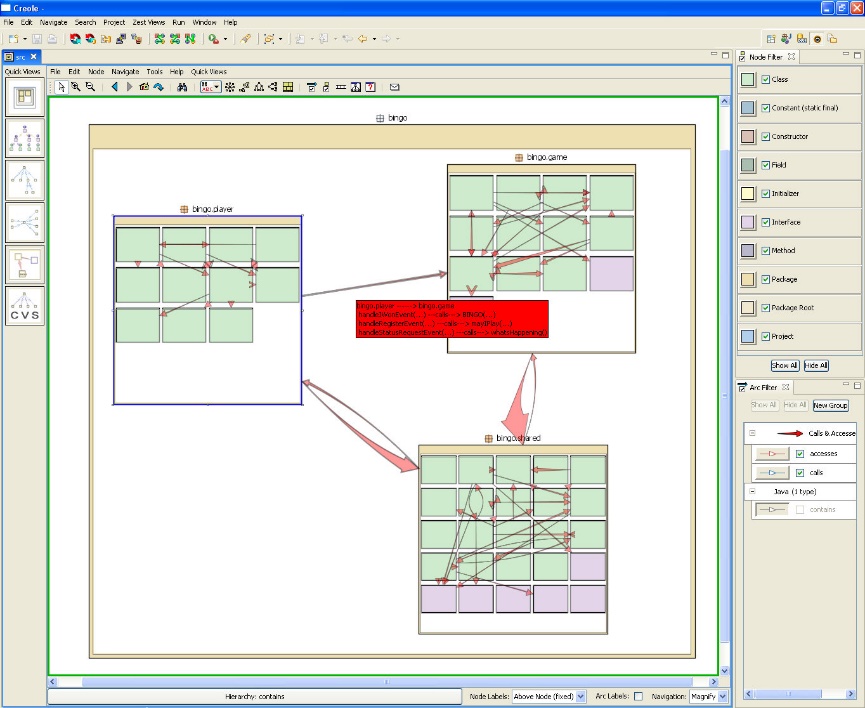
Alati za vizualizaciju programske podrške (naslov)

Alati za vizualizaciju programske podrške se baziraju na razvoju 4 glavna aspekta vizualizacije: kognitivan aspekt, aspekt izlaznih informacija (eng.output information), interakciji i računalni aspekt. Kognitivan aspekt se očituje u pristupu prema načinu ili strategiji razumijevanja koda, o izlaznim informacijama koda ovisi kompleksnost i na koji će se način najbolje moći vizualizirati kod. Interakcija korisnika s alatom omogućava razlaganje kompleksnih prikaza na jednostavnije pa svi alati u današnje vrijeme teže omogućavanju interakcije korisnika s alatom. Zadnji aspekt se odnosi direktno na alat za vizualizaciju i njegovu računalnu izvedbu. U to se ubraja sve od specifikacije sučelja poput veličine i boja pa sve do kompleksnosti alata. količine memorije i sl. Naravno, u današnje vrijeme je taj aspekt alata vrlo izražen i očekuje se što bolja izvedba u tom pogledu. [referenca->[3732\_1293.pdf (dur.ac.uk)](http://etheses.dur.ac.uk/3732/1/3732_1293.pdf?UkUDh:CyT)]

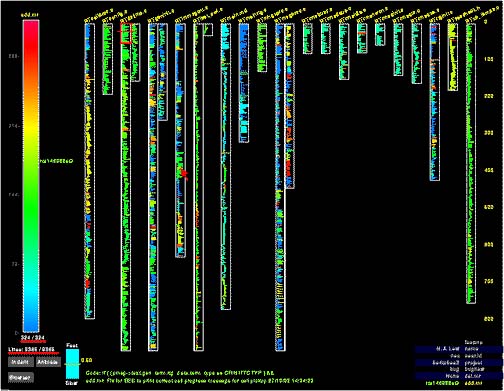
Alati bazirani na SHriMP tehnici (podnaslov)

Među prvim alatima za vizualizaciju bili su alati bazirani na prikazu grafova. Oni su imali naglasak na kognitivnom aspektu vizualizacije. No, grafički prikaz cijelog koda bi često bio jednako kompliciran kao i iščitavanje linija kodova bez ikakvog prikaza, pa se razvila tehnika pod nazivom SHriMP koja filtriranjem i grupiranjem informacija unutar grafova pojednostavlja vizualizaciju koda. Pojam je nastao kao idiom od pojmova jednostavna hijerarhijska višestruka perspektiva. Alati koji koriste ShriMP prikazuju podatke u 2D prikazu te se za bolje razumijevanje konteksta u kojem se pojavljuju određeni dijelovi koda uvodi animiranje grafova. Neki od najpoznatijih alata koji se koriste za vizualizaciju u Javi, a koji su ujedno i bazirani na ShriMP tehnici su Stand-Alone SHriMP, Creole i Jambalaya. [referenca->[(PDF) SHriMP views: an interactive environment for information visualization and navigation. (researchgate.net)](https://www.researchgate.net/publication/221516994_SHriMP_views_an_interactive_environment_for_information_visualization_and_navigation)]Zbog kompleksnosti prikaza grafova koristi se najčešće samo za vizualizaciju dijelova koda, a ne cijelog softvera ili nekog programa. Slika prikazuje isječak iz Creole alata te je preuzeta sa stranice <https://thechiselgroup.org/creole/>. [referenca->[Visualization techniques for program comprehension (drdc-rddc.gc.ca)](https://cradpdf.drdc-rddc.gc.ca/PDFS/unc70/p526037.pdf)]



SeeSoft sistem (podnaslov)

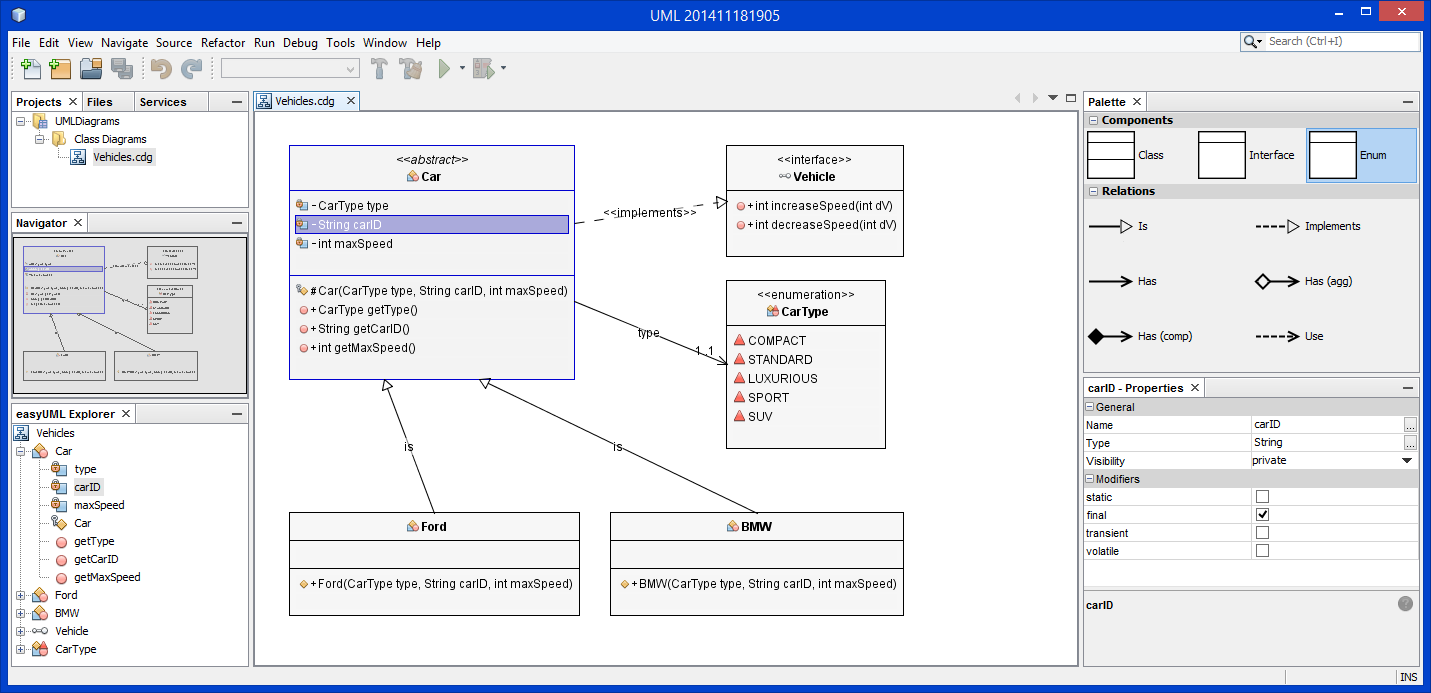
Kompleksnost prikaza pomoću grafova usmjerio je razvoj vizualizacije prema alatu SeeSoft. Ovaj sistem je nastao kako bi se omogućilo lakše vizualno praćenje statistike programske podrške, a omogućuva vizualizaciju i do 50000 linija koda. Za razliku od SHriMP sistema ne bazira se na grafovima, a aspekt koji je naglašen u ovom pristupu je interakcija korisnika s alatom. Informacije koje pruža SeeSoft alat su: autor svake linije koda, datum promijena u kodu, informaciju je li promijena unesena kako bi se ispravila greška ili da bi se dodala nova funkcionalnost te informacije o izvršavanju koda poput koliko se često izvršava, kada je zadnji put neki dio izvršen i sl. Vizualizacija koda u SeeSoft alatu se može zamisliti kao printanje cijelog koda svih datoteka te ljepljenje jednog pored drugog i zatim gledanje cijele slike iz velike daljine. Na slici je prikazan isječak iz alata preuzet sa stranice [www.researchgate.net](http://www.researchgate.net). [referenca->[Visualization techniques for program comprehension (drdc-rddc.gc.ca)](https://cradpdf.drdc-rddc.gc.ca/PDFS/unc70/p526037.pdf)]



Na vrhu su prikazana imena datoteka, a svaki stupac prikazuje kod unutar svake datoteke. Svaki red piksela grafičkog sučelja prikazuje liniju koda unutar datoteke. S desne strane označen je broj linija koda unutar datoteka. Kod je pobojan na način da su one linije koje su napisane prve (najstarije linije koda) obojane u plavu boju, a zatim se svaka novija linija poboja u boju koja je prema spektru sve bliža crvenoj boji. Alat pruža mogućnost korištenja povećala kako bi se izdovojio samo dio koda iz određene datoteke te se taj dio poveća kako bi ga se lakše moglo analizirati. Osim toga prikaže se i statistika točno tog dijela koda. Grafičko sučelje pruža i mogućnost interakcije kako bi se lakše predočile promijene unutar koda i povezanosti datoteka. Klikom na određenu boju prikažu se sve linije koda nastale u to vrijeme, a klikom na ime datoteke prikažu se sve linije koda iz drugih datoteka na koje se referencira ta datoteka. Ovim pristupom novi programeri mogu puno lakše upoznati kod koji nikad do tada nisu vidjeli, a stari programeri mogu puno brže i lakše pronaći i pratiti promijene koje se izvršavaju u programskoj podršci.[referenca->[Seesoft-a tool for visualizing line oriented software statistics - Software Engineering, IEEE Transactions on (kent.edu)](http://www.cs.kent.edu/~jmaletic/softvis/papers/eick1992.pdf)]

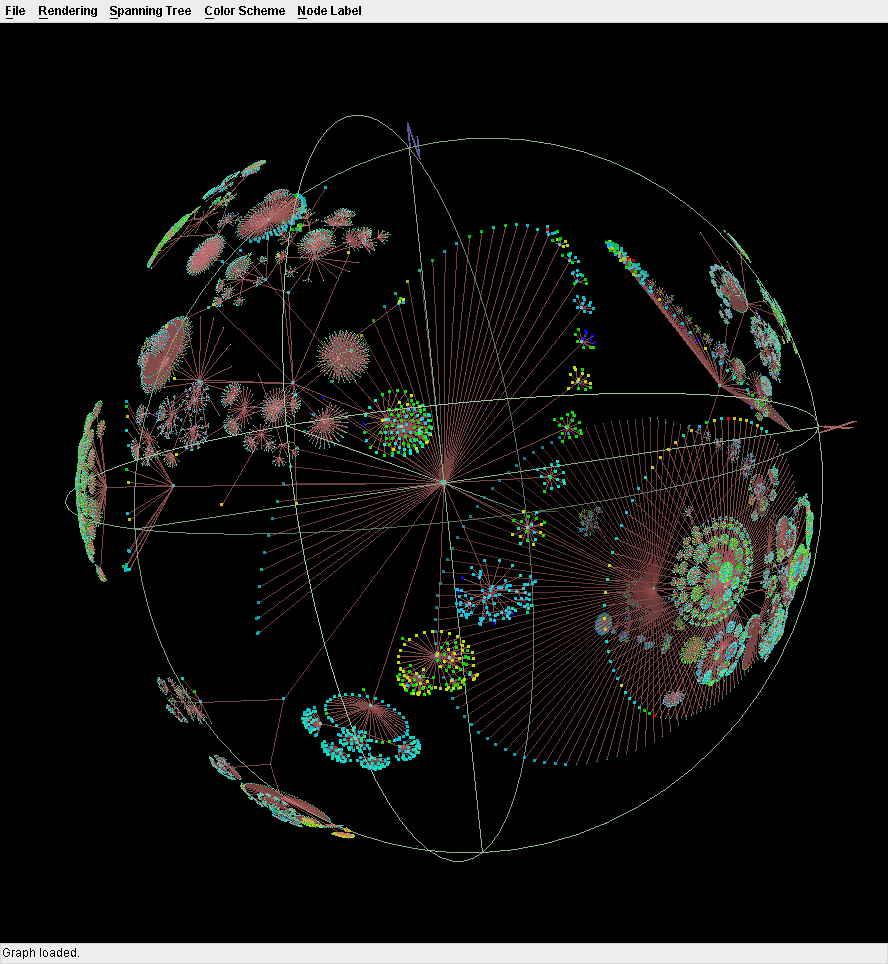
Sistem UML dijagrama (podnaslov)

Vizualizacija koda putem UML dijagrama razvijala se paralelno s grafovima, no važna je premosnica između 2D i 3D alata za vizualizaciju jer je naglasak u ovom pristupu na modeliranju i vezama između dijelova koda, a ne na samoj strukturi svakog dijela koda zasebno. Osim toga, ovakav pristup omogućava generaliziranje manjih dijelova kako bi se dobila slika cijelog sustava „odozgo“. Na taj način se riješava problem grafičkog prikaza koji nije čitljiv za prikaz cijele programske podrške odjednom. Trenutno je dostupna velika količina besplatnih alata i aplikacija za kreiranje UML dijagrama, a svi se baziraju na principu obrnutog inženjeriga. Neki od najpoznatijih alata su: VisUML, UML Designer, easyUML, [ObjectAid UML plugin](http://www.objectaid.com/), [ModelGoon](http://www.modelgoon.org/?p=494), [Omondo](http://www.uml2.org/), MoDisco i [AmaterasUML](http://amateras.sourceforge.jp/cgi-bin/fswiki_en/wiki.cgi?page=AmaterasUML). Slika prikazuje grafičko sučelje easyUML alata i preuzeta je s web stranice <http://plugins.netbeans.org/plugin/55435/easyuml>. [referenca->[UML Diagram Types | Learn About All 14 Types of UML Diagrams (creately.com)](https://creately.com/blog/diagrams/uml-diagram-types-examples/)]



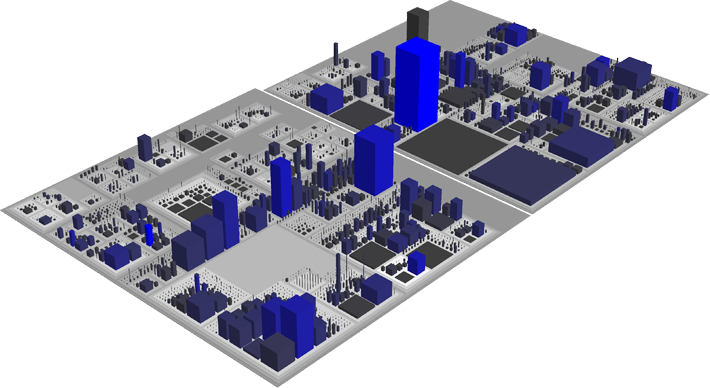
Prvi 3D alati (podnaslov)

Prvi korak prema 3D vizualizaciji koda bio je prenošenje 2D grafova u 3D sistem kako bi se dodatno omogućio istovremeni prikaz više dijelova koda te prikazivanje veza i referenca između njih. Jedan od najpoznatijih alta koji koristi ovaj sistem je Walrus. Iako u teoriji može prikazati i milijun čvorova u praksi se koristi na softverima s manje kompleksnom strukturom za koje je dovoljno nekoliko stotina tisuća čvorova za prikaz svih dijelova koda. Walrus također pruža i mogućnost interakcije s grafovim tako da je ovaj sistem u velikoj količini unaprijedio vizualizaciju programske podrške. Slika prikazuje grafičko sučelje Walrus alata i preuzeta je s web stranice <https://www.caida.org/tools/visualization/walrus/>. [referenca->[Walrus - Graph Visualization Tool (caida.org)](https://www.caida.org/tools/visualization/walrus/)]



Alati bazirani na SoftwareWorld-u (podnaslov)

Kako 2D prikaz koda nije bio dovoljno razumljiv za sve veće softvere koji su nastajali, a 3D prikaz grafova nije mogao prikazati kompleksne softvere, razvila se nova tehnika 3D vizulaizacije kodova pod nazivom SoftwareWorld. Baza ovakvih alata su virtualna okruženja u kojima je programska podrška prikazana kao građevine unutar gradova. Ovakav pristup kodu se smatra prirodnijim za ljudsko poimanje svijeta pa se na taj način razumijevanje koda približava ljudskoj intuiciji. Aspekt koji se najviše ističe u ovakvim alatima je računalna strana samih alata. SoftwareWorld se sastoji od nekoliko razina vizualizacije, a to su: svijet (cijela programska podrška), država (paketi), grad (datoteke), kvartovi (klase), građevina (konkretne metode ili funkcije unutar klasa). Primjeri prvih takvih alata su CodeCity i EvoSpace, a oni se uglavnom koriste za vizualizaciju objektno orijentiranih jezika baš zbog navedenih razina vizualizacije.[referenca->[3732\_1293.pdf (dur.ac.uk)](http://etheses.dur.ac.uk/3732/1/3732_1293.pdf?UkUDh:CyT)] Ovakav pristup po prvi put pruža mogućnost prikaza cijelog koda bez ograničenja u broju linija, a da se razumljivost prikaza ne narušava količinom koda. 2011. godine izvršen je eksperiment u kojem se alat CodeCity pokazao 24% točniji u zadacima obrnutog inženjeringa te mu je trebalo 12% manje vremena da izvrši vizualizaciju nego kada se koriste pluginovi bazirani na grafovima u Eclipse-u ili proračunske tablice s metričkim podacima. Tako da se pokazalo da veća grafička kompleksnost alata nije usporila proces vizualizacije, čak suprotno, a osim toga naglasak na grafici nije pogoršao izvedbu alata, već je i taj element napredovao. Slika prikazuje vizualizaciju koda u alatu CodeCity, a preuzeta je sa stranice <https://wettel.github.io/codecity.html>. [referenca->[CodeCity (wettel.github.io)](https://wettel.github.io/codecity.html)]



2013. godine dva profesora s Mađraskog sveučilišta [Gergő Balogh](https://ieeexplore.ieee.org/author/37078417000) i [Árpád Beszédes](https://ieeexplore.ieee.org/author/37297785500) primijenili su SoftwareWorld vizualizaciju na popularnu igricu MineCraft što je detaljno opisano u članku [[tp-balogh-gergo-codemetropolisvisualisation.pdf (u-szeged.hu)](http://www.inf.u-szeged.hu/~beszedes/research/tp-balogh-gergo-codemetropolisvisualisation.pdf)]. Smatrali su da su značajke igrice poput visoke rezolucije, detaljne teksture i mogućnost široke 3D scene upravo ono što nedostaje dosadašnjim alatima za vizualizaciju. Kako bi programsku podršku prikazivali u MineCraft-u stvorili su konverzacijski alat koji procesira izvorni kod i pomoću prevoditelja kreira građevine, kvartove i vrtove. Iako je prevoditelj nastao samo kao prototip, ideje primjenjene u ovom radu doprinjele su razvoju novih ideja po pitanju napretka grafike ovakvih alata i još više unaprijedile jednostavnost i razumljivost vizualizacije. Slika prikazuje vizualizaciju klase u Minecraft-u, a preuzeto je iz gore navedenog članka. [referenca->[tp-balogh-gergo-codemetropolisvisualisation.pdf (u-szeged.hu)](http://www.inf.u-szeged.hu/~beszedes/research/tp-balogh-gergo-codemetropolisvisualisation.pdf)]

