

ABAbuilder

ABAbuilder manuál

Editor diagramov

Nástroj ABAbuilder poskytuje možnosť grafickej tvorby simulačných modelov a generovanie kódu. Je možné vytvoriť tri typy diagramov pre popísanie simulačného modelu na rôznej úrovni detailu: diagram hierarchie agentov, diagram komponentov agenta a diagram ABAGrafu. Všetky typy diagramov poskytujú nasledovné funkcie:

Panel s komponentami a správami

V ľavej časti aplikácie sú panely s prvkami simulačného modelu, ktoré je možné pridať do diagramov. Panely sú dva: v prvom sú komponenty (agenti, asistenti, uzly Petriho siete) a v druhom hrany (správy, hrany petriho siete). Panely obsahujú rôzne prvky v závislosti od aktuálneho typu diagramu.

Panel s atribútmi

Po označení prvku je možné upravovať jeho atribúty v paneli atribútov v pravej časti aplikácie. Niektoré položky sú len informačné (typ komponentu) a nie je možné ich meniť.

Pridanie nového komponentu

Požadovaný komponent sa označí v paneli s komponentami a stlačením myši na prázdne miesto diagramu je vytvorený nový komponent. Po vybraní komponentu v paneli je ich možné do diagramu pridávať ľubovoľný počet, až kým nie je zrušený výber prvku v paneli - stlačením na prvok diagramu, kliknutím pravým tlačidlom myši alebo výberom prvku v paneli s komponentami.

Pridanie novej hrany

Požadovaná hrana je vybraná v paneli s hranami a vytvorená ťahaním myši medzi zdrojovým uzlom a cieľovým. Hrana je vytvorená len v prípade, že je možné dané uzly prepojiť.

Selekcia

Prvky diagramu je možné označiť a potom ďalej upravovať.

Uzly diagramu sú označené kliknutím myšou a hrany je možné označiť stlačením na ich názov alebo šípku.

Je možné vybrať niekoľko prvkov súčasne stlačením na prázdne miesto diagramu a ťahaním myši alebo postupným vyberaním komponentov s podržaním tlačidla ctrl.

Stlačením ctrl+A je možné vybrať všetky komponenty.

Odstránenie prvku

Označený prvok je odstránený stlačením jedného z tlačidiel delete alebo backspace. Je možné odstrániť viacero prvkov súčasne.

Kopírovanie prvkov

Označené uzly diagramu je možné skopírovať stlačením ctrl+C a ctrl+V.

Zmena uzla hrany

V prípade potreby zmeny koncového uzla hrany je možné hranu z daného uzla odpojiť ťahaním jej šípky. Pustením tlačidla myši nad uzlom, ktorý môže byť hranou prepojený, je hrane pridaný nový koncový uzol. V prípade, že je tlačidlo myši pustené mimo uzla alebo nad uzlom, ktorý nemôže byť prepojený danou hranou, je ako koncový uzol hrany nastavený pôvodný uzol.

Diagram hierarchie agentov

Diagram hierarchie agentov popisuje simulačný model na najvyššej úrovni. Definuje jednotlivých agentov, ich hierarchiu a správy, ktoré si posielajú. Agent môže byť dvoch typov: riadiaci a dynamický. To akého je agent typu determinuje, aké správy vie odoslať a prijať.

Diagram sa vždy nachádza v prvej záložke, ktorá jediná nemôže byť zatvorená. Tento diagram je otvorený pri spustení programu. Ostatné diagramy sú otvorené dvojklikom na

jednotlivých agentov pričom typ diagramu, ktorý je otvorený závisí od nastavenia atribútov agenta (konkrétne hodnoty atribútu "Use ABAGraph").

Agenti

Diagram hierarchie agentov poskytuje dva typy agentov:

Riadiaci agenti tvoria pevnú štruktúru, ktorá je definovaná pri spustení simulácie a počas behu simulácie sa už nemení. Riadiaci agent modeluje špecifickú časť systému, pričom môže modelovanie určitých častí delegovať podriadeným agentom. V diagrame je reprezentovaný elipsou zelenej farby.

Dynamickí agenti reprezentujú entity simulačného modelu, ktoré sa vedú do istej miery sami rozhodovať a riešiť problémy. V diagrame je reprezentovaný elipsou modrej farby.



Hrany, ktoré spájajú agentov môžu byť dvoch typov: správy a vzťahy.

Parent je hrana, ktorá špecifikuje vzťah nadradeného agenta a tým definuje hierarchickú štruktúru simulačného modelu. Je jedinou hranou diagramu, ktorá nie je správou. Je niekoľko obmedzení spôsobu, akým môžu byť agenti prepojení hranou parent. Nástroj používateľovi nedovolí prepojiť agentov hranou parent neprípustným spôsobom. Agenti tvoria strom (acyklický graf). Agent na vrchole hierarchie, nazývaný boss, jediný nemá predka. Pri generovaní kódu je v prípade, že používateľ nedefinoval hierarchiu s jedným agentom bez predka, zobrazené upozornenie. Nástroj nedovolí spojiť dvoch agentov vzťahom typu parent, ak by bola porušená jedna z podmienok:

- agent má najviac jedného predka
- nemôže existovať orientovaný cyklus z hrán typu parent

Správy

Správy sú spôsobom, akým spolu agenti komunikujú. Je niekoľko typov správ, ktoré si medzi sebou môžu agenti posilať, každý pre špecifický účel.



Notice (oznámenie) je základným typom správy, na ktorú sa neočakáva žiadna odpoveď. Agenti ju používajú na informovanie iných agentov o vzniknutej situácii, prípadne ich žiadajú o vykonanie činností, o ktorých uskutočnení nepotrebujú byť informovaní.

Request (žiadosť) je správa obsahujúca určitú požiadavku, pričom odosielateľ očakáva na ňu od adresáta odpoveď vo forme správy typu Response.

Response (odozva) je správa, ktorá reprezentuje odpoveď na správu typu Request, pričom môže byť doručená iba jej odosielateľovi.

Call (telefonát) predstavuje možnosť okamžitej komunikácie medzi agentami. Správa je prijímačom agentovi doručená okamžite, mimo štandardného postupu doručovania správ.

Goal (cieľ) je správa, ktorú využívajú riadiaci agenti na priradenie cieľov dynamickým agentom. Dynamickí agenti sú povinní plniť priradené ciele, no môžu sledovať aj vlastné lokálne ciele, ak tieto nie sú v rozpore s cieľom priradeným riadiacim agentom. Po splnení cieľa dynamický agent informuje o tejto skutočnosti riadiaceho agenta zaslaním správy typu Done.

Done (hotovo) je správa, ktorú zasiela dynamický agent svojmu riadiacemu agentovi v okamihu splnenia priradeného cieľa. Riadiaci agent na túto správu zvyčajne reaguje zadaním nového cieľa danému dynamickému agentovi (zaslaním správy typu Goal).

Transfer (presun) je správa, ktorú dynamický agent zasiela riadiacemu agentovi, aby mu signalizoval, že sa práve snaží opustiť oblasť jeho pôsobnosti. Riadiaci agent po prijatí tejto správy zvyčajne identifikuje riadiaceho agenta, ktorý je zodpovedný za oblasť, do ktorej dynamický agent vstupuje a odovzdá mu ho.

Cancel (zrušenie) je správa, ktorú riadiaci agent využíva na zrušenie v minulosti zadaných cieľov dynamickým agentom.

Handover (odovzdanie) je správa, ktorú využíva riadiaci agent na presunutie dynamického agenta do pôsobnosti iného riadiaceho agenta. Prijímajúci riadiaci agent sa stáva novým riadiacim agentom daného dynamického agenta.

Entrust (zverenie) je správa, ktorou riadiaci agent dočasne odovzdáva spravovanie daného dynamického agenta inému riadiacemu agentovi. Očakáva sa, že dynamický agent po vykonaní definovaných úkonov opätovne prejde do pôsobnosti odosielaťa tejto správy. Dočasné odovzdávanie dynamických agentov je možné vykonávať aj viacnásobne.

Return (návrat) je správa využívaná pri návrate spravovania dynamického agenta pôvodnému riadiacemu agentovi, ktorý spravovanie tohto dynamického agenta dočasne odovzdal inému riadiacemu agentovi správou typu Entrust.

Typy správy, ktoré môžu byť odoslané, sú obmedzené typom agenta adresáta a agenta odosielaťa.

- Riadiaci agent môže poslať inému riadiacemu agentovi správy typu call, entrust, return, handover, notice, request a response.
- Riadiaci agent môže poslať dynamickému agentovi správy typu call, cancel, goal a notice.
- Dynamický agent môže poslať riadiacemu agentovi správy typu call, done, notice a transfer.
- Dynamický agent môže poslať dynamickému agentovi správy typu call a notice.

Nástroj nedovolí spojiť dvoch agentov správou, ktorá nie je validná.

Jedným z atribútov agenta, ktorý môže tvorca modelu nastaviť v paneli s atribútami, je atribút "Use ABAGraph", ktorý vyjadruje, či je správanie daného agenta popísané ABAGrafom. Je dvojhodnotový, preto je preň v paneli použitý checkbox. Od hodnoty tohto atribútu okrem iného závisí aj to, ktorý diagram sa zobrazí po dvojkliku na agenta (diagram komponentov agenta alebo diagram ABAGrafu).

Medzi dvoma komponentami môže byť správ pomerne veľa, preto je pre prehľadnosť možné skryť správy posielané medzi dvoma komponentami dvojklikom na jednu z hrán spájajúcich tieto komponenty. Dvojklikom na hranu "Multiple" sa opäť rozbalia.

Diagram komponentov agenta

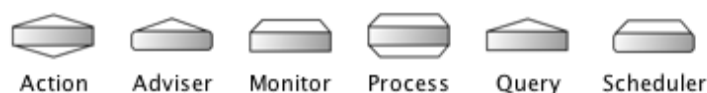
Diagram modeluje vnútornú štruktúru agenta. Je otvorený dvojklikom na agenta ktorý nie je ovládaný ABAGrafom. Pre zatvorenie diagramu je možné kliknúť mimo agenta alebo urobiť dvojklik na prázdne miesto alebo stlačiť tlačidlo esc.

Tento diagram sa skladá z komponentov agenta (manažér a asistenti) a správ, ktoré si posielajú. Asistenti komunikujú výhradne s manažérom. Nemajú dovolené posielat si správy navzájom, ani komunikovať s inými agentami. Nástroj ABAbuilder preto nedovolí vytvorenie hrany medzi dvoma asistentami.

Manažér (reasoner) je v diagrame vytvorený implicitne pri vytvorení agenta. Používateľ ho nemôže zmazať ani vytvoriť ďalšieho.)

Asistenti

Asistenti sú dvoch typov: kontinuálny a okamžitý. Sú od seba graficky rozlíšený tým, že horná (pri procese a akcii aj dolná) hrana kontinuálnych asistentov je plochá, zatiaľ čo pri okamžitých asistentoch tvorí trojuholník.



Proces (process) je kontinuálnym asistentom, ktorý vykonáva rozhodnutia manažéra o zmene stavu systému. Po spustení autonómne mení hodnoty stavových premenných.

Plánovač (scheduler) je kontinuálnym asistentom, ktorý podporuje rozhodovanie manažéra tým, že mu poskytuje návrhy riešenia problémov. Plánovač vytvára plány riešenia problémov, ktoré sú založené na predikcii vzniku situácií vyžadujúcich rozhodnutie. V zvolených časových okamihoch vykonáva svoju funkciu aj bez iniciatívy manažéra.

Monitor (monitor) je kontinuálnym asistentom, ktorý sprístupňuje informácie o stave prostredia. Vykonáva dlhodobé snímanie istej časti prostredia a sprostredkuje manažérovi tie informácie, ktoré sú pre neho významné.

Akcia (action) je okamžitým asistentom, ktorý vykonáva rozhodnutia manažéra o zmene stavu systému. Zabezpečuje okamžitú zmenu stavu.

Poradca (adviser) je okamžitým asistentom, ktorý podporuje rozhodovanie manažéra tým, že mu poskytuje návrhy riešenia problémov. Poradca je pasívny komponent, ktorý iba na pokyn manažéra okamžite navrhne spôsob riešenia problému (optimalizačný algoritmus).

Dotaz (query) je okamžitým asistentom, ktorý sprístupňuje informácie o stave prostredia. Dotaz sprostredkuje informáciu na pokyn manažéra okamžite.

Pri vytvorení kontinuálneho asistenta sú medzi ním a manažérom automaticky vytvorené správy start a finish a pri vytvorení okamžitého asistenta je medzi ním a manažérom automaticky vytvorená správa execute - jediná správa, ktorú vie okamžitý asistent spracovať.

Správy

Podobne ako v diagrame hierarchie agentov môžu byť správy posielané v rámci agenta niekoľkých typov.



Start (štart) je správa, po ktorej prijatí začne kontinuálny asistent, ktorý je jej adresátom, svoju činnosť.

Break (zrušenie) je správa, ktorou môže manažér ovplyvniť beh kontinuálneho asistenta, ktorý je po prijatí tejto správy ukončený. Tento typ správy sa používa, keď už nemá zmysel, aby kontinuálny asistent dokončil svoju činnosť.

Execute (vykonaj) je typom správy určenej výhradne pre okamžitého asistenta, ktorý okamžite vykoná zodpovedajúcu činnosť (identifikovanú kódom a parametrami správy) a okamžite správu vráti manažérovi s vyznačeným výsledkom svojej činnosti. Správy tohto typu obchádzajú mechanizmus doručovania prostredníctvom poštovej schránky agenta a sú doručované priamo a okamžite.

Finish (koniec) je správa, ktorú každý z kontinuálnych asistentov povinne pošle svojmu manažérovi v okamihu ukončenia svojej činnosti.

Notice (oznámenie) sú správy, ktoré môžu byť posielané kedykoľvek v priebehu činnosti kontinuálneho asistenta. Asistent ich využíva na informovanie manažéra o dôležitých skutočnostiach vyžadujúcich jeho reakciu.

Hold (zdrž) je jedinou správou, ktorá môže mať požadovaný čas doručenia rôzny od aktuálneho simulačného času. Pomocou tohto typu správy je v architektúre ABAsim realizovaný posun simulačného času. Správu zasiela kontinuálny asistent vždy iba sám sebe.

Diagram ABAGrafu

ABAGraf je modifikovaná Petriho sieť popisujúca správanie agenta.

Diagram je možné otvoriť v diagrame komponentov agenta dvojklikom na agenta ak je nastavené, že agent používa na popis svojho správania ABAGraf. Pre diagram je vytvorená záložka.

Správanie agenta je definované spôsobom, akým reaguje na správy. Množina správ, ktoré prijíma ABAGraf je daná atribútmi hrán vychádzajúcich zo vstupných miest ABAGrafu. Môže sa jednať o tie isté správy, ktoré boli definované v diagrame hierarchie agentov a v takom prípade pri generovaní kódu s nimi musí nástroj pracovať, ako by sa jednalo o tú istú hranu. Toto je realizované párovaním hrán podľa ich názvu. Okrem toho je používateľ vizuálne notifikovaný o tom, že došlo k spárovaniu tým, že z tabuľky s atribútmi zmiznú niektoré riadky, ktoré už nemajú zmysel pre spárovanú hranu (id správy, ktoré je už nastavené v diagrame hierarchie agentov).

Nie je vyžadované, aby boli všetky správy spárované. To umožňuje, aby boli správy definované v kóde a súčasne spracované interpreterom ABAGrafu. Ak správa existuje aj v diagrame hierarchie agentov aj v ABAGrafe, tak sú príslušné hrany spárované a nástroj zohľadňuje to, že hrany reprezentujú tú istú správu.

V aktuálnej verzii nástroja je pri práci so spárovanými hranami jedno obmedzenie. Hrany musia byť definované v správnom poradí - najskôr v diagrame hierarchie agentov, až potom v ABAGrafe.

Nástroj ABABuilder poskytuje možnosť vybrať si, či konkrétny agent používa ABAGraf alebo nie. Ak nie, je potrebné napísať implementáciu manažéra vo forme kódu, v opačnom prípade je správanie manažéra definované ABAGrafom.

ABAGraf je zložený z niekoľkých typov uzlov a hrán. Základné rozdelenie uzlov je na miesta a prechody. Sú dva typy prechodov: jednoduchý a rozhodovací. Miesta môžu byť rozdelené na externé a interné. Úlohou prechodu je presun správ medzi miestami. Prechod má ľubovoľný počet vstupných a výstupných hrán. Po splnení podmienok je prechod aktivovaný a správy sú presunuté zo vstupných miest na výstupné. Výšku prechodu je možné upraviť v paneli s atribútmi, ale pre urýchlenie práce je tiež možné zväčšiť výšku prechodu na dvojnásobok dvojklikom myši a zmenšiť výšku prechodu na polovicu dvojklikom myši a podržaním tlačidla alt.

Jednoduchý prechod je prechod, ktorý je aktivovaný, keď je na jeho vstupných miestach požadovaný počet hrán. Je preto potrebný mechanizmus na určenie smerovania správ. Toto je realizované tým, že na výstupnej hrane je možné definovať podmienku, ktorá definuje spôsob poslania správy. Jedna správa môže byť poslaná aj po viacerých hranách, preto sú možné dva spôsoby jej zaslania: preposlanie originálu správy a poslanie jej kópie, čo je tiež definované v podmienke hrany. Podmienka sa skladá z dvoch častí: spôsobu poslania správy a identifikátora zdroja, z ktorého správa prichádza. Napríklad podmienka pre poslanie správy, ktorá prišla z prechodu s identifikátorom 9 je v prípade, že sa odosiela originál "=9" a v prípade, že sa odosiela kópia "+9". Pokiaľ nezáleží na tom, ktorá značka je preposlaná, je tiež možné namiesto identifikátora zdrojového prechodu do podmienky napísať hviezdičku, teda "=*" alebo "+*". Implicitná hodnota podmienky je "=*".

V diagrame je reprezentovaný čiernym obdĺžnikom. Pod ním je identifikátor prechodu (napríklad: t1, t2, ...) a nad ním je počet správ požadovaných pre aktiváciu prechodu (v prípade, že je potrebných viac než jedna správa, inak nie je počet zobrazený).

Rozhodovací prechod je prechod, ktorým je implementované vetvenie. Výstupné miesto, na ktoré je správa poslaná je vybraté podľa podmienky, ktorá je vyhodnotená podľa hodnoty atribútov správy. Správy ABASim majú niekoľko preddefinovaných atribútov, ktoré používa rozhodovací prechod. Jedným z jeho parametrov je parameter správy, podľa ktorého má rozhodovať. V diagrame je reprezentovaný čiernym obdĺžnikom s menším bielym obdĺžnikom na ľavo od čierneho. Nad prechodom je rozhodovací parameter a pod ním jeho identifikátor. Prechod typu code je na rozdiel od ostatných prechodov zaujímavý tým, že jednotlivé vstupné hrany tvoria páry s výstupnými hranami.

Pre uľahčenie práce je pre prechod typu code implementovaných niekoľko funkcií. Pri pridávaní novej hrany nástroj vo väčšine prípadov nedovolí spojiť dve miesta hrán. Výnimkou je, ak je zdrojom hrany externé miesto. Namiesto toho, aby nástroj vypísal správu o tom, že takéto prepojenie nie je povolené, prepojí externé miesto s prechodom typu code jednou hranou a súčasne vytvorí druhú hranu, ktorou prepojí prechod s cieľovým uzlom alebo prechodom.

Pri vyplňovaní niektorých atribútov jednej z hrán prechodu typu code je súčasne nastavený iný atribút spárovanej hrany na opačnej strane prechodu, ktoré vyplýva z nastavenia prvej hrany.

Pri odstraňovaní jednej z hrán prechodu typu code je odstránená aj s ňou spárovaná hrana na opačnej strane prechodu.

Pri automatickom usporiadaní hrán sa predchádza prekrižovaniu hrán len na strane zdroja. Hrany na opačnej strane prechodu zostávajú spárované a je teda na používateľovi, aby vhodným usporiadaním cieľových komponentov zabezpečil dobrú čitateľnosť diagramu.

Interné miesto reprezentuje okamžitých asistentov agenta. Okamžitý asistenti majú metódu execute, ktorej parametrom je správa. Pri vykonaní execute môžu nastavovať parametre správy, ktoré sú používané rozhodovacími prechodmi. Musí mať práve jednu vstupnú a práve jednu výstupnú hranu.

Externé miesto môže reprezentovať agenta, kontinuálneho asistenta alebo absorbciu. Nemôže mať súčasne vstupné a výstupné hrany, ale ich počet je ľubovoľný. Reprezentujú

miesta, na ktorých vznikajú a zanikajú značky Petriho siete. Vznik značky znamená doručenie správy komponentu, na ktorom táto značka vznikla. Zánik značky znamená odoslanie správy komponentom, v ktorom značka zanikla.

Absorbcia je špeciálnym externým výstupným miestom ABAGrafu, ktoré nereprezentuje komponent simulačného modelu. Správy doručené absorbcii nie sú poslané ďalej. Jej úlohou je dealokácia správ.

Referencie komponentov

Každé miesto ABAGrafu môže byť z dvoch typov: komponent alebo referencia na komponent. Pri editácii diagramu sa s obidvoma typmi komponentov pracuje rovnako, rozdiel je len pri ich vytváraní a pri generovaní kódu.

Všetci agenti v ABAGrafe sú referenciami na komponenty v diagrame hierarchie agentov. Je ich preto pre použitie v ABAGrafe potrebné najskôr vytvoriť v diagrame hierarchie agentov.

Referencia môže byť vytvorená dvomi spôsobmi:

- vybratím originálu komponentu a stlačením tlačidla "Make reference" v paneli s atribútmi,
- vybratím typu komponentu "Reference" v paneli s komponentami a následným výberom originálu komponentu v Comboboxe v paneli s atribútmi.

Graficky sú v diagrame referencie asistentov agenta odlišené znakom "R" v pravom dolnom rohu. Pre agentov nemá zmysel rozlišovať, či sa jedná o referenciu, keďže ABAGraf neobsahuje originály agentov, preto sú zobrazení rovnako ako v diagrame hierarchie agentov. Pre uľachčenie orientácie v diagrame je po dvojkliku myšou na referenciu asistenta zvýraznený jeho originál (v prípade, že originál nie je aktuálne zobrazený, urobí sa scroll plátna).

Generovanie kódu

Generovanie kódu je spustené z menu "Run->GenerateCode...". Zobrazí sa okno, kde je možné vybrať programovací jazyk a cestu k priečinku, do ktorého je kód generovaný. Cesta

je implicitne vyplnená na cestu, na ktorej je projekt uložený. Z tohto dôvodu je odporúčané uložiť projekt ABABuilderu do priečinku zo zdrojovým kódom použitého IDE (src dir).

Kód je možné generovať opakovane. Postup práce môže byť nasledovný:

1. Vytvoriť diagram
2. Generovať kód
3. Modifikovať kód
4. Upraviť Diagram
5. Goto krok 2

Generátor kódu teda upraví kód, ktorý mohol byť modifikovaný používateľom o sekcie, ktoré boli zmenené v diagrame bez toho, aby boli poškodené časti, ktoré doň pridal používateľ. Je ale niekoľko obmedzení na zmeny, ktoré môže používateľ v kóde urobiť. Generátor kódu rozoznáva tri typy kódu: nemodifikovateľný, modifikovateľný a používateľský. Toto rozlíšenie je dané metadátami v komentároch, ktoré začínajú reťazcom "meta!" nasledovaným parametrami.

Nemodifikovateľný kód by nemal byť menený používateľom. V prípade pregenerovania súboru bude všetko, čo používateľ do tejto sekcie pridal, prepísané novou verziou nemodifikovateľného kódu. Tvorí blok, ktorý začína parametrom metadát tag="begin" a končí tag="end".

Modifikovateľný kód môže používateľ zmeniť, ale len obmedzeným spôsobom. Používateľ by nemal meniť hlavičky metód označených ako modifikovateľné, ale môže doplniť telo metód. Tieto obmedzenia vyplývajú z toho, že je potrebné nejakým spôsobom prepojiť nemodifikovateľný kód s používateľským. Hlavičky metód modifikovateľného kódu tvoria rozhranie pre nemodifikovateľný kód, ktorý tieto metódy používa.

Kód je označený ako modifikovateľný prostredníctvom metadát v komentári nad hlavičkou metódy alebo triedy.

Samotná trieda je tiež modifikovateľným kódom, to znamená, že používateľ môže upravovať jej telo ľubovoľne, ale úprava hlavičky triedy je obmedzená tým, že používateľ by nemal zmeniť názov triedy a nadtriedu triedy, pretože táto časť je určená modelom a keby súbor

pregeneroval, tak by boli tieto zmeny prepísané hodnotami modelu. Môže však ľubovoľne pridávať rozhrania. Pre premenovanie triedy je potrebné premenovať príslušný komponent v diagrame.

Používateľský kód je všetko ostatné, čo používateľ nástroja ABAbuilder doplní do vygenerovaného kódu mimo sekcií označených ako modifikovateľný a nemodifikovateľný kód. Po pregenerovaní súboru tento kód zostáva nezmenený.

Zoznam skratiek

IDE

ctrl+N	Vytvorí nový projekt
ctrl+S	Uloží projekt
ctrl+O	Otvorí existujúci projekt

Editor

ctrl+A	Označí všetky komponenty diagramu
ctrl+C	Skopíruje označené komponenty
ctrl+V	Vloží skopírované komponenty do diagramu
double-click na správu	Skryje/rozbalí správy pre zvýšenie prehľadnosti (ak sú aspoň dve)
ťahanie šípky hrany	Zmena adresáta

ABAGraf

double-click na prechod	Zdvojnásobí výšku prechodu
alt+double-click na prechod	Nastaví výšku prechodu na polovicu
prepojenie vstupného a interného miesta	Vytvorí dve hrany - jednu medzi vstupným miestom a "Code" prechodom a druhú medzi "Code" prechodom a interným miestom
double-click na referenciu	Zobrazí originál