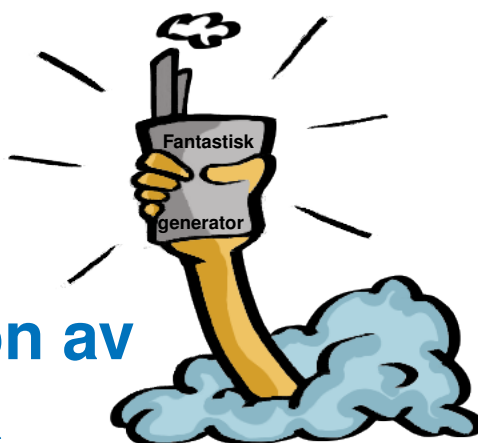


# Effektiv konstruktion av utbyggbara semantiska editorer



Av Emma Söderberg

Institutionen för datavetenskap  
Lunds universitet

Programmeringsverktyg som semantiska editorer hjälper oss att utveckla och förstå mjukvara, vilket är viktigt i ett samhälle där vi måste kunna lita på att mjukvaran runt omkring oss fungerar. Tyvärr är utvecklingen av semantiska editorer ofta en tidskrävande och komplicerad uppgift, och majoriteten av språk saknar programmeringsverktyg. Hur gör vi det enklare att konstruera och underhålla semantiska editorer?

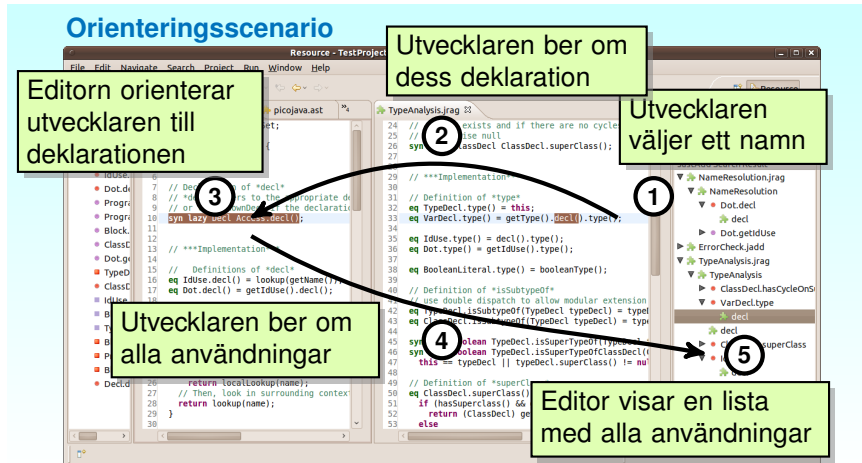
**P**rogrammeringsspråk utgör en central del av datavetenskapen, de avgör *vad* vi kan instruera en dator att göra och *hur* vi gör det. Efterhand som vi upptäcker nya bättre sätt att kommunicera med datorer utvidgar vi existerande språk eller skapar nya.

Dessutom, liksom jargong, kan vi skapa språk ämnade för en specifik domän, som till exempel HTML, ska-

pat för att konstruera websidor. Resultatet är en rik flora av språk och språkutvidgningar.

Samtidigt som programmeringsspråk hjälper oss att uttrycka lösningar på problem, hjälper programmeringsverktyg oss att utveckla lösningar, samt att förstå andras lösningar.

Tyvärr är utvecklingen av verktyg tidskrävande och komplex, och de flesta programmeringsspråk saknar verktyg.



**Figur:** Ett exempel på ett orienteringsscenario där editorn hjälper utvecklaren att förstå hur saker hänger ihop i programmet.

Om vi kan göra konstruktionen av verktyg enklare kan vi hjälpa fler mjukvaruutvecklare. Om vi dessutom kan göra det enkelt att utvidga dessa verktyg kan vi stödja språkutvidgningar.

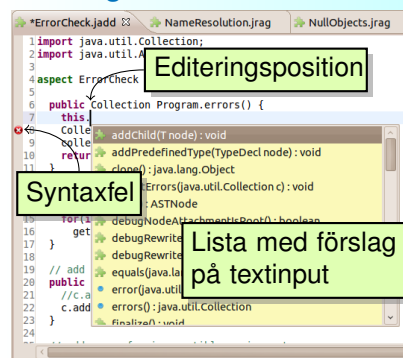
Ett annat exempel på en semantisk editortjänst är en så kallad editeringsassistent som exempelvis kan hjälpa en utvecklare att leta upp vilka namn som är giltiga att använda på en viss plats i ett program.

## Semantiska editorer

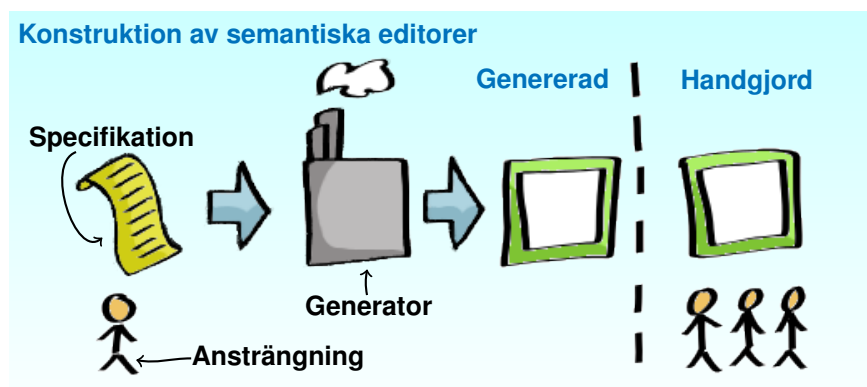
En semantisk editor är ett exempel på ett programmeringsverktyg som förstår sig på språksemantik. Denna förståelse gör det möjligt att erbjuda sofistikerade editortjänster. En semantisk editor kan exempelvis hjälpa en utvecklare att orientera sig i ett program.

Ofta i programmering definieras ett namn på ett ställe i ett program för att sedan användas på många andra ställen i samma program. En editor kan visa var ett namn är deklarerat, samt var ett namn används. Något som kan vara svårt att reda ut för hand för språk med avancerad semantik.

## Editeringsassistent



**Figur:** Ett exempel på hur en editeringsassistent kan hjälpa en utvecklare.



**Figur:** En illustration av editorgenerering till vänster, och manuell konstruktion till höger. De små gubbarna symboliserar ansträngning.

### Konstruktion av utbyggbara semantiska editorer

Ett vanligt angreppssätt för att göra mjukvaruutveckling effektivare är att generera kod, snarare än att skriva den själv, eller att återanvända kod.

För editorutveckling kan detta innebära att man försöker generera koden som beskriver hur editorn ska bete sig, eller att grafiska editorkomponenter återanvändas, eller båda.

I den här avhandlingen undersöks hur semantiska texteditorer kan genereras från formella språkbeskrivningar. Specifikt, undersöks hur semantiska editorer kan genereras från så kallad referensattributgrammatiker, en formalism för att beskriva språksemantik.

Avhandlingen redogör för hur semantiska editortjänster för orientering, alternativa programvyer, analys samt editeringsassistans kan genereras. Därtill visas hur sofistikerade editortjänster, beroende av kunskap om flödet i ett program, kan genereras och dessutom utvidgas för att hantera språkutvidgningar.

Utöver detta undersöks hur felaktig input kan hanteras i en texteditor. Detta

kan vara utmanande eftersom program under editering ofta kan bli strukturellt trasiga. Det kan då vara svårt att förstå vad ett program betyder, men man vill ändå kunna erbjuda editortjänster som behöver förstå just detta.

För att reda ut detta behöver man laga programmet, och ju bättre man kan laga det, desto bättre tjänster kan man erbjuda. I avhandlingen beskrivs en teknik för att laga strukturellt trasiga program med hjälp av formatteringsinformation som många gånger annars ignoreras vid textöversättning.

En annan viktig sak som avhandlingen undersöker är hur man effektivt kan beräkna den semantiska information som tjänsterna i en semantisk editor behöver genom att undvika att beräkna om information i onödan.

### Sammanfattning

Det är en stor utmaning att försöka göra det enklare att utveckla semantiska editorer, men bidragen i den här avhandlingen har tagit oss några viktiga steg på vägen.