

GRAFISKO SASKARŅU IZSTRĀDES PROBLĒMU RAKSTUROJUMS UN RISINĀJUMI *GRAPHICAL USER INTERFACE DEVELOPMENT PROBLEM DESCRIPTION AND SOLUTIONS*

Autori: **Arturs ŠKAPARS**, e-pasts: varabei@inbox.lv,
Aigars LADUŠEVŠ, e-pasts: aigaigarss@inbox.lv
Zinātniska darba vadītājs: Dr.sc.ing. **Artis TEILĀNS** e-pasts: artis.teilans@rta.lv
Rēzeknes Tehnoloģiju akadēmija
Rēzekne, Atbrīvošanas aleja 115

Abstract: *This article is about graphical user interface development problem description and possible solution summary. The goal of this work is to delve into specifics of common graphical user interface development problems and controversies, and list possible solutions that can be used to overcome issues.*

Keywords: *command line interface, graphical user interface, icon, pointer.*

Ievads

Laikam ejot, programmatūras saskarnes ir attīstījušās no komandu rindu konsolēm, kuras prasīja no lietotāja specifisku komandu sintakses zināšanas, uz grafiskajām saskarnēm. Salīdzinot ar komandu rindu saskarnēm, grafisko saskarņu galvenā priekšrocība ir tajā, ka pat sarežģītākās operācijas var tikt izpildītas ar pāris peles klikšķiem, kamēr komandu rindu saskarnēs šī pati operācija varētu prasīt no lietotāja ievadīt vairākas komandas teksta veidā. Grafiskās saskarnes ne tikai vizuāli attēlo programmatūras informāciju, bet arī sniedz intuitīvu informāciju par programmatūras sastāvdaļām. Labs piemērs tam ir specifiskas ikonas, kuras attiecas uz kādas darbības kontekstu, kā piemēram ikona, kurā attēlota zobrata, vai uzgriežņa atslēgas bilde dod lietotājam intuitīvu priekšstatu ka attiecīgā sadaļa visdrīzāk ir saistīta ar programmatūras iestatījumiem. Tieši šī iemesla dēļ ir iespējams ērti darboties ar tehniku, kura atbalsta grafiskās saskarnes un tieši tādēļ grafiskās saskarnes izmanto gandrīz vai katra modernā tehnoloģija. Grafiskās saskarnes ir sastopamas mobilajās ierīcēs, datoros, sadzīves tehnikā, automašīnās un daudz kur citur.

Neatkarīgi no ierīces veida, vai tas ir dators, vai automašīnas GPS sistēma, programmatūra šādām ierīcēm nemitīgi tiek atjaunināta, pievienojot jaunas funkcijas, paplašinot un modificējot esošās, un ne katrs lietotājs aizdomājas par domu procesu aiz grafiskās saskarnes izstrādes, vai pilnveidošanas. Grafiskās saskarnes izstrādes, vai modificēšanas procesā izstrādātāja priekšā stājas vairākas problēmas, kuras var traucēt vēlamajai saskarnes funkcionalitātei. Grafiskās saskarnes mērķis ir attēlot vizuālu informāciju un lietotāja-programmatūras mijiedarbības elementus tādā veidā, lai tas būtu maksimāli ērti lietotājam. Bieži vien uzlabojot, vai pievienojot kādu funkciju var pasliktināties kāda pavisam cita funkcija, kas savukārt rada šo pretrunu grafisko saskarņu izstrādes sfērā, kad pievienojot vairāk funkcionālo elementu var samazināties saskarnē attēlotās informācijas daudzums. Tieši ar šāda veida problēmām saskaras saskarņu izstrādātāji dienu dienā. Bet par laimi vairākas no šāda veida problēmām laika gaitā ir radušas vairākus radošus risinājumus, daži no kuriem tiks apskatīti šajā darbā.[4]

Liela elementu daudzuma attēlošana saskarnē

Viena no izplatītākajām problēmām grafiskajās saskarnēs ir funkcionālo elementu pārpilde, kā rezultātā tiek samazināta telpa, kurā lietotājam tiek attēlota vēlamā informācija. Piemēram grafiskās rediģēšanas programmā lietotājam ir pēc iespējas labāk jāredz rediģējamais grafiskais objekts, bet tāpat arī lietotājam nepieciešama pieeja pie rediģēšanas funkcijām, kurām tāpat jābūt attēlotām programmas saskarnes telpā. No tā izriet sakarība - jo lielāks

daudzums funkciju attēlots saskarnē, jo mazāk programmatūras saskarnes telpas tiek atvēlēts informācijas attēlošanai uz lietotāja ekrāna.

Viens no risinājumiem šai problēmai ir lielākas grafiskā izvada ierīces izmantošana ar augstāku izšķirtspēju, kas ļaus samazināt uz ekrāna redzamo elementu izmēru, tādā veidā paaugstinot programmatūras saskarnes ietilpību. Kaut arī šis risinājums ir piemērojams šai problēmai, tas prasa papildus uzmanību tieši no lietotāja puses, nevis no izstrādātāja, tieši izstrādātājam ir jābūt par galaproduktu, kurš nonāk līdz lietotājam.

Vēl viens risinājums ietver sevī programmatūras virtuālās telpas pielietošanu, kas ļautu programmatūras saskarnei saturēt papildus telpu, neatkarīgi no lietotāja ekrāna izmēriem. Šis risinājums ir sastopams grafiskās rediģēšanas programmās.

Grafiskās saskarnes elementu sadalījums kategorijās un apakškategorijās ļauj samazināt uz ekrāna attēloto funkcionālo elementu daudzumu, bet tajā pat laikā tas prasa no lietotāja vairāk darbību lai tikt pie vēlamās funkcijas.

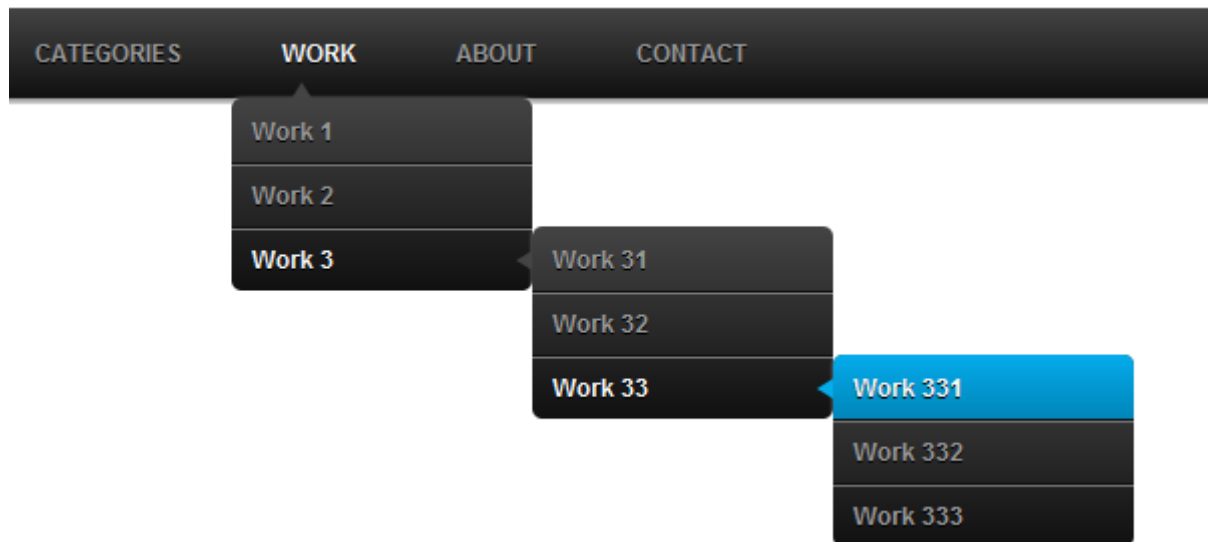
Iespēja attēlot funkcijas rediģējamās paneļos, kuros lietotājs var pievienot vēlamās funkcijas un kuri paliek paslēpti, līdz lietotājs neizvēlas specifisku opciju, vai nenovieto peles kursoru specifiskā vietā. Šādā veidā iespējams realizēt lietotājam vēlamās informācijas attēlošanu samazinot funkcionālo elementu attēlošanu saskarnē. Kaut arī šis risinājums ir efektīvs, tas prasa no lietotāja papildus zināšanas par programmatūru.

Izkrītošās izvēlnes ir viens no populārākajiem risinājumiem šai problēmai, kas pastāv gandrīz vai katrā programmatūras saskarnē. Lietotājs navigē starp vairākām apakškategoriju izvēlnēm, līdz neatrod vēlamo funkciju (skatīt 1. attēlu).

Iespēja pietuvināt un attālināt saskarnes informatīvo sadaļu ļauj saglabāt funkcionālo saskarnes elementu daudzumu un tajā pat laikā ļauj lietotājam regulēt informācijas daudzuma attēlošanu saskarnē.

Viens no interesantākajiem risinājumiem šai problēmai ir saskarnes elementu izvietošana uz triju dimensiju kuba plaknēm, starp kurām pārslēdzas lietotājs. Šis veids ir plaši sastopams mūsdienu mobilajās ierīcēs.

Apvienojot augstākminētos risinājumus ir iespējams saskarnes informācijas un funkcionālo elementu attēlošanas pretrunu.[1]



1. attēls. Izkrītošais izvēlnes logs

Saskarnes grafisko elementu izmēri

Viens no svarīgākajiem saskarnes kritērijiem ir saskarnes elementu redzamība. Saskarnes tiek veidotas lai nodrošināt lietotāja-mašīnas mijiedarbību un lai panākt šīs mijiedarbības efektivitāti lietotājam ir skaidri jāsaprot saskarnes izvadītā informācija. Grafisko saskarņu gadījumā informācijai jābūt skaidri redzamai un salasāmai, kas var nozīmēt lielāku ikonu, vai

teksta izmēru nepieciešamību. Bet palielinot elementu un teksta izmēru var samazināties saskarnē attēlotās informācijas, vai funkcionālo elementu daudzums.

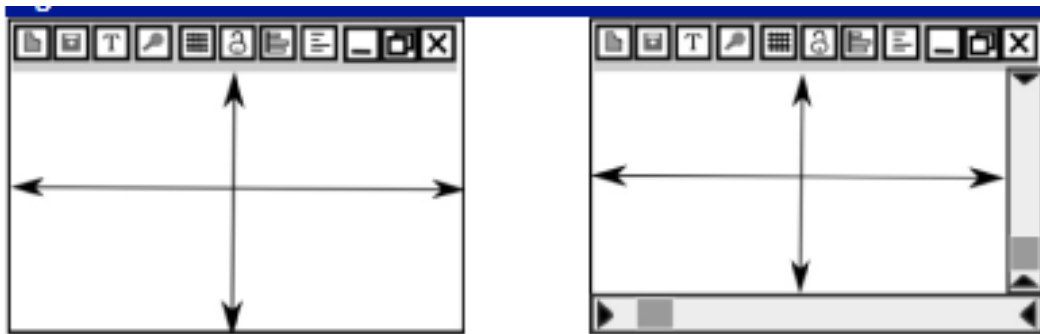
Labs risinājums šāda veida problēmai ir attēlojamo elementu skaita samazināšana, attēlojot tikai svarīgākos un pārējos pēc lietotāja pieprasījuma, elementu grupēšana un piekļuve tiem izvēloties atbilstošo grupas identifikatoru. Šādā veidā elementi tiek hierarhiski tiek izvietoti attiecīgajās kategorijās un lietotājs navigē starp to apakškategorijām. Kaut arī šis veids ir efektīvs un plaši pielietots, tāpat tas prasa no lietotāja laiku.

Dinamiska izmēra grafiskās ikonas, kuras maina savu izmēru kad lietotājs novieto peles kursoru uz ikonas virsmas. Kaut arī risinājums ir efektīvs un plaši pielietots, reizēm pie attiecīgās monitora izšķirtspējas var novest pie tā ka paplašinātā grafiskā ikona var bloķēt kādu citu ikonu.

Kā viens no inovatīvākajiem risinājumiem ir virtuālās realitātes aksesuāru izmantošana, ar kuru palīdzību lietotājs var atrasties virtuālajā telpā un aplūkot programmatūras saskarni 360 grādu rādiusā. Šis risinājums nav tik populārs kā citi analogi, jo vairumam tehnoloģiju nav virtuālās realitātes saskarnes atbalsta, bet iespējams nākotnē tam būs plašāks pielietojums.

Izslīdošās saskarnes izmantošana ir līdzīgs variants vienam no augstāk minētajiem, bet atšķirība ir tajā, ka lietotājam pēc pieprasījuma tiek attēlota vēl viena programmatūras saskarne nevis atsevišķs tās elements. Šī saskarne satur kategorizētu informāciju un grafiskos mijiedarbības elementus.

Slīdņu izmantošana lai manipulēt ar saskarnes virtuālās telpas izvietojumu ir labs risinājums šai problēmai (skatīt 2. attēlu). Šādā veidā saskarnē iespējams attēlot vairāk grafisko elementu nesamazinot to izmērus, bet tajā pat laikā slīdņi aizņem vietu saskarnē. Šis risinājums kaut arī ir efektīvs, bet tajā pat laikā tas rada jaunu problēmu – slīdņi aizņem vietu saskarnes robežās un tādā veidā samazina saskarnes attēlojuma laukumu. Kā risinājums slīdņu problēmai ir iespējams aizvietot slīdņus ar atsevišķām pogām, ar kuru palīdzību tiek veiktas slīdņu funkcijas, kā arī labs risinājums ir attēlot slīdņus tikai pēc lietotāja pieprasījuma, piemēram kad lietotājs novieto peles kursoru tieši uz saskarnes malas.[1]



The space is further reduced
 by using scroll bars

2. attēls. Slīdņu izmantošana saskarnē

Lietotāji un lietotāju vēlmes

Saskarnes ir paredzētas tieši lietotājiem, lai tie spētu efektīvi darboties ar konkrēto tehnoloģiju un loģisks solis saskarņu problēmu risināšanā ir saziņa ar gala lietotājiem. Neskatoties uz to cik funkcionālu saskarni vēlas lietotājs, tāpat šis pats lietotājs arī vēlas šo pašu saskarni ērtu, saprotamu, intuitīvu un patīkamu darbā. Šajā punktā rodas kārtējā problēma. Vairāki lietotāji var vēlēties vairākas dažādas saskarnes īpašības un nesatiekot šīs vēlmes lietotāju darbs var tikt traucēts.[3]

Kā risinājumu izstrādātāji var piedāvāt vairākas veidnes no kurām lietotājs var izvēlēties sev piemērotāko. Šīs veidnes attēlo saskarnes elementu novietojumu un saturu. Veidnēm arī

piemīt profesionālā līmeņa kategorijas, kā piemēram saskarnes veidne priekš iesācējiem, vai profesionāļiem.

Vēl viens risinājums ir izstrādātājiem izveidot iespēju pašiem lietotājiem rediģēt saskarnes elementus un to saturu, tādā veidā lietotājs spēs izveidot sev piemērotu darba saskarni. Kaut arī efektīvs, šis risinājums prasa no lietotāja jau iepriekš esošas zināšanas par programmatūru un tās saskarni lai efektīvi izveidot sev piemērotu saskarnes dizainu.[1]

Tāpat gan lietotāji, gan saskarņu izstrādātāji vēlas lai saskarnes ir viegli lietojamas, viegli apgūstamas un lai tajās ir viegli navigēt, jeb citiem vārdiem lai saskarnes ir draudzīgas lietotājam. Kā arī lietotāji vēlas šo lietotāja draudzības līmeni atkarīgu no lietotāju profesionālajām iemaņām darbībā ar konkrēto saskarni. Iesācēji nevēlēsies darboties ar saskarni, kura ir paredzēta profesionāļiem, jo priekš iesācēja tā var būt sarežģīta, grūti navigējama un mulsinoša. Tas tāpat attiecas uz profesionāļiem, kuri nevēlas darboties ar iesācēja līmeņa saskarnes izklājumu.[2]

Kā risinājumu izstrādātāji var pievienot dažādas saskarņu kategorijas, kuras satur gan profesionāļu, gan iesācēju vēlnes, kā arī pievienot lietderīgus teksta veida paziņojumus pēc lietotāja pieprasījuma.

Viens no inovatīvākajiem risinājumiem šāda veida problēmai ir intelektuālā aģenta iesaiste lietotāja darbībā ar saskarni. Intelektuālā aģenta programmatūra novēro lietotāja darbības un sniedz tās funkcijas, kuras būtu nepieciešamas lietotājam noteiktajā laikā. Šis risinājums nav tik populārs kā varētu būt citi, pateicoties tam ka ir pielietojams ne katrā darbības sfērā.[1]

Lietotāja kļūmes

Grafiskās saskarnes sniedz lietotājiem darbību brīvības sajūtu, kad lietotājs spēj manipulēt ar saskarni rediģējot tās izmērus un atrašanās vietu uz monitora, vai citas izvadierīces, un spēj rediģēt saskarnē esošos elementus atbilstoši savām vēlmēm. Kā arī saskarnes piedāvātās operācijas tiek izpildītas ar pāris taustiņu un/vai peles klikšķu un kustību veikšanu. Bieži vien šāda izpildes brīvība var novest pie kļūmēm un mulsināt lietotājus. Šis ir kārtējais saskarņu izstrādes aspekts, par kuru ir jāparūpējas izstrādātājiem. Gan izstrādātāji, gan lietotāji vēlas saskarnes darbības plūstošas un intuitīvas, pretējā gadījumā samulsināti lietotāji lieki tērē laiku cenšoties saprast savu darbību sekas saskarnē.[3]

Bieži vien izstrādātāji implementē saskarnēs funkcijas, kuru mērķis ir validēt lietotāja izdarīto izvēli. Visbiežāk šī funkcija tiek realizēta izvadot lietotājam teksta paziņojumu, kurā īsumā tiek aprakstīta operācija un tiek pieprasīts apstiprinājums turpmākajai darbībai. Kaut arī efektīvi, šis paņēmieni patērē nedaudz no lietotāja laika.

Dažos gadījumos, kad lietotājs pieļauj kļūdu, šī kļūda ir balstīta uz lietotāja apmulsuma, kuru var radīt liels daudzums funkcionālo elementu saskarnē un lietotājs var izvēlēties nepareizo. Šajā gadījumā izstrādātāji pievieno funkciju, paslēpj ar pašreizējo darbību nesaistītās funkcijas, tādā veidā attēlojot mazāku funkciju daudzumu lietotājam, kas savukārt samazina iespēju izvēlēties ar operāciju nesaistīto funkciju.

Izstrādes procesā saskarnē iespējams pievienot funkciju, kura kļūdas gadījumā piedāvās lietotājam risinājumu, balstoties uz lietotāja veiktajām darbībām, kuras noveda pie kļūdas. Šāda veida funkcija ir sastopama integrētās izstrādes vides saskarnēs un tā piedāvā risinājumus pēc lietotāja pieprasījuma.

Izstrādes procesa laikā iespējams parūpēties par dažādu tipa saskarņu veidnēm, kuras paredzētas lietotājiem ar atšķirīgu pieredzes līmeni, kā piemēram saskarnes veidne priekš iesācējiem, vai priekš profesionāļiem. Šādā veidā, izvēloties sev attiecīgo veidni saskarne saturēs funkcijas paredzētas iesācēja darbam, vai profesionāļa darbam.

Vēl viens risinājums lai novērst lietotāja kļūmi ir grafiski attēlot lietotājam darbības rezultātu neizpildot pašu darbību, pēc kā lietotājam būs pieejams vairāk informācijas, lai pieņemt lēmumu par darbības izpildi, vai pārtraukšanu.[2]

Summary

Nowadays, many devices use graphical user interface to allow user-machine communications. These interfaces provide valuable information and interaction options to user. Graphical user interface developers have to balance amount of information that is displayed to user and amount of functional elements on the screen space.

Literatūras saraksts

- [1] “Contradictions in a GUIThe Triz Journal.” [Online]. Available: <https://triz-journal.com/demonstrating-contradictions-in-a-gui/>.
- [2] “Improve Product Design: Watch What End Users DoThe Triz Journal.” [Online]. Available: <https://triz-journal.com/innovation-tools-tactics/incremental-innovation-tools/improve-product-design-watch-end-users/>.
- [3] “Ideal Final Interface - What Should Have Come First, User Interface or Function?The Triz Journal.” [Online]. Available: <https://triz-journal.com/ideal-final-interface-what-should-have-come-first-user-interface-or-function/>.
- [4] “History of the graphical user interface.” [Online]. Available: History of the graphical user interface - Wikipedia https://en.wikipedia.org/wiki/History_of_the_graphical_user_interface.