



Er LabVIEW forritunarmál til almennra nota?

Oft heyrir ég fjörugar umræður um það, hvort LabVIEW sé forritunarmál til almennra nota, eða hvort það sé einungis sérhæft forritunarkerfi fyrir mælingar og sjálfvirkni. Skipast þátttakendur gjarnan í tvær andstæðar fylkingar. Þeir sem ekki eru LabVIEW-sinnar benda á að ýmsar aðgerðir í vinsælum forritunarmálum vantar í LabVIEW, en LabVIEW-sinnar benda hins vegar á hinar flóknustu lausnir sínar til almennra nota, lausnir þar sem mælingar og sjálfvirkni koma alls ekki við sögu.

Skoðanakönnun meðal LabVIEW-notanda myndi trúlega gefa svipaða niðurstöðu og skoðanakönnun sem nýlega var gerð meðal forritara okkar, þar sem langflestir voru þeirrar skoðunar að LabVIEW hefði nægar aðgerðir til að teljast forritunarmál til almennra nota, enda notuðu þeir LabVIEW á þann hátt. Það sem helst taldist vera frábrugðið skipti í reynd litlu máli í þeim almennu lausnum sem gerðar voru.

Röng spurning

Hvað sem öllum skoðanakönnunum líður, þá tel ég spurninguna ranga og sá sem reynir að svara hennir getur auðveldlega lent í ógöngum. Mér þykir spurningin álíka út í hött og ef spurt væri hvort bill sé staður til að sitja í. Auðvitað er hægt að sitja í bíl en ef það er það eina sem gert er við hann, þá er lítill tilgangur í að eiga bílinn. Betra væri að spyrja: Er hægt að nota LabVIEW sem forritunarmál til almennra nota? Enn betra væri að spyrja: Er hægt að nota LabVIEW til að smíða almennar lausnir?

Nýja spurningin er ekki kýrskír því það er ekki endilega alveg ljóst hvað teljast almennar lausnir, en hún beinir athyglinni frá deilunni (sem stundum minnir helst á trúarbragðadeilur) um það hvort LabVIEW geti yfirleitt talist vera forritunarmál. Margir telja það ekki vera forritunarmál vegna þess að það er ekki skrifað sem texti, og að það sé ekki raðkeyrt. Og það kemur á óvart að þegar ákveða skal hvað teljist forritunarmál og hvað ekki, þá eru þeir þröngsýnastir sem menntaðir eru í tölvuvísindum.

Það sem mikilvægast er við nýju spurninguna er að hún snýr fullyrðingunni við. Upphaflega spurningin gefur óbeint til kynna að forritun til almennra nota sé á vissan hátt vandasamari eða eins konar yfir-skipanasett mælinga og sjálfvirkni sem þar með sé orðið undirkerfi almennrar forritunar. Í reynd er þessu öfugt farið.

Almennt, þá fást forritunarkerfi fyrir mælingar og sjálfvirkni við öll sömu viðfangsefni eins og almenn forritunarkerfi. Nefna má gagnagerð og aðferðafræði, skáarvinnsla, netsamskipti, notendaskil, tengsl við gagnagrunna, útprentun og fleira. En forritunarkerfi fyrir mælingar og sjálfvirkni þurfa einnig að fást við fjölda viðfangsefna sem almenn forritunarkerfi eru laus við, eins og tengsl við raunverulegar mælingar og stýringar, rauntímakerfi og ytri vélbúnað. Mælinga- og sjálfvirknikerfi nota augljóslega yfir-skipanasett miðað við þarfir almennra forritunarkerfa.

Ef tæki A og tæki B má nota til vissra verkefna, en tæki B hefur mun fleiri aðgerðir sem gagnast enn fleiri verkefni, hvort tækið er í reynd almennara? Þannig er með LabVIEW. Hæfni LabVIEW til mælinga og sjálfvirkni er ekki vegna þess að undirstöðuþættir forritunar séu takmarkaðir á einhvern hátt, heldur vegna þess að þeir eru auknir og bættir.

Vegna þessa er mikilvægara að spyrja: Er hægt að nota LabVIEW til að smíða almennar lausnir? Í stað þess að spyrja: Er LabVIEW almennt forritunarmál? Við viljum ekki takmarka viðhorf eða áframhaldandi þróun LabVIEW með því að líta einungis á það sem forritunarmál.

LabVIEW er miklu meira en forritunarmál. Það er afar þróað forritunarumhverfi, sem leyfir notendum að ljúka frumgerðum á örstuttum tíma, og styður listavel við áframhaldandi hugbúnaðarþróun stig af stigi, allt frá mælikerfum, sjálfvirkni og sjálfstæðum rauntímalausnum og til almennra lausna. Og með nýju tengslunum við FPGA (field programmable gate arrays) er LabVIEW einnig orðið vélbúnaðarþróunartæki.

Gagnaflæði

Hjarta LabVIEW er vel skipulagt gagnaflæði. Hugtakið um gagnaflæði hefur lengi verið þekkt og er vel rannsakað. Gagnaflæði er miklu betra reiknilíkan en skipanaflæðið í vinsælum textabundnum forritunarmálum vegna þess að það er að uppleggi gert fyrir samsíðavinnslu. C/C++ og Basic eru það ekki og þurfa þess vegna að reiða sig á kerfisköll til stýrikerfisins til að ná samsíðavinnslnni. Þessi annmarki gerir það að verkum að þýðandinn getur ekki tryggt að sameiginlegur kóði sé nægjanlega verndaður, sem aftur leiðir til þess að erfitt er að smíða kerfi fyrir samsíðavinnslu. Þetta vandamál eru ekki til staðar í LabVIEW. Jafnvel byrjandi getur byggt lausn sem byggist á samsíðavinnslu. Og það þarf ekki sérstakar tilfæringar eða kunnáttu til að keyra lausnina á mörgum tengdum örgjörvum.

Lengi hefur verið litið á Gagnaflæði sem hönnunaraðferð í viðskiptahugbúnaði. Það hefur þróast sem valkostur í tölvuhönnun til að forðast flöskuháls þann sem kenndur er við Von Neumann. Greining á Gagnaflæði er aðalatriðið í bættri virkni þýðanda. En hvernig væri að forrita beint með gagnaflæðið innbyggt? Hin eðlilega framsetning á gagnaflæði er teikning eða flæðirit. En það var fyrst með mús og grafik á tölvuskjám að það varð mögulegt að gera slíkar teikningar á tölvu. Til samanburðar, þá er textainnsláttur á gagnaflæði sambærilegur við að gera skriflega lýsingu á götukorti, og það er bæði tímafrekt og flókið.

Þegar ég sýni LabVIEW flæðirit, þá heyrir ég stundum þessa spurningu: "Hvar er kóðinn?" rétt eins og flæðiritið sé gagnslaust nema það búi til texta. Þá get ég ekki annað en dáðst að því, hversu vel hugbúnaðariðnaðinum hefur gengið að sannfæra heiminn um að takmörkun algengustu forritunarmála sé í raun og veru mikill kostur. Í raun er þessu öfugt farið því það er mikill ókostur að takmarka sambandið milli forritarans og forritsins sjálfs við einfaldan texta. Þegar fólk meðhöndlar geisladisk með tónlist, þá spyr enginn hvar textinn sé. Við höfum ekki, og þurfum ekki, textaútgáfu af innhaldi geisladiska, vegna þess að við höfum hugbúnaðarverkfæri sem geta meðhöndlað og spilað tónlist á tvítöluformi. Sama gildir um myndefni. Myndbandstæki taka upp og spila myndskleið án þess að texti komi nærri.

Af hverju eru forritunarmál öðruvísi? Af sögulegum ástæðum, þá var nauðsynlegt að hafa sérstakan ritþór til að skrifa forrit með og þýðanda til að þýða forritið í vélamál, og auðveldast var að tengja kerfið saman með einfaldasta samnefnaranum, stafrófinu. Þegar tölvur urðu stærri og afkastameiri, þá komu fram sérhönnuð þróunarkerfi en samt var haldið í einfaldasta samnefnarann, stafrófið. Þýðandinn hirðir ekkert um allar þær upplýsingar sem koma fram í skipulagi textans. Það hafa verið gerðar margar tilraunir til að búa til þýðanda sem byggist á forsníði textans, en þær hafa allar mistekist vegna þess að innsláttur staf fyrir staf er svo mikið torf. Þýðandinn les einungis 7 bita textastreng sem sleginn hefur verið inn með ritþóri. Við notum mismunandi textastærð og liti og snið þegar gögnin eru birt fyrir fólk að lesa en það er engin tilraun gerð til að nota þessar víddir í forritunarritþórum eða þýðundum.

Það vekur undrun að ýmsir vísindamenn sem gert hafa tilraunir með myndræna framsetningu forritunarkerfa hafa aðeins unnið við hluta vandamálsins. Þeir hafa gert ritþór sem býr til mynd sem þýðandinn vinnur úr. Myndin er forritið og það er jafn auðvelt að skilja það þegar það

hefur verið prentað út, eins og að sjá það á skjánum. En þýðandinn hirðir hins vegar ekki um það hvernig myndin er gerð þegar hann þýðir frá grunni.

Í LabVIEW er tekinn annar póll í hæðina. Flæðirit LabVIEW er í sjálfu sér ekki mikið meira en teikningar með hafsþjó af upplýsingum sem birtast eftir þörfum, t.d. skýringartextar sem ekki þurfa að flæða um allan skjá. Það má prenta út LabVIEW forrit en það er miklu auðveldara að skoða það og fletta því innan úr LabVIEW. En, þýðandinn þarf ekki að þýða flæðiritið því það hefur þegar verið þýtt. Myndþórin setur saman þýðinguna jafnóðum meðan flæðiritið er teiknað. Það er sent er til þýðandans er miklu umfangsmeira en sú teikning sem sést á skjánum. Og þess vegna verður þýðingin mjög hraðvirk og notandinn verður tæpast var við hana. Af þessum sökum verður engin töf, það má prófa breytingu strax eftir að hún hefur verið gerð.

Hraði þýðandans er einungis ein af mörgum ástæðum fyrir auknum afköstum sem forritarar verða varir við þegar þeir vinna í LabVIEW. Vegna þess að myndþórin smíðar þýðinguna jafnóðum, þá getur hann samstundis sagt frá villum sem þá er auðvelt og fljótlegt að lagfæra.

Myndþórin er með ríkulegt sett af skipunum og aðgerðum til að setja saman notandaviðmót. Sú staðreynd að sérhver eining, stundum kölluð vía, er með eigið notandaviðmót gerir að verkum að gagnvirk prófun má auðveldlega framkvæma, skref fyrir skref, og án þess að það þurfi að skrifa meiri kóða. Sá hluti forrits sem þarf að ljúka áður en hægt er að gera prófanir að einhverju marki er miklu minni í LabVIEW en í hefðbundnum forritunarmálum. Þessi staðreynd gerir þróunar- og prófanaferilinn mjög hraðvirkan.

Það er einnig auðvelt að nota gagnagerðir í flæðiritinu. Strengi og fylki má senda fram og aftur í aðgerðir, án þess að hugsa um atriði eins og minnisþörf, sem gerir að verkum að ýmiss vandamál koma aldrei fyrir, eins og tapað eða yfirskrifað minni .

Heildaráhrif allra þessara eiginleika í LabVIEW er stórauðin afköst forritara. Vitnisburður úr mörgum áttum bendir til þess að afköstin í LabVIEW séu fjórum til tíu sinnum meiri en afköst í hefðbundnum forritunarmálum. Þetta er aðalástæða þess að ekki ætti að telja LabVIEW vera almennt forritunarmál. Það er af miklu hærri getuflokki sem hæfir jafnt borðtölvum sem sjálfstæðum örgjörfum. Það væri ákaflega hógvært að segja að LabVIEW sé einungis forritunarmál.

Að lokum

Á meðan við fáumst við að þróa og bæta LabVIEW, þá munu afköst og gæði aukast, aðgerðum fjölga, og LabVIEW mun keyra á fleiri og fleiri örtölvuknúnum tækjabúnaði. Við látum ekki takmarka okkur af hinum hefðbundnu landamærum milli forritunarmáls, ritþórs, þýðanda, villuleitara, tækjarekils og þess háttar, því við trúum því að sýn okkar á forritunarmál geri mögulegt að bæta gæði og einfaldleikann í forritun. Og, með því að vinna náið með LabVIEW notendum munum við sjá alla þessa möguleika blómstra.

Að lokum, er LabVIEW almennt forritunarmál? Nei, það er miklu meira en það. Má nota LabVIEW til að búa til almennar lausnir. Já, svo sannarlega.

Um höfundinn

Jeff Kodosky var einn stofnanda National Instruments árið 1976 og var framkvæmdastjóri rannsókna og þróunar frá 1980 til ársins 2000. Hann er best þekktur fyrir að hafa fundið upp LabVIEW, sem er aðalflugbúnaðarkerfi fyrirtækisins.