrulesAction() dient zum Laden von den Regeln, die im Modul angezeigt werden. Dabei wird geprüft ob eine Regel überhaupt anwendbar ist.

EasyInferenceModul

Dieses Ontowiki-Modul ist das grafische User Interface der Anwendung. Hier werden die Regeln angezeigt und aktiviert/deaktiviert. Bei der Aktivierung/Deaktivierung von Regeln wird die entsprechende Action im Controller aufgerufen. Weiterhin kann im Modul ein Inferenzmodel aktiviert/deaktiviert, sowie generiert werden. Dabei werden ebenfalls entsprechende Actions im Controller aufgerufen.

EasyInferenceMenu

Diese Klasse erweitert das Moduls um ein Menu. Es dient dazu die unterschiedlichen Nutzeraktionen getrennt voneinander darzustellen, was zur Verbesserung der Benutzerfreundlichkeit und Übersichtlichkeit des Moduls führt.

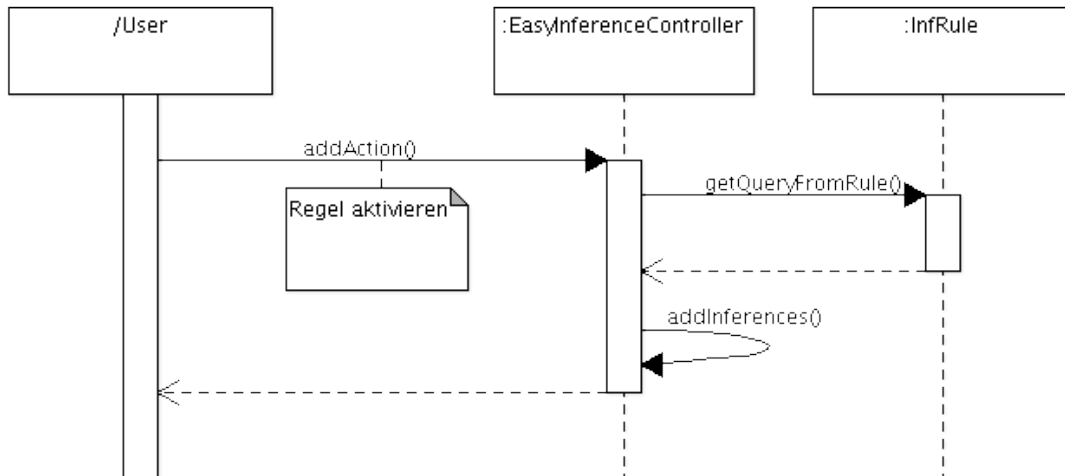
EasyInferencePlugin

EasyInferenzPlugin ist ein OntoWiki-Plugin, das die inferierten Daten hervorhebt. Ob es sich bei den anzuzeigenden Daten um Inferenzen handelt wird dabei in der Event-Methode onDisplayObjectPropertyValue geprüft. Das Plugin ist ebenfalls dafür zuständig, dass die Inferenzregel-Ontologie in Ontowiki geladen wird, falls diese dort noch nicht existiert.

Dynamisches Modell:

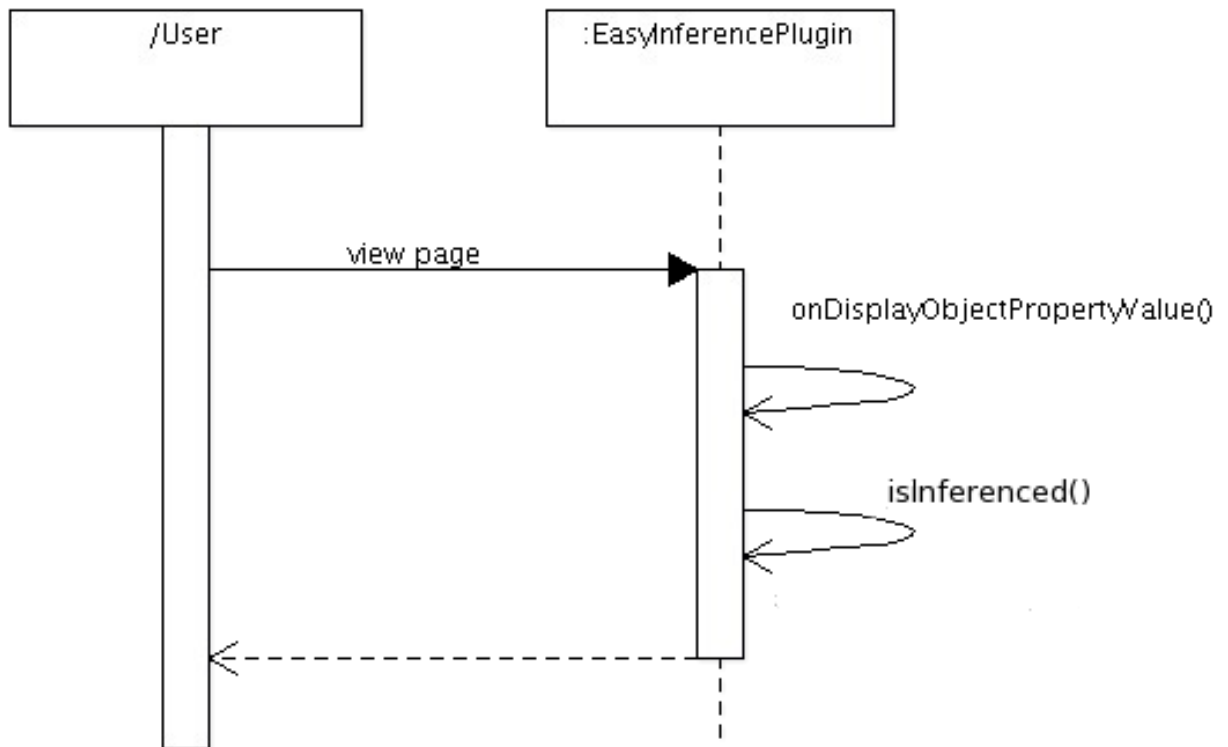
Sequenzdiagramme:

Generieren von Inferenzen



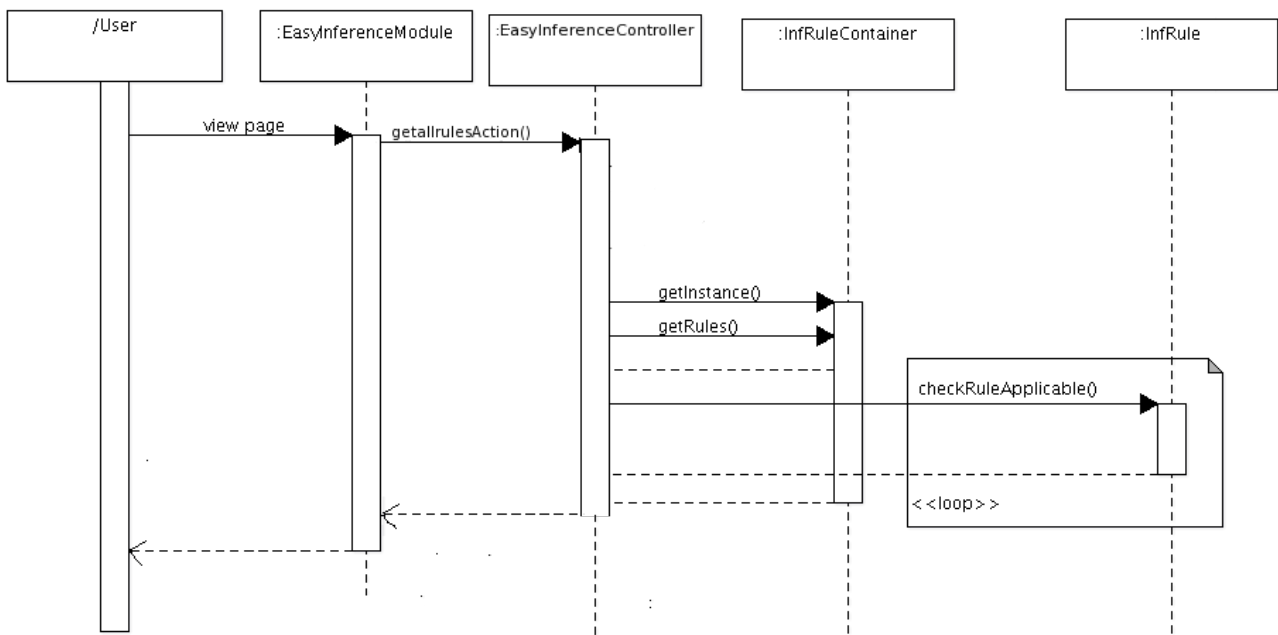
Der Benutzer hat eine Regel ausgewählt und auf den generate-Button geklickt. Es wird die `generateAction()`-Methode des `EasyInferenzController` aufgerufen, diese ruft die Methode `getQueryFromRule()` auf um eine SPARQL-Query zu bekommen. Mit Hilfe dieser Query werden in der `addInferences()`-Methode die inferierten Daten abgefragt und diese dann in das Inferenzmodell geschrieben.

Markierung der Inferenzen



Der Benutzer befindet sich auf der Ontowiki Seite und hat eine Resource ausgewählt. Durch die Anzeige der Eigenschaft dieser Resource, wird der Event `onDisplayObjectPropertyValue()` in der `EasyInferencePlugin` Klasse ausgelöst. Daraufhin markiert das PlugIn die Inferenzen, die es vorher durch die Methode `isInferenced()` prüft.

Anzeigen von Regeln in Modul-Boxen



Nachdem eine Ressource ausgewählt wurde, wird das Modul angezeigt. Dabei werden durch die `getallrulesAction()`-Methode im Controller die anzuzeigenden Regeln ermittelt. Zu jeder dieser Regeln wird `checkRuleApplicable()` aufgerufen, um zu prüfen, ob die Regel auf die Ressource anwendbar ist. Ist sie anwendbar, wird sie zum Hinzufügen angezeigt. Falls eine dieser Regeln schon aktiviert ist, wird sie zum Löschen angezeigt.