



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

Vaidas Valukonis

ARCHESTRA IDE ŽMOGUS MAŠINA SAŠAJOS TYRIMAS

Baigiamasis magistro projektas

Vadovas

Doc. dr. Leonas Balaševičius

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS

**ELEKTROS IR ELEKTRONIKOS FAKULTETAS
AUTOMATIKOS KATEDRA**

ARCHESTRA IDE ŽMOGUS MAŠINA SĄSAJOS TYRIMAS

Baigiamasis magistro projektas
Valdymo technologijos (kodas 621H66001)

Vadovas

Doc. dr. Leonas Balaševičius

Recenzentas

Projektą atliko

Vaidas Valukonis

KAUNAS, 2015



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos fakultetas

(Fakultetas)

Vaidas Valukonis

(Studento vardas, pavardė)

Valdymo technologijos, 621H66001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „ArchestrA IDE žmogus mašina sąsajos tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 ____ m. _____ d.

Kaunas

Patvirtinu, kad mano **Vaido Valukonio** baigiamasis projektas tema „Archestra IDE žmogus mašina sąsajos tyrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Valukonis, V. Archestra IDE žmogus mašina sąsajos tyrimas. Magistro baigiamasis projektas / vadovas doc. dr. Leonas Balaševičius; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Automatikos katedra.

Kaunas, 2015. 45 psl.

SANTRAUKA

Pagrindinis darbo tikslas išsiaiškinti grafikos elementų kūrimo galimybes ArcestrA IDE programiniame pakete.

Darbe buvo išanalizuotos ArcestrA IDE galimybės, struktūra ir simbolių kūrimo pakopos. Taip pat apibūdinti pagrindiniai skirtumai tarp ArcestrA ir InTouch programinių paketų. Tam, kad būtų parodytos ArcestrA IDE grafikos galimybės, buvo sukurta simbolių biblioteka, detalai aprašant simbolių kūrimą. Taip pat išsiaiškinti pagrindiniai skirtumai ir privalumai kuriant aplikacijas, kai naudojama ArcestrA IDE ir InTouch grafinė biblioteka. Sukurtos aplikacijos, kuriose žmogus mašina sąsajos realizavimui naudojama tik ArcestrA IDE grafinė biblioteka arba InTouch grafinė biblioteka. Iširta sukurtai grafinei sąsajai atvaizduoti reikalinga atmintis, kai aplikacija vykdoma projektavimo ir valdymo stotyse.

Reikšminiai žodžiai (iki 8 žodžių):

SCADA

ArcestrA IDE

Žmogus mašina sąsaja (HMI)

Valukonis, Vaidas. Research of Archestra IDE human machine interface. Final project of *masters degree* / supervisor Assoc. Prof. Dr. Leonas Balaševičius; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, Department of Automation.

Kaunas, 2015. 45 psl.

SUMMARY

The main objective of this research was to find out the possibilities of creating the graphic symbols using Archestra IDE software.

There were analyzed the main possibilities and the structure of Archestra IDE and the main steps of symbol creation. Also there were described the main differences between Archestra and InTouch software. In order to show the possibilities of the graphics of Archestra IDE, there was created the Symbol Library also describing the creation of symbol in detail. Also there were described the main differences and advantages of creation of applications using Archestra IDE and InTouch graphic libraries. Moreover, there were certain applications created where in order to realize the human machine interface there were used only Archestra IDE graphic library or InTouch graphic library. During this research, there were analyzed the memory which was necessary to represent the graphic interface when the application was running in engineering stations and visualization nodes.

Keywords (up to 8 words):

SCADA

Archestra IDE

Žmogus mašina sąsaja (HMI)

Turinys

Įvadas	1
1. Arcestra IDE apžvalga	2
1.1 Skirtumai tarp InTouch ir Arcestra IDE įrankių.....	7
2. Aplikacijos kūrimas Arcestra IDE	9
2.1 InTouchViewApp šablonas.....	12
2.2 ViewEngine šablonas.....	14
2.3 Arcestra IDE simbolių apžvalga	15
2.4 Grafinės sąsajos elementų kūrimas.....	16
2.4.1 Šablono „Sraigtas“ kūrimas	18
2.4.2 Šablono „Kranas“ kūrimas	23
2.4.3 Šablono „Talpa“ kūrimas	26
2.5 Aplikacijos kūrimas	28
2.5.1 Globali simbolių korekcija.....	29
3. Intouch grafinės sąsajos panaudojimas Arcestra IDE aplinkoje	30
4. InTouch ir Arcestra IDE grafinei sąsajai reikalingų resursų palyginimas	34
Išvados	39
Literatūros sąrašas.....	40

Įvadas

Šiuolaikiniame pasaulyje nuolatos tobulėjant ir sudėtingėjant procesams būtinos technologijos, kurios padėtų suvaldyti procesus ir optimizuoti jų sąnaudas. Optimizuojant procesus galima sumažinti gamybos kaštus, gamybą padaryti efektyvesnę, o produktą kokybiškesnį. Vienas iš būdų patikimai suvaldyti, stebėti ir kontroliuoti procesus yra supervizorinio valdymo ir duomenų surinkimo sistemos (SCADA).

SCADA sistemoje sukūrus atitinkamą žmogus mašina sąsają galima stebėti ir valdyti procesus iš vienos vietos. Pasinaudojus šia programine įranga galima tai daryti ir nutolus nuo proceso vietos. Tai vienas patogiausių būdų valdyti procesus. Naudojant SCADA sistemas galima kaupti parametrų kitimo istoriją, taip daryti darbo našumo ar apimties analizes. Minėtose sistemose naudojami aliarmai, kurie gali būti kaupiami archyve ar tiesiog signalizuoti apie susidariusią situaciją. Viena iš SCADA sistemų yra Wonderware sukurta tokios sistemos platforma su integruotu ArchestrA IDE modulių (dar kitaip vadinama Galaxy).

Viena pagrindinių Wonderware sistemos platformos su ArchestrA IDE savybių lyginant su kitomis SCADA sistemomis yra tai, jog aplikacijos kuriamos iš suformuotų simbolių (šablonų), kas leidžia sutrumpinti panašių projektų atlikimo trukmę. Prieiga prie aplikacijų ir simbolių serverio taip pat leidžiama tik per *Galaxy*, vartotojai prileidžiami tik prie operatoriaus stoties. Tai suteikia papildomo saugumo nuo nesankcionuotos intervencijos. Iš paruoštų šablonų kuriant aplikacijas matomas dar vienas privalumas prieš kitas SCADA sistemas, t.y. script'us galima įdėti į simbolius, taip sumažinant klaidų script'e paieškos kaštus.

Darbo tikslas:

1. Ištirti ArchestrA IDE grafines galimybes, kuriant simbolius ir atminties sunaudojimą naudojant aplikacijas, kurtas panaudojant InTouch ir ArchestrA IDE grafines priemones.

Darbo uždaviniai:

1. Aprašyti ArchestrA IDE struktūrą ir funkcines galimybes.
2. Nustatyti pagrindinius skirtumus tarp InTouch ir ArchestrA IDE sistemų.
3. Pademonstruoti simbolių kūrimo ypatybes ArchestrA IDE sistemoje.
4. Ištirti grafines atminties sunaudojimą naudojant aplikacijas, kurtas InTouch ir ArchestrA IDE grafinėmis priemonėmis.

1. ArchestrA IDE apžvalga

SCADA (supervizorinio valdymo ir duomenų surinkimo sistema), tai kompiuterinė sistema, skirta kaupti ir analizuoti duomenis, bei valdyti procesą ar nutolusią įrangą realiuoju laiku.

ArchestrA IDE aplinka apima proceso valdymo ir duomenų surinkimo aplikacijos kūrimą. Ji apjungia tokius programinius modulius: InTouch, gamybos valdymo sistemą (MES), pramoninį aplikacijų serverį (IAS), receptūrinio valdymo programą InBatch, duomenų archyvavimo sistemą(Historian Server) ir kt. Naudojantis šiais moduliais informacija surenkama iš nutolusių įrenginių ir patalpinama serveryje. ArchestrA IDE leidžia kurti šablonus, kurie naudojami aprašant vieną ar keletą objektų, kas leidžia nesikartoti ir neapkrauti procesoriaus.

Procesams vizualizuoti naudojami objektai yra apibūdinti Archestra IDE kaip pramoninių programinės įrangos produktų aplinka iš Wonderware Corp.

Geriausiai žinomas produktas yra InTouch, kuriame galima projektuoti ir vykdyti sukurtas vizualizacijos aplikacijas, naudojant grafinius įrankius iš inTouch **Graphic tools**. Naudojant InTouch kaip pagrindą buvo sukurtas Wonderware Application Server modulis, kuriam realizuota ArchestrA technologija; pastaroji apjungia programinės įrangos architektūras (ang. *software architecture*) su produktais ir paslaugomis tam, kad būtų pasiektas efektyvus projektavimas ir automatikos ir informacijos aplikacijų vystymas pramoninėje srityje.

Archestra IDE technologija sujungia InTouch funkcionalumą ir Aplikacijų serverį (ang. *Application Server*) į vystomąją aplinką pavadinimu Archestra IDE. Šioje aplinkoje naudojamas objektais paremtas procesų kitimo bei kompleksinių elementų su iš anksto nustatytais parinktimis vaizdavimas.

InTouch vyrauja grafika, kurioje naudojami tradiciniai grafikos modeliai arba grafikos modelių bibliotekos su iš anksto sukurtomis ir apibūdintomis grafinių simbolių bibliotekomis. Po InTouch sujungimo su Application Server atsiranda galimybė naudoti ArchestrA grafinių simbolių bibliotekas.

Vizualizacijos programinių įrangų InTouch ir Application server integracijos koncepcija turi monitoringo sistemas su moderniais grafiniais objektais iš ArchestrA simbolių bibliotekos. WindowMaker aplinkoje (vizualizacijų kūrimo aplinka InTouch programoje) yra įmanoma naudoti ir InTouch, ir ArchestrA grafinius simbolius, tačiau tik po to, kai atliekamas abiejų produktų – InTouch ir Application Server - instaliavimas ir funkcinis sujungimas.

Archestra IDE turi ne tik grafinę vartotojo aplinką su objektų vaizdavimu, bet ir kitų įvairių objektų su iš anksto nustatytu funkcionalumu ir įvairiapusėmis galimybėmis, kurios padeda sukurti ir pateikti (ang. *deploy*) kompleksines aplikacijas.

Šie modernūs ArchestrA grafiniai objektai turi išplėstinio redagavimo ir esančių ypatybių modifikavimo galimybes, o ypač iš anksto suformatuotas ypatybes (ang. *properties*).

Aplikacijos kūrėjas ArchestrA grafikos šablonais naudojami iš grafinės bibliotekos, kuri patalpinta integruotoje kūrimo (ang. *development*) aplinkoje Archestra IDE, skiltyje *Graphic Toolbox*. Šioje skiltyje grafikos šablonai yra surūšiuoti specifiniuose įrankių rinkiniuose pagal jų pritaikymo sritį.

Archestra IDE grafiniai simboliai gali būti pritaikyti sprendimuose, kuriuose jie naudojami labai glaudžiai susieti su aplikacijų šablonais ir yra kitų funkcinių objektų dalis Archestra IDE programoje. Šis grafinis simbolis paprastai yra patalpinamas aplikacijos šablone, atvaizduojant proceso kitimą taip, kad **Automation object** yra sukuriama šio šablono nuoroda.

Kai kurie grafiniai simboliai, ypač iš grafinių įrankių bibliotekos *Displays*, yra funkcionalūs tik tuomet, kai tam tikrame aplikacijos šablone yra sukuriama jų nuoroda. Grafiniai simboliai iš Archestra įrankių bibliotekos *Displays*, realizuoti vizualizacijų aplikacijose, turi komunikavimo būsenos arba sistemos informacijos savybes. Tokiu būdu ši grafika turi būti prijungta prie specialių aplikacijos šablonų tam, kad visa informacija būtų prieinama pačiai vizualizacijai.

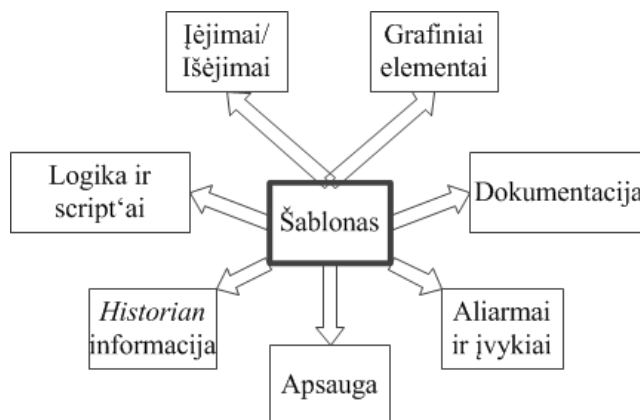
Keletas grafinių elementų ir šablonų, išdėstytų grafinėse bibliotekose su iš anksto nurodytomis funkcijomis, gali būti naudojami skirtingose pramonės šakose, todėl jie gali būti laikomi visiškai užbaigtais ir užrakintais nuo grafinio ir funkcinio keitimo. Kita vertus, yra galimybė šiuos objektus modifikuoti ArchestrA struktūroje, naudojant *Symbol Editor*, kai tuo tarpu dažniausiai naudojamos grafinių simbolių modifikacijos yra prieinamos WindowMaker (InTouch vizualizacijoje) konfigūravimo ir kūrimo aplinkoje. Pagrindinis kūrimo langas susideda iš tinklelio ir įvairių komponentų, pritaikytų grafinių simbolių su specifinėmis funkcijomis kūrimui.

Symbol Editor aplinka yra glaudžiai susieta su Microsoft .NET platforma tam, kad būtų kuriami objektai, naudojant grafinius simbolius ir *script*’ą iš .NET. Grafiniai simboliai iš ArchestrA bibliotekos turi modernių projektavimo savybių ir susideda iš grafinių sluoksnių, o pastarieji susideda iš elementarių šešėlių ir atspindžių posluoksnių. Galimybė importuoti InTouch *SmartSymbol* grafinius simbolius sukuria progą sujungti Archestra IDE ir InTouch grafinius simbolius. *Symbol Editor* skiltyje galima tik kurti, konfigūruoti ir modifikuoti istorijos grafikų (angl. *Trend*) simbolius.

Archestra IDE grafinių simbolių modifikavimas ir konfigūravimas yra atliekamas naudojant Archestra IDE įrankį *Symbol Editor* (žr. 2.8 pav.), iššaukiant grafinio simbolio atidarymą. [1]

Naudojantis sukurtais šablonais sukuriamas proceso modelis, kuris konfigūruojamas sudėtingoms supervizorinio valdymo funkcijoms įgyvendinti. Proceso modelis kuriamas nuosekliai įgyvendinant žemiau pateiktus žingsnius[2]:

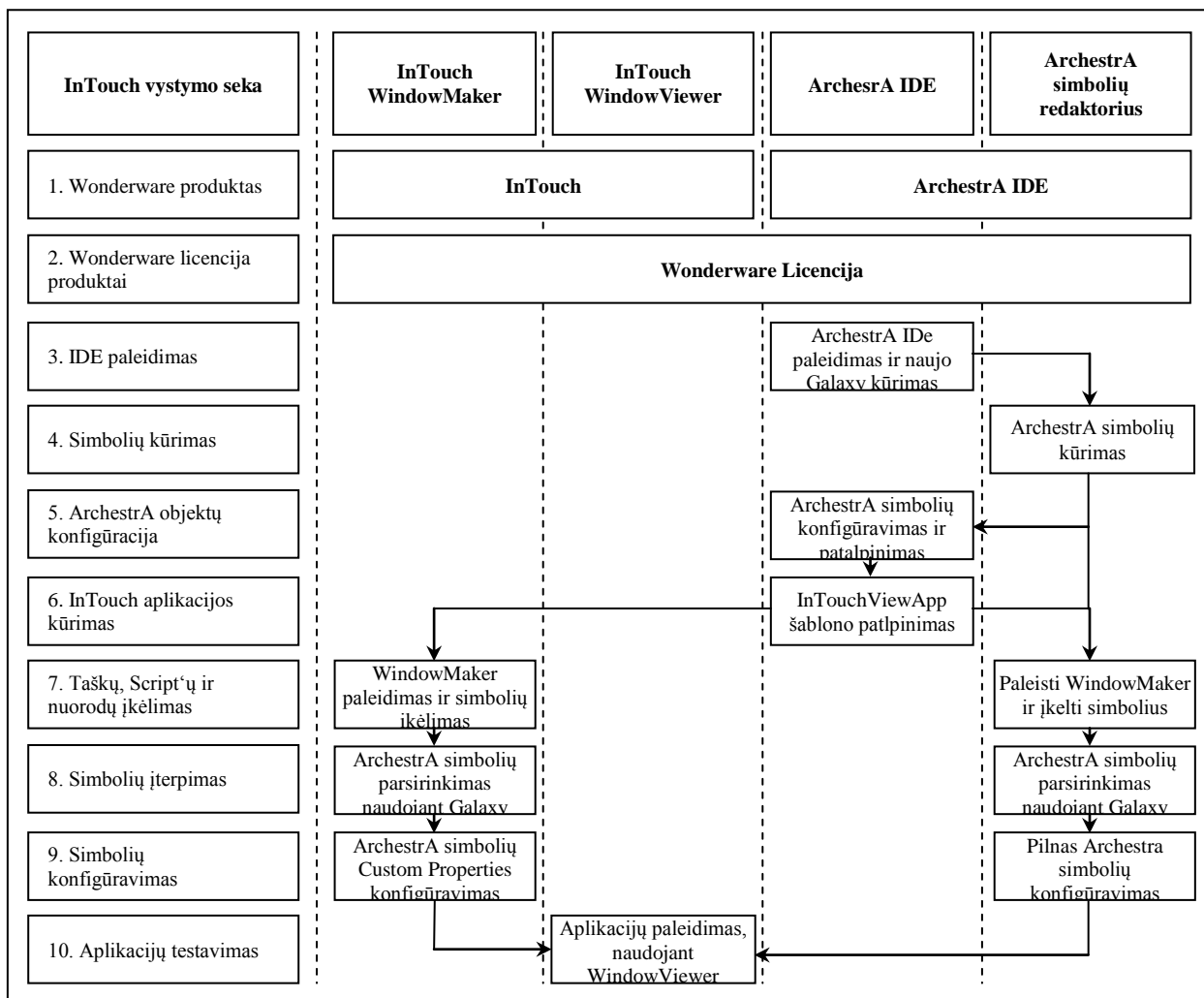
1. Proceso analizė, kad identifikuoti subprocesus ir įrangos išsidėstymą.
2. Sukuriamas panašios įrangos sąrašas. Identifikuojamos nepasikartojančios operacijos.
3. Kiekvienam įrenginiui yra sukuriami objektų šablonai (žr. 1.1 paveiksle).



1.1 pav. Šablono struktūra

4. Sudėtingesnių įrenginių šablonai sudaromi apjungiant paprastesnių objektų šablonus.
5. Sukonfigūruojami Objektų šablonų atributai, kurie leidžia prieiti prie proceso įėjimų-išėjimų.
6. Kuriama aplikacija naudojantis sukurtais šablonais. Aplikacijos grafinė sąsaja realizuojama Wonderware InTouch programinėje įrangoje.
7. ArchestrA IDE sukurta ir patikrinta aplikacija perkeliama į operatorių kompiuterius.
8. Pakeitimai padaryti objektų šablonuose atsispindi aukštesnio lygio komponentuose.

Proceso modelio kūrimo žingsniai atvaizduoti 1.2 paveiksle.

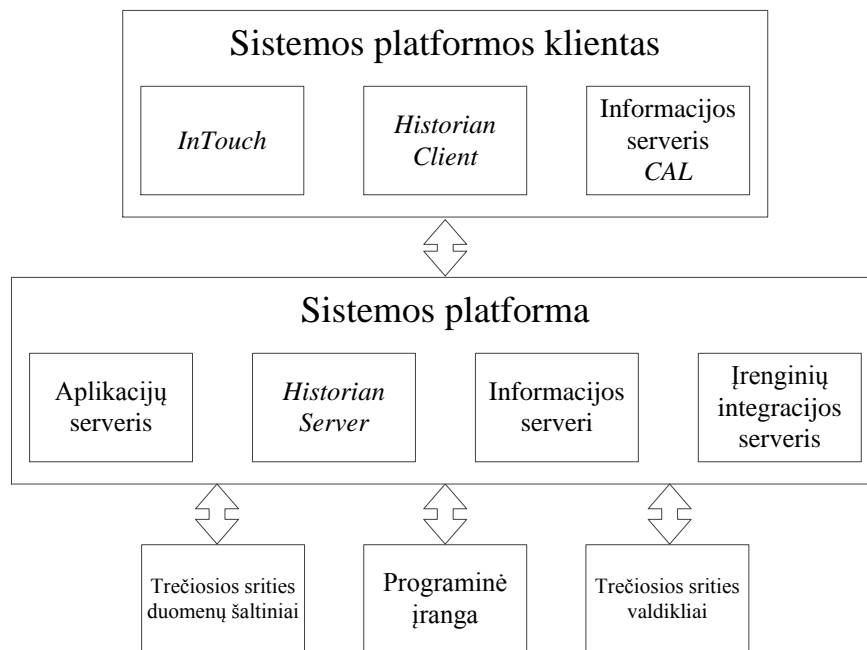


1.2pav. Aplikacijos kūrimo tvarka

ArchestrA IDE sistemos platforma susideda iš Sistemos platformos kliento, pačios Sistemos platformos ir kitų įrenginių. Sistemos platformos kliento grupei priklauso *InTouch*, *Historian Client* ir Informacijos serveris *CAL.InTouch* programinis paketas naudojamas kurti aplikaciją iš ArchestrA IDE sukurtų simbolių. *Historian Client* naudojamas registruoti ir sekti įvykius sistemoje. Informacijos serveris naudojamas sukurti operatoriui prieinamos informacijos terpę. Svarbu paminėti, jog Sistemos platformos klientai savo stotyse nekaupia informacijos ir neturi pačių aplikacijų, jų stotyse tik sukuriama terpė prieiti prie duomenų patalpintų Sistemos platformoje (žr. 1.3 paveiksle).

Pačią sistemos platformą sudaro „Aplikacijų serveris“, *Historian server*, „Informacijos serveris“ ir „Įrenginių integracijos serveris“. Aplikacijų serveryje saugomos aplikacijos ir simboliai. Kuriant aplikacijas ar darant pakeitimus, veiksmai atliekami minėtame serveryje. *Historian server* naudojamas rinkti visus procesų „aliarmus“, įvykius ir kintamųjų duomenis. Informacijos serveris naudojamas rinkti papildomai informacijai apie kintamuosius ir besikeičiančius duomenis. Įrenginių integracijos serveris naudojamas susieti šablonus su nutolusia įranga (pvz. programuojamais loginiais valdikliais) (žr. 1.3 paveiksle).

Trečiajame lygyje apibrėžiama įranga, kuri perduoda duomenis sistemos platformai ar vykdo sistemos nurodymus (žr. 1.3 paveiksle).



1.3 pav. ArchestrA IDE sistemos platforma

1.1 Skirtumai tarp InTouch ir ArchestrA IDE įrankių

Animacijoms realizuoti reikalingi įrankiai tokie, kaip kintamųjų aprašymo, grafikos animacijos priskyrimo ar veikimo logikos script'o kūrimo. Kai kurie ArchestrA IDE simboliams priskiriamų kintamųjų duomenų tipai skiriasi nuo InTouch animacijose naudojamų kintamųjų tipų. 1.1 lentelėje pateiktas duomenų tipų palyginimas.

1.1 lentelė. Kintamųjų tipų palyginimas [3]

Eil. Nr.	InTouch aplinkoje	ArchestrA aplinkoje	Apibūdinimas
1.	Discrete	Boolean	Loginė reikšmė. Pvz. 0 ar 1
2.	Integer	Integer	Sveikieji skaičiai. Pvz. -4, 7 ar 22
3.	Real	Float or Double	Slenkančio kablelio reikšmė su skirtingu tikslumu. Pvz. 3.141, -5.332, ar 1.343e+17
4.	Message	String	Eilutė reikšmė. Pvz. „Sveiki“
5.	n/a	DateTime	Datos ir laiko reikšmė. Pvz. „04/13/2006 04:03:22.222 AM“
6.	n/a	ElapsedTime	Slenkančio kablelio reikšmė nurodanti laiką išreikštą sekundėmis. Reikšmė atvaizduojama kaip parodyta žemiau, tačiau išsaugoma kaip slenkančio kablelio reikšmė. [-][DDDDDD] [HH:MM:]SS[.ffffff] Galimos reikšmės yra: <ul style="list-style-type: none"> • DDDDDD nuo 0 iki 999999 • HH nuo 0 iki 23 • MM nuo 0 iki 59 • SS nuo 0 iki 59 • .ffffff sekundės dalys
7.	n/a	InternationalizedString	Specialieji simboliai.

Pagrindiniai skirtumai tarp kintamųjų matyti 5-7 1.1 lentelės eilutėse. 5 ir 6 eilutėje apibūdinami datos ir laiko kintamieji, kurių neturi InTouch. Taip pat matyti, jog InTouch neturi specialiųjų simbolių, kas reikalinga kuriant daugiakalbius projektus.

Kai kurios animacijos taip pat skiriasi ir sugrupuotos lyginant InTouch animacijas ir ArchestrA simbolių animacijų redaktorių (žr. 1.2 lentelė).

1.2lentelė. Grafikos animacijų palyginimas [3]

Eil. Nr.	InTouch animacijos tipas	ArchestrA IDE simbolių animacijos tipas
1.	User Inputs – Discrete	User Inputs – Boolean
2.	User Inputs – Analog	User Inputs – Analog
3.	User Inputs – String	User Inputs – String
4.	Slider - Vertical	Slider – Vertical
5.	Slider – Horizontal	Slider – Horizontal
6.	Touch Pushbuttons –Discrete Value	Pushbutton – Boolean
7.	Action	Action Scripts
8.	Show Window	(not supported)
9.	Hide Window	(not supported)

Eil. Nr.	InTouch animacijos	ArchestrA simbolių animacijos redaktorius
10.	Line Color - Discret	Line Style - Boolean
11.	Line Color - Analog	Line Style - Truth Table
12.	Line Color - Discrete Alarm	converted to Line Style
13.	Line Color - Analog Alarm	converted to Line Style
14.	Fill Color - Discrete	Fill Style – Boolean
15.	Fill Color - Analog	Fill Style - Truth Table
16.	Fill Color - Discrete Alarm	converted to Fill Style
17.	Fill Color - Analog Alarm	converted to Fill Style
18.	Text Color - Discrete	Text Style - Boolean
19.	Text Color - Analog	Text Style - Truth Table
20.	Text Color - Discrete Alarm	converted to Text Style
21.	Text Color - Analog Alarm	converted to Text Style
22.	Object Size - Height	Height
23.	Object Size - Width	Width
24.	Location - Vertical	Location Vertical
25.	Location - Horizontal	Location Horizontal
26.	Percent Fill - Vertical	% Fill Vertical
27.	Percent Fill - Horizontal	% Fill Horizontal
28.	Miscellaneous - Visibility	Visibility
29.	Miscellaneous - Blink	Blink
30.	Miscellaneous - Orientation	Orientation
31.	Miscellaneous - Disable	Disable
32.	Miscellaneous - Tooltip	Tooltip
33.	Value Display - Discrete	Value Display - Boolean
34.	Value Display - Analog	Value Display - Analog
35.	Value Display - String	Value Display - String

8-9 1.2 lentelės eilutės matyti, jog InTouch turi galimybę rodyti ir paslėpti langus. ArchestrA IDE tokios galimybės neturi, tačiau tai nėra reikalinga, nes ArchestrA aplinkoje kuriami simboliai, kurie vėliau yra naudojami InTouch aplikacijose.

1.2 lentelės 12-13, 16-17 ir 20-21 eilutėse minimi InTouch „aliarmai“. ArchestrA simbolių redaktoriuje „aliarmai“ yra prijungiami prie animacijų kaip papildoma funkcija.

Script'as ArchestrA simbolių redaktoriuje gali būti parašytas taip pat, kaip InTouch WindowMaker. Esminiai skirtumai tarp script'ų pateikti 1.3 lentelėje.

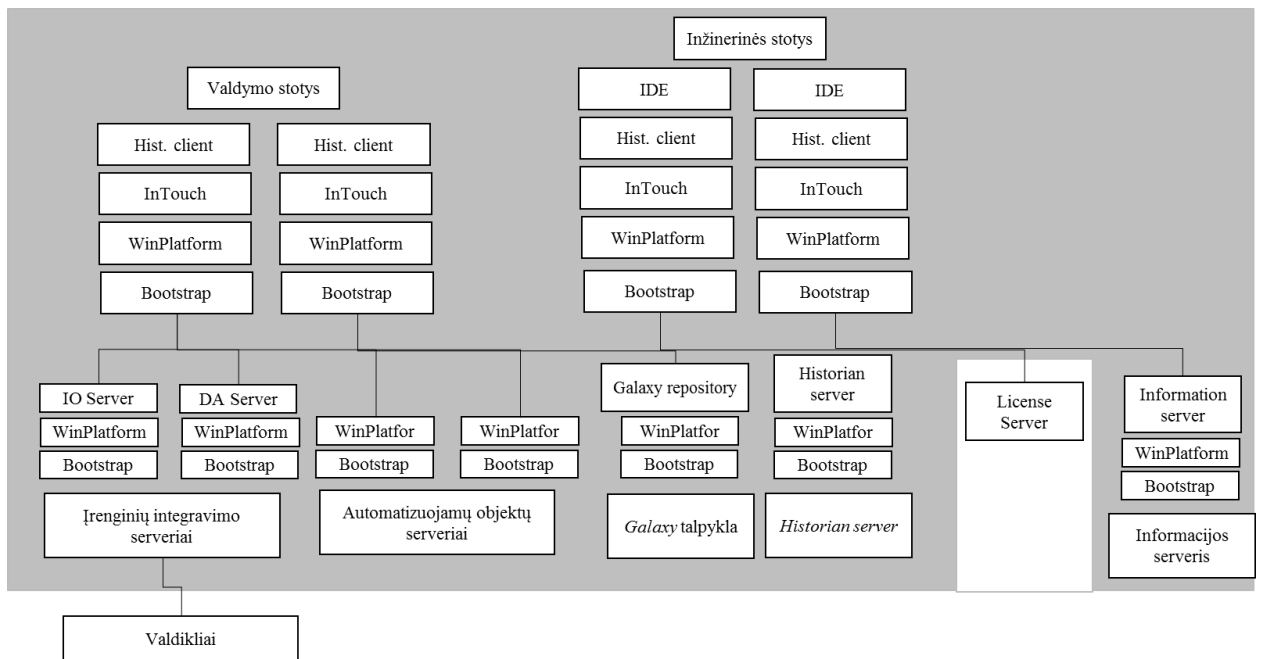
1.3 lentelė. Script'ų tipų palyginimas [3]

Eil. Nr.	InTouch script'o tipas	ArchestrA simbolio redaktoriaus script'as
1	Aplikacijos script'as	Nepalaikoma
2	Lango script'as	Symbol Predefined Script
3	Klavišų script'as	Klavišais aktyvinamas veiksmo script'as
4	Sąlygos script'as	Simbolio vardo script'as aktyvinamas komandomis kai tiesa (OnTrue), kai netiesa (OnFalse), kol tiesa (WhileTrue) ir kol netiesa (WhileFalse)
5	Duomenų apsikeitimo script'as	Simbolio vardo script'as su duomenų apsikeitimo aktyvikliu (angl. <i>trigger</i>)
6	Greitosios funkcijos	Nepalaikoma
7	ActiveX įvykio script'as	Nepalaikoma
8	Veiksmo script'as	Veiksmo script'o animacijos

1.3 lentelėje matyti, jog pagrindiniai skirtumai nurodyti 1, 6 ir 7 eilutėse. 1 eilutėje matyti, jog ArchestrA simbolio redaktorius neturi galimybės naudoti „Application Script“ funkcijos, tačiau rašant script‘ą kiekvienam simboliui atskirai, gaunamos analogiškos savybės. 6 eilutėje matyti, jog ArchestrA taip pat neturi komandos „QuickFunction“, tačiau šios analogiški veiksmai simboliui gali būti atlikti sukuriant poscript‘į(Named Scripts) ArchestrA script‘o redagavimo lange. 7 eilutėje matyti, jog ArchestrA simbolio redaktoriaus script‘e nėra komandos ActiveX Event Script.

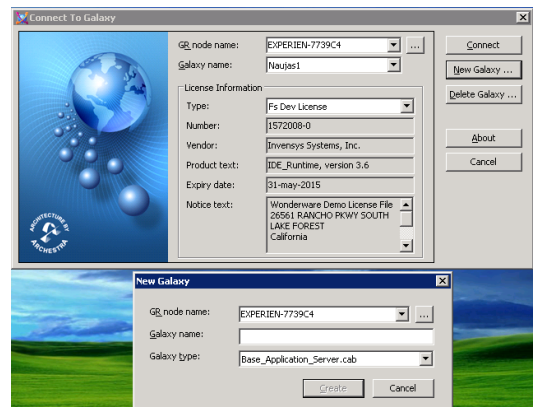
2. Aplikacijos kūrimas Archestra IDE

Galaxy yra SCADA aplikacija apimanti žmogus-mašina sąsajos kūrimą, istorinių duomenų serverį, įėjimų/išėjimų serverį ir kt. *Galaxy* kontroliuoja visų aplikacijos procesų darbą (žr. 2.1 pav. pilką foną), t.y. aplikacijų kūrimą, jų redagavimą, duomenų apsikeitimą su įėjimų/išėjimų įrenginiais, aliarmų registraciją, duomenų saugojimą ir pan. *Galaxy* sudaro valdymo stotys, inžinerinės stotys, informacijos serveris, *Historian server*, *Galaxy* talpykla, automatizuojamų objektų serveriai ir įrenginių integravimo serveriai. Kadangi visi aukščiau minėti procesai valdomi per *Galaxy*, visi jie turi būti patalpinti bendroje platformoje (*WinPlatform*). Inžinerinėse stotyse turi būti patalpinta tiek InTouch (WindowMaker ir WindowViewer), tiek ArchestrA IDE. Naudojantis šiais programiniais paketais kuriami ar koreguojami simboliai, o iš jų daromos aplikacijos, kurios talpinamos automatizuojamų objektų serveriuose. Įrenginių integravimo serveriais apsikeičiama duomenimis su išoriniais įrenginiais. Gauti duomenys per *Galaxy* talpinami *Historian server*. Gauti duomenys panaudojami istorijai, analizei ar aliarmams. Valdymo stotyse užtenka patalpinti įrankį WindowViewer. Šiuo įrankiu valdymo stotys gali išsikviesti aplikacijas iš automatizuojamų objektų serverių ir gauti bei siųsti duomenis į *Historian server*. Informacijos serveriu galima prisijungti prie aplikacijų kaip ir valdymo stotyse panaudojant WEB (žr. 2.1 pav.).



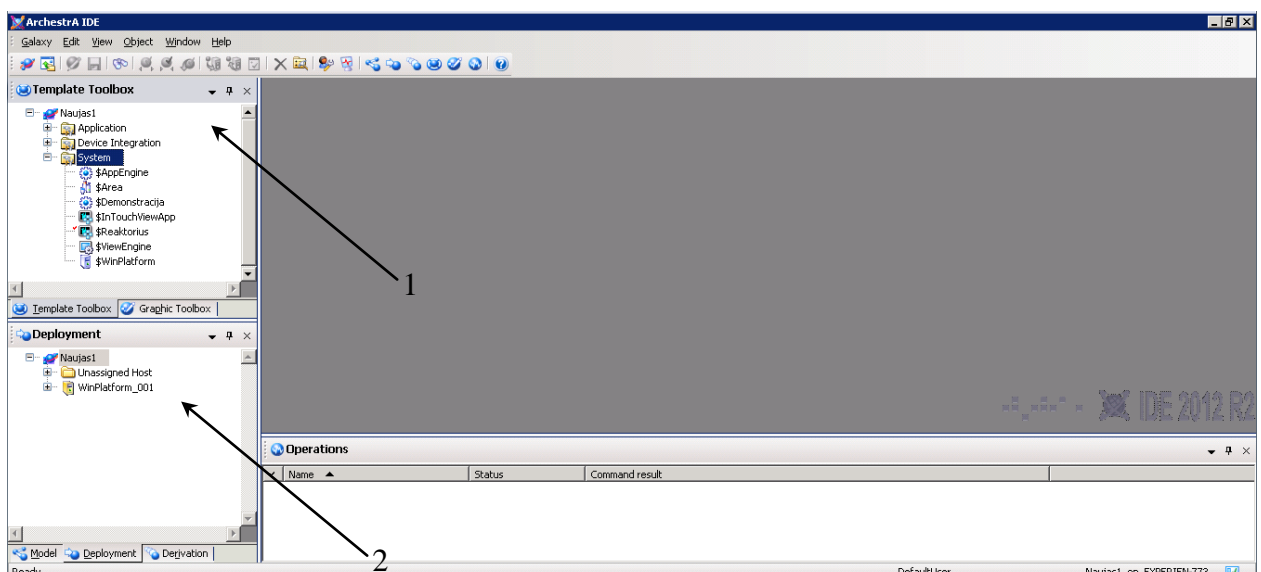
2.1 pav. Galaxy struktūra

Paleidus programinį paketą atveriamas aplikacijos pradinio konfigūravimo langas (2.2 pav.). Nuspaudus mygtuką „New Galaxy...“ atsivėrusiame lange nurodomas *Galaxy* vardas ir tipas „*Base_Application_Server.cab*“. Pasirinkus sukurtą *Galaxy* aplikaciją, prie jos prisijungiama paspaudus mygtuką „Connect“.



2.2 pav. Galaxy kūrimo langas

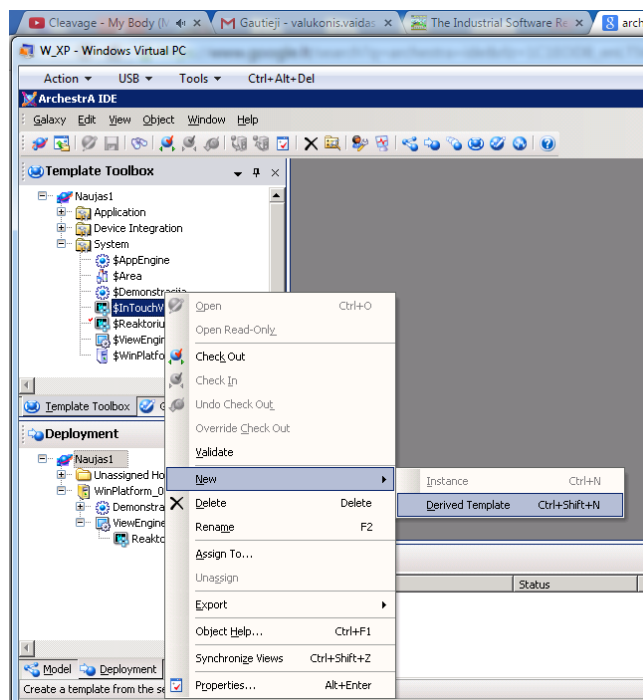
Prisijungus prie *Galaxy* patenkama į pagrindinį Archestra IDE langą, pavaizduotą 2.3 paveiksle:



2.3 pav. Pagrindinis Archestra IDE langas

2.3 paveiksle „1“ - pavaizduotas įrankių ruošinių laukas (ang. *Template Toolbox*). Įrankių lauke pateiktas InTouch aplikacijos ruošinys (ang. *IntouchViewApp*), aplikacijos variklio ruošinys (ang. *AppEngine*), srities (ang. *Area*) ir platformos (ang. *WinPlatform*) ruošiniai. Visi šie ruošiniai privalo būti panaudoti kuriant *Galaxy* aplikaciją. Grafinių įrankių ruošinių (ang. *Graphic Toolbox*) lauke pateikti standartiniai Archestra IDE simbolių ruošiniai.

2.3 paveiksle „2“ pažymėtas aplikacijos procesų išdėstymo (ang. *Deployment*) langas. Aplikacijos procesai taip pat gali būti atvaizduojami pagal proceso modelį (ang. *Model*) arba priklausomybę (ang. *Derivation*). Kortelėje **Model** formuojamas modelis, t.y. išdėstomi grafikos elementų ruošiniai taip suformuojant grafikos variklį su animacijoms skirtais elementais. Kortelėje **Deployment** suformuojama platforma, kurioje patalpinami variklio ir aplikacijos moduliai. Kortelėje **Derivation** procesus galima sudėlioti pagal priklausomybę vienas nuo kito.

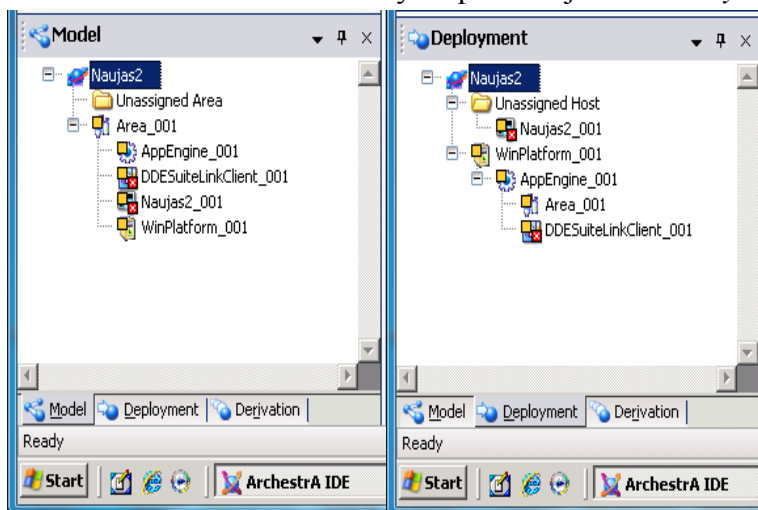


2.4 pav. Modelio sukūrimas.

Norint šablonus panaudoti kuriamoje aplikacijoje reikia atlikti keltą žingsnių.

Lange, kuris pavaizduotas 2.4 paveiksle reikalinga sukurti InTouch aplikacijos modelio ruošinį. Modelis sukuriamas ant ruošinio „\$InTouchViewApp“ spustelėjus dešiniu pelės klavišu ir pasirinkus *New/Derived Template*. Sukurtas ruošinys pavadinamas norimu vardu. Šiuo atveju tai bus „\$Reaktorius“.

Visi norimi ruošiniai yra panaudojami *Galaxy* aplikacijoje ant atitinkamo ruošinio



2.5 pav. Kortelės „Model“ ir „Deployment“.

spustelėjus dešinį pelės klavišą ir atsiradusiame lange suaktyvinant **New/Instance** komandą. Kuriant aplikaciją, joje reikia panaudoti šiuos ruošinius: „\$Reaktorius“, „\$Area“, „\$ViewEngine“ ir „\$AppEngine“. Norint jungtis prie valdiklio, dar reikalinga panaudoti „\$DDESuiteLinkClient“ ruošinį,

esantį „Device Integration“ kataloge. Visų minėtų ruošinių vykdymo kodas (*Instance*) turi atsirasti lange „Model“. „Imti ir mesti“ (ang. Drag-and-drop“) principu į „Area_001“ perkeliama AppEngine_001, DDESuiteLinkClient_001, WinPlatform_001 ir Reaktorius_001. Aplikacijos procesų išdėstymo lange „Deployment“ tuo pačiu „drag-and-drop“ principu hierarchine tvarka išrikiuojami WinPlatform_001, ViewEngine_001 ir Reaktorius_001. Atlikus minėtus veiksmus languose „Model“ ir „Deployment“ turėtų būti 2.5 paveiksle pateiktas vaizdas.

Modulis „Area_001“ naudojamas kaip erdvė, kurioje patalpinami kiti moduliai. Šis modulis gali turėti savo submodulius, jei to reikalauja atitinkami aliarmai ar valdymo algoritmai.

Modulis „AppEngine_001“ naudojamas objektų tvarkymui. Šis modulis, tai variklis objektų integracijai laike.

Modulis „WinPlatform_001“ yra pagrindinė bazė, kurioje stovi modeliuojamo objekto moduliai. Šis modulis skaičiuoja įvairius statistinius duomenis, juos pateikia aliarmams, istorijos serveriui, paleidinėja variklį („AppEngine“) [4].

Sukurta aplikacija turi būti nusiųsta (ang. *Deploy*) į vykdymo stotį. Joje operatorius gali naudotis ja proceso stebėjimui ir valdymui, tačiau negali prieiti prie jos koregavimo, kas apsaugo nuo išorinės intervencijos į sistemą.

2.1 InTouchViewApp šablonas

InTouchViewApp šablonas leidžia kurti InTouch aplikaciją Wonderware Application Server aplinkoje. Taip pat InTouchViewApp šablonas leidžia valdyti InTouch aplikacijų registraciją (angl. *check-in*), išregistravimą (angl. *check-out*) ir pateikimą (angl. *deployment*).

InTouchViewApp šablonas apima Application Server InTouchProxy objekto galimybes, tokiu būdu InTouch kintamieji (angl. *tags*) gali būti nurodyti kaip InTouchViewApp šablono atributai, kuriuos nurodo Application Server Message Exchange klientai.

InTouchViewApp šablonas naudojamas įtraukti InTouch aplikacijas į bendrą sistemą, kuri apibūdinama kaip Wonderware Application Server Galaxy. Application Server sistema valdo:

- Kelių vartotojų registravimą ir išregistravimą;
- Apsaugos nustatymus, t.y. kas gali pateikti InTouch aplikaciją ir kada tai gali padaryti;
- Atsargines kopijas ir duomenų atstatymą;
- Pateikimą (angl. *deployment*) ir atšaukimą(angl. *undeployment*).
- InTouchViewApp šablonas leidžia aplikacijos objektams veikti ViewEngine šablone tam, kad turėtų prieigą prie InTouch kintamųjų, bet tik tuo atveju, jei kintamieji yra sukurti kaip Application Server atributai. Kintamieji yra ieškomi Application Server atributų naršyklėje.

Sukurti InTouch aplikaciją su InTouchViewApp šablonu yra naudojamas \$InTouchViewApp šablonas. Kuriant aplikaciją yra du pasirinkimai – kurti naują aplikaciją arba kurti esamos aplikacijos kopiją. Abiem atvejais atveriant InTouch objektą paleidžiamas WindowMaker – InTouchViewApp objektas neturi savojo redaktoriaus.

Yra galimybė priskirti keletą InTouchViewApp objektų tam pačiam ViewEngine, taip sudarant keletą InTouch aplikacijų, veikiančių vienoje valdymo stotyje.

InTouch aplikacijos paleidimui, naudojant Application Server Platform, reikia:

- Sukurti ViewEngine objektą ir priskirti jį prie Platform;
- Sukurti InTouchViewApp šablono pavyzdį (angl. *Instance*) iš šablonų, kurie susiję su InTouch aplikacija;

- Susieti InTouchViewApp ir ViewEngine šablonus;
- Naudoti ViewEngine objektą kartu su jam priklausančiu InTouchViewApp objektu
- Paleisti InTouch Application Manager ir, pasirinkus aplikaciją, ją paleisti naudojant WindowView. Visi minėti žingsniai reikalingi tik kuriant aplikaciją pirmą kartą.

Kai kurie InTouchViewApp šablono ribojimai:

- InTouchViewApp pavyzdžiai negali būti kuriami iš InTouchViewApp pagrindinių šablonų;

- Tik vieno lygio InTouchViewApp pavyzdys gali būti sukurtas iš pagrindinio InTouchViewApp šablono;

- InTouchViewApp pavyzdys negali būti kuriamas iš egzistuojančio InTouchViewApp pavyzdžio;

- InTouchViewApp šablonas negali būti kitų šablonų šaltinis;

- Tik ViewEngine šablonai gali savyje turėti (angl. *host*) ir pateikti InTouchViewApp šablonus.

- Jei InTouchViewApp šablono aliarmai turi priklausyti minėtai erdvei (angl. *area*), lange **Model** InTouchViewApp šablonas gali būti susietas su Area šablonu.

- Jei InTouchViewApp šablonas yra užkrautas į valdymo stotį, nėra leidžiama jį atšaukti (angl *undeploy*).

- Kai failo kopijavimo procesas InTouchViewApp šablono kūrimo pabaigoje vis dar vyksta, vienintelė leidžiama operacija yra atšaukti (angl. *undeploy*) šabloną. Visa tai nutrauks išskleidimo operaciją ir pašalins visus dalinai nukopijuotus failus

Kai kuriamas InTouchViewApp šablonas, pasinaudojant esančia InTouch aplikacija, InTouch aplikacijos duomenys perkeliama į duomenų saugojimo direktoriją. Originalūs InTouch aplikacijos duomenys nėra modifikuojami ar ištrinami.

Į duomenų saugyklą negalima patekti tiesiogiai ar daryti prielaidas apie saugyklos struktūrą.

Kai InTouch aplikacijai daromi pakeitimai naudojant WindowMaker paleistą iš Archestra IDE, originali aplikacija nėra keičiama.

Kai InTouchViewApp šablonas yra sukurtas, ištrinus originalią InTouch aplikaciją, InTouchViewApp šablonas nebus paveiktas.

Jei InTouchViewApp šablonas susietas su egzistuojančia InTouch aplikacija yra ištrinamas, aplikacijos duomenys yra ištrinami iš saugyklos, tačiau egzistuojanti aplikacija nėra pažeidžiama.

Kai yra kuriamas InTouchViewApp šablonas naujai InTouch aplikacijai, WindowMaker turi būti paleistas naudojant Archestra IDE. Aplikacijos kūrimas vyksta taip pat, kaip WindowMaker būtų paleistas naudojant InTouch aplikacijų redaktorių.

2.2 ViewEngine šablonas

ViewEngine šablonas yra supaprastintas AppEngine šablonas. ViewEngine šablonas apima tik InTouchViewApp objektus. ViewEngine šablonas palaiko pagrindinius funkcija, tokius kaip: pateikimą (angl. *deployment*), atšaukimą (angl. *undeployment*), paleidimą (angl. *startup*) ir išjungimą (angl. *shutdown*). ViewEngine nepalaiko dubliavimo.

ViewEngine gali valdyti keletą InTouch šablonų.

ViewEngine šablonas:

- Yra susietas ir pateiktas su WinPlatform šablonu;
- Vykdo ir valdo InTouchViewApp šablonams – ViewEngine apklausimų dažnis (angl. *scan rate*) apibrėžia visų jo valdomų InTouchViewApp šablonų apklausimų dažnį.
- InTouchViewApp šablonas turi logiką, todėl, pateikus ir paleidus inicializaciją šablonas gali komunikuoti su kitais šablonais, naudojantis Galaxy;
- Neturi būti paleistas įrankyje WindowViewer tam, kad vykdyti script'us ar prieiti prie proceso duomenų;
- Gali valdyti Application Server *script*'ams ir UDAs. Tai leidži tėti procesus, kai WindowViewer yra išjungtas.
- Turi savo paties veikiančius diagnostikos atributus, kurie gali būti stebimi, suformuoti kaip aliarmai ir fiksuojami istorijoje.

2.3 Arcestra IDE simbolių apžvalga

Arcestra IDE simboliai yra grafiniai simboliai, kurie gali būti panaudoti vizualizuoti duomenis InTouch aplikacijose. Simboliai yra kuriami Arcestra IDE platformoje ir gali būti panaudoti tik InTouch aplikacijose. Application Server pateikiamas su Arcestra IDE simbolių biblioteka, kurią sudaro katalogai su iš anksto sukurtais ir sukonfigūruotais grafikos simboliais.

Naudojant Wonderware Application Server ir InTouch System Platform galima:

- Iš Arcestra simbolių bibliotekos įkelti sukurtus Arcestra simbolius į InTouch aplikacijos langą;

- Kurti naujus simbolius;
- Arcestra simbolius integruoti į Arcestra šablonus.

Simbolių kūrimui naudojamas simbolių redaktoriaus langas, kuriame simboliai kuriami iš pagrindinių grafinių elementų, tokių kaip : stačiakampis, linija ar teksto elementai.

Sukurti Arcestra simboliai gali būti panaudoti kitame simbolyje arba InTouch aplikacijos lange.

Pasinaudojant Arcestra IDE galima:

- Kurti naujus simbolius;
- Koreguoti simbolius, pasinaudojant Arcestra IDE simbolių redaktoriumi;
- Sisteminti simbolius, naudojant **Graphic Toolbox**;
- Dubliuoti simbolius;
- Importuoti ir eksportuoti simbolius;
- Ištrinti simbolius;
- Konfigūruoti simbolių operacijų saugumą;
- Atidaryti simbolius skaitymo (angl. *read-only*) režimu.

Priklausomai nuo projektavimo reikalavimų galima pasirinkti, kur ir kaip simboliai turi būti talpinami.

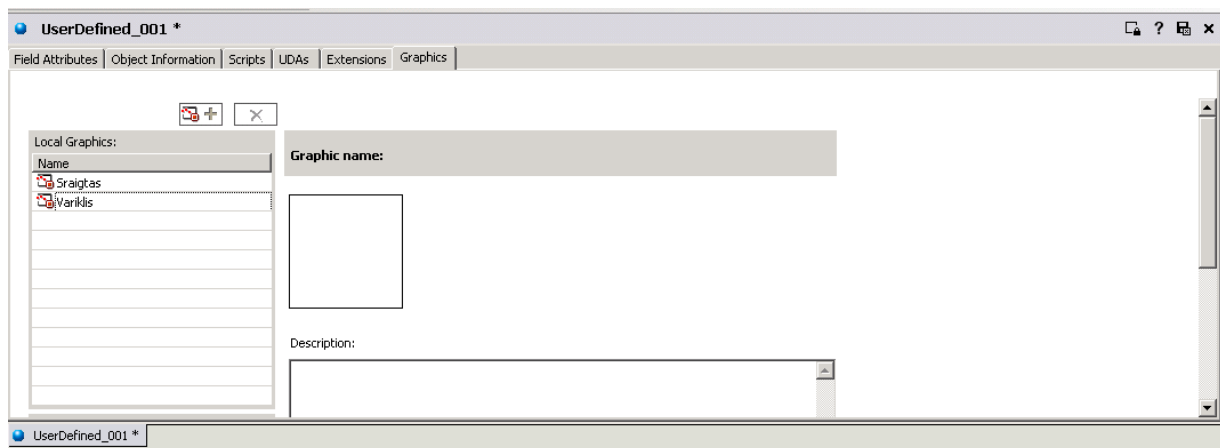
- Simboliai gali būti talpinami Grapihs Toolbox, apibūdinant juos kaip standartinius simbolius, kurie gali būti panaudoti kitam simboliui kurti. Taip pat taip patalpinti simboliai gali būti naudojami InTouch HMI;

- Simbolį patalpinus kaip AutomationObject modelį, simbolis gali būti naudojamas kaip pavyzdys (angl. *instance*) daugybę kartų.

- Simbolį patalpinus kaip AutomationObject pavyzdį, simbolis naudojamas tik vienam šablonui, kuriam jis buvo sukurtas.

Sukūrus Archestra IDE simbolį AutomationObject lange yra **Graphics** kortelė, kurioje Archestra simboliai gali būti kuriami, koreguojami, pervadinami ar ištrinami. Šie simboliai atsiranda **Local Graphics** laukelyje

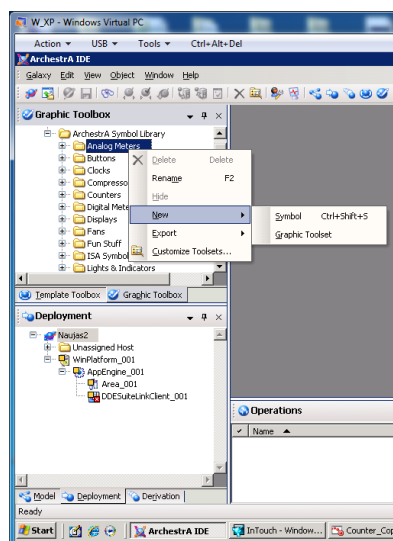
Atlikus bet kokius pakeitimus simboliuose, kilusiuose iš AutomationObject, šie simboliai atnaujinami visose aplikacijose, kuriose buvo patalpintas simbolio pavyzdys.



2.6 pav. AutomationObject lango Graphics kortelė

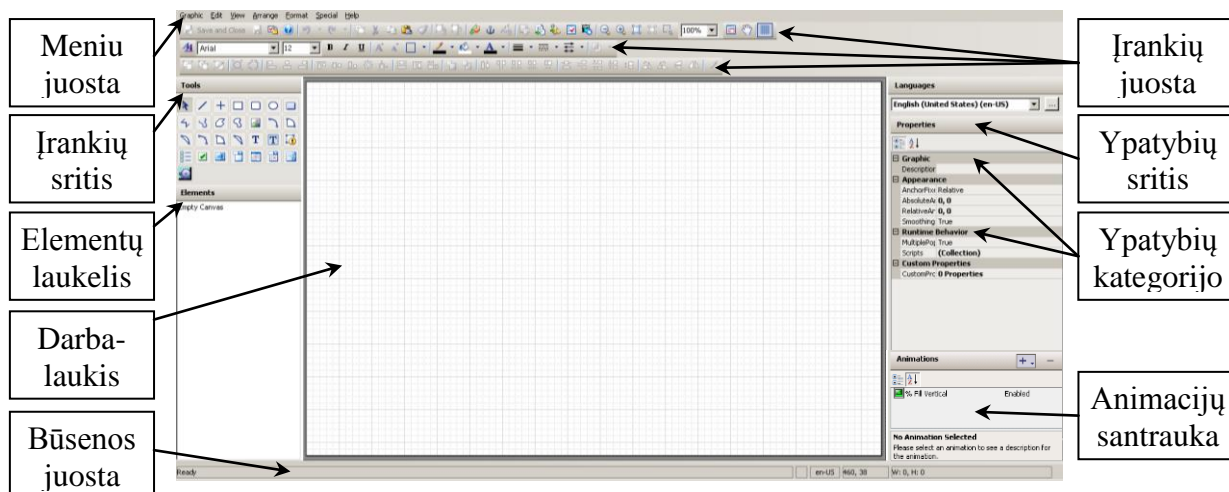
2.4 Grafinės sąsajos elementų kūrimas

Vartotojo kuriami simboliai yra talpinami naujame Archestra IDE „Graphic Toolbox“ bibliotekos aplanke. Naujas aplankas sukuriama dešiniu pelės klavišu spustelėjus ant „Archestra Symbol Library“ ir pasirinkus *New/Graphic Toolset*. Naujo simbolio kūrimo kelias identiškas, tik ant naujai sukurto aplanko pasirenkama *New/Symbol* komanda (2.7 pav.)



2.7 pav. Simbolio kūrimas

Du kartus paspauduskairįjį pelės klavišą ant naujai sukurto simbolio patenkama į simbolių kūrimo langą, pavaizduotą 2.8 paveiksle.



2.8 pav. Simbolių kūrimo langas

ArchestrA IDE simbolių redaktorius apima šias sritis:

Įrankių juosta–apima pagrindinės funkcines komandas, įtraukiant tokias kaip iš anksto sukonfigūruotų simbolių įkėlimą iš ArchestrA IDEsimbolių bibliotekos;

Įrankių sritis – pagrindiniai grafiniai įrankiai, skirti kurti simbolius;

Elementų laukelis – nurodomi automatiškai formuojami kuriamų elementų pavadinimai;

Darbalaukis – tinklę turinti erdvė, kurioje formuojami simboliai iš elementų;

Būsenos juosta – pateikiama informacija apie pažymėtą elementą;

Ypatybių sritis – pateikiama pažymėtam elementui ar darbalaukyje esančiam ArchestrA IDEsimboliui priklausantis ypatybių sąrašas (elementai gali būti koreguojami šioje srityje arba mygtukais iš įrankių juostos);

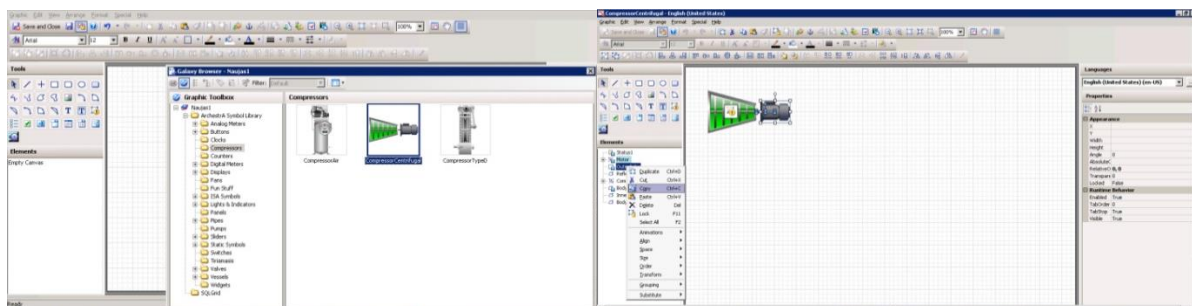
Animacijos santrauka – nurodomos animacijos, priklausančios pažymėtam simboliui ar elementui.

Iš ArchestrA IDEsimbolių bibliotekos įkelta grafika gali būti koreguojama darbalaukyje, pažymint atskirus simbolio elementus ir naudojant įrankius iš įrankių juostos. Paprastai naudojami įrankiai iš įrankių srities yra linija, kvadratas, kreivė, 2-jų taškų arka, paveikslėlio įterpimas, laikrodis, kalendorius ir pan. Naudojanti lauku **Ypatybių sritis** elementams gali būti priskiriamos koordinatės, pradžios koordinatės, pasukimo kampas ir pan. Baigus brėžti simbolį, jam priskiriama animacijos, kintamieji ir kt.

2.4.1 Šablono „Sraigtas“ kūrimas

Sraigto simbolio kūrimui galima panaudoti bibliotekoje esančius simbolius arba jų dalis.

Pagal 2.4 skyriuje pateiktą aprašymą sukuriama naujas simbolio „Sraigtas“ šablonas ir atidaromas simbolio kūrimo langas. Iš simbolių bibliotekos **Graphic Toolbox** atveriamas simbolis *Archestra Symbol Library\Compressors\CompressorsCentrafugal*. Elementų laukelyje (**Elements**) pažymimi ir nukopijuojami elementai **Motor** ir **OuterAxle** (2.9 pav.).

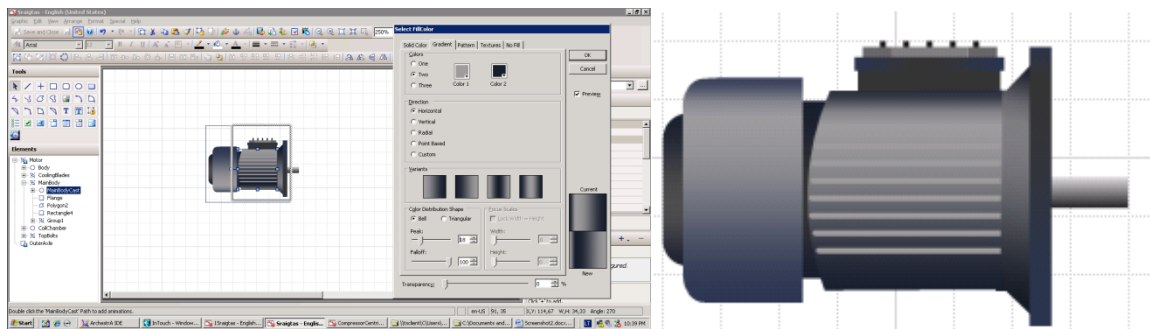


2.9 pav. Simbolio dalies nukopijavimas

Nukopijuota simbolio dalis (variklis su ašimi) įterpiama į naujai kuriamo simbolio kūrimo lange. Laukelyje **Elements** pažymimas elementas **Motor**, kuriam laukelyje **Properties** nurodomos koordinatės (X:89,Y:60), Angle (180); atitinkamai elementui **OuterAxle** – (X:125,Y82). Laukelyje **Elements** išskleidus elementus **Motor** ir **MainBody** pažymimi **Rectangle1**, **Rectangle6**, **Rectangle5**, **Rectangle3** ir **Rectangle8** elementai. Nuspaudus klavišą F3, pastarieji elementai sugrupuojami į vieną elementą **Group1**. Naujai sukurtas elementas pasukamas 180°, jo Y koordinatė pakeičiama į 60, o pats elementas patalpinama už likusių elementų ant jo nuspaudus dešinįjį pelės klavišą ir pasirinkus komandą *Order\Send To Back*. Pažymėjus elementą **TopBolts** jo Y koordinatė pakeičiama į 58.

Kitu žingsniu panaikinamos elementų animacijos. Du kartus paspaudus ant simbolio atsivėrusiame lange spaudžiama „–“. Šios procedūros atliekamos elementams **Motor** ir **OuterAxle**.

Elemento *MainBody\MainBodyCast* spalvos keičiamos pažymint elementą ir naudojant komandą *Fill Collor\More Gradients*. Atsivėrusiame lange laukelyje **Variants** pasirenkamas antrasis spalvų pasiskirstymo variantas ir laukelyje **Color Distribution Shape** įrašoma senoji **Peak** reikšmė, 18 (2.10 pav.). Atitinkamai pakeičiamos elementų **Body**, **Flange** ir **Polygon2** spalvos. Sukurtas simbolis pavaizduotas 2.10 paveiksle.



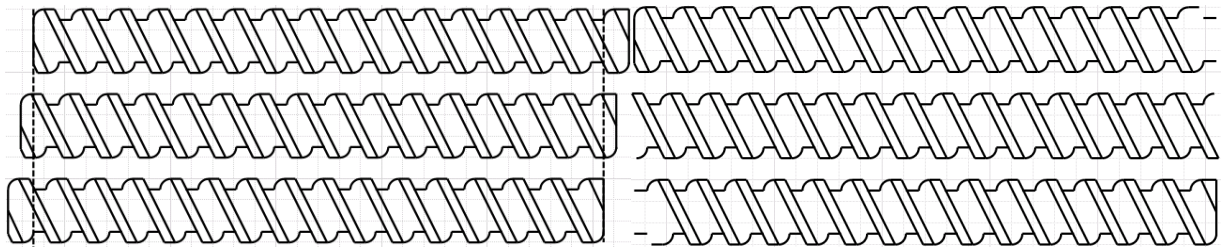
2.10 pav. Elemento atspalvių keitimas

Toliau linijos brėžimo įrankiu nubrėžiama bet kokia linija. Laukelyje **Properties** parenkamos linijos koordinatės X: 140,Y:120 , plotis (**Width**) 15 ir aukštis (**Height**) 30. Minėtas elementas dubliuojamas nuspaudus klavišų kombinaciją CTRL+D. Naujojo elemento koordinatės parenkamos X:145, Y:120. Nubrėžiama kita linija, parenkamos koordinatės X:140, Y:120, plotis 5 ir aukštis 0. Minėtas elementas dubliuojamas ir jam parenkamos koordinatės X:155, Y:150. Minėtas elementas vėl dubliuojamas ir jam parenkamos koordinatės X:147, Y:146, plotis 6 ir aukštis 0. Tada 2 taškų arkos (**2 Point Arc**) brėžimo įrankiu nubrėžiama bet kokia arka. Jai parenkamos koordinatės X:153, Y:120, plotis 5, aukštis 5, pradinis kampas (**StartAngle**) 0 ir judėjimo kampas (**SweepAngle**) -90. Elementai **Line5** ir **Arc1** sugrupuojami ir dubliuojami. Pažymimas elementas **Group2** ir jam parenkamos koordinatės 160:145 ir kampas 180°. Elementai **Group1** ir **Group2** išgrupuojami klaviatūros klavišais SHIFT+F3. Visi naujai nubrėžti elementai pažymimi (išskyrus **Motor** ir **OuterAxle**) ir sugrupuojami. Naujai sugrupuotas elementas dubliuojamas ir nustatomos koordinatės X:158, Y:120. Toliau nepanaikinus elemento žymėjimo naudojant dubliavimo komandą dubliavimo komanda padaroma 15 elemento dublikatų. Visi naujai dubliuoti elementai išgrupuojami. Brėžiama bet kokia linija, kurios koordinatės X:417, Y:120, aukštis 26. Brėžiama kita linija, kurios koordinatės :, plotis 7. Brėžiama dar viena linija, kurios koordinatės: plotis 7 ir aukštis 15. Ištrinamas galinis trumpasis horizontalus elementas, išsikišęs už naujai formuojamo simbolio matmenų. Toliau dubliuojamas elementas **Arc2** ir jam nustatomos koordinatės X:135, Y:120. Pažymimi elementai **Line4**, **Line5** ir **Arc1** ir sugrupuojami. Gautas elementas dubliuojamas ir nustatomos koordinatės X:137, Y:145. Sudaryti elementai išgrupuojami. Nubrėžiama linija, parenkamos koordinatės X:135, Y:135, plotis 7 ir aukštis 15. Brėžiama kita linija, parenkamos koordinatės X:135, Y:146, plotis 2 ir aukštis 4. Brėžiama dar viena linija ir jai nustatomos koordinatės X:135, Y:125, aukštis 21. Pažymėjus visus pastaruosius nubrėžtus simbolius, jie sugrupuojami. Atlikus visus minėtus veiksmus gaunamas simbolis, pavaizduotas **Error! Reference source not found.** paveiksle.



2.11pav. Sraigtas

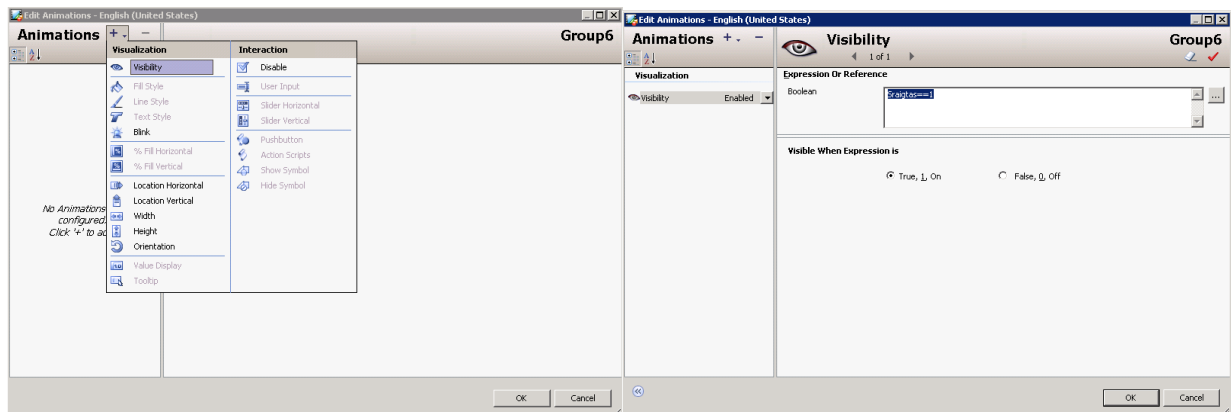
Sugrupuotas elementas dubliuojamas du kartus. Vienam jų priskiriamos koordinatės X:129, Y:160, kitam – X:123, Y:200. Naujieji elementai išgrupuojami ir ištrinami jų elementai esantys už punktyrinės linijos (2.12 pav.). Ištrinus elementus elementai turėtų atrodyti kaip pavaizduota 2.12 paveiksle dešinėje.



2.12pav. Elementai

Toliau užpildomos trūkstamos formuojamų elementų dalys. Nubrėžiama linija ir jai priskiriamos koordinatės X:105, Y:205 ir aukštis 20. Brėžiama kita linija, kurios koordinatės X:105, Y:205, plotis 13 ir aukštis 25. Brėžiama dar viena linija, kurios koordinatės X:105, Y:214, plotis 8 ir aukštis 16. Nubrėžtos trys linijos sugrupuojamos ir padaromas dublikatas. Vieno dublikato koordinatės yra X:392, Y:120, kampas 180° , o kito koordinatės X:135, Y:205, kampas 0° . Toliau brėžiamos dvi linijos, kurių duomenys yra X:135, Y:160, aukštis 30, kitos – X:135, Y:160, plotis 2. Nubrėžtos linijos sugrupuojamos. Sugrupuotas elementas dubliuojamas, dublikatui priskiriamos koordinatės X:404, Y:160, kampas 180° . Gauti trys elementų rinkiniai, kurių kiekvienas atvaizduoja sraigo padėti skirtingu pasisukimo kampu. Kiekvienas iš jų atskirai sugrupuojamas. Taip gaunamos trys atskiros elementų grupės.

Kiekvienai elementų grupei du kartus paspaudus ant grupės atsivėrusiame **Edit Animations** lange priskiriamos animacijos. Animacijų meniu pasirinkus **Visibility** pasirinktį, laukelyje **Boolean** įrašoma „Sraigtas==0“. Animacijos priskyrimas elementų grupei pavaizduota 2.13 pav.

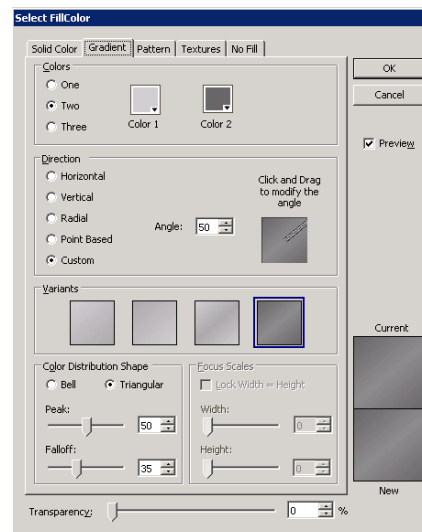


2.13 pav. Elementų animacijos kūrimas

Kitoms elementų grupėms atitinkamai į laukelį **Boolean** įrašoma „Sraigtas==1“ ir „Sraigtas==2“.

Toliau visų elementų grupių Y koordinatė pakeičiama į 120 (X koordinatė nekeičiama). Pažymimos visos trys elementų grupės ir jos dar kartą sugrupuojamos klaviatūros klavišo F3 pagalba. Bendrai sugrupuoto elemento koordinatės pakeičiamos į X:164, Y:70. Taip sraigtas prijungimas prie variklio.

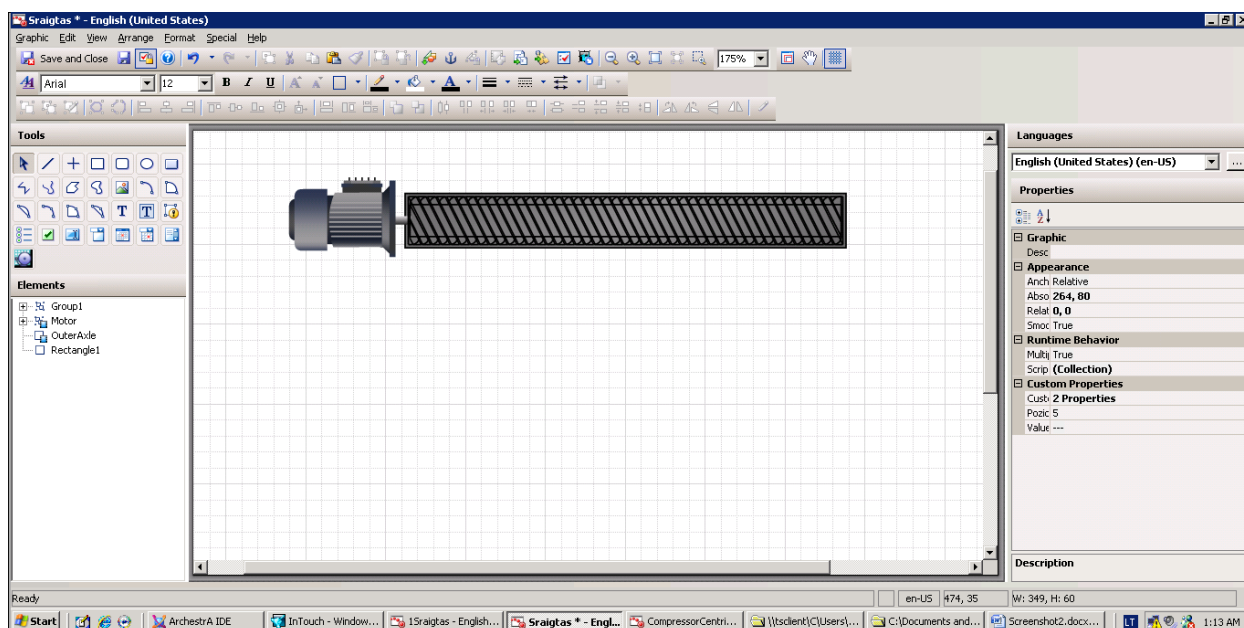
Kuriant sraigto korpusą, brėžiamas bet koks stačiakampis įrankių juostoje pasirinktu brėžimo įrankiu **Rectangle**. Jį pažymėjus pasirenkamas elemento spalvos keitimo įrankis **Fill Color** ir jo funkcija **More Gradients...**. Atsivėrusio lango laukelyje **Colors** pasirenkama **Two**. Pirmosios spalvos komponentai: Red:204, Green:204, Blue:204, Hue:0, Sat:0, Lum:204. Antrosios spalvos komponentai: Red:102, Green: 102, Blue: 102, Hue:0, Sat:0, Lum: 102. Laukelyje **Direction** parenkama pasirinkti **Custom** ir nustatomas kampas **Angle** 50. Laukelyje **Variants** pasirenkamas ketvirtas iš eilės fono užpildymo būdas.



2.14 pav. Spalvos parinkimo nustatymai

Laukelyje **Color Distribution Shape** parenkamai sekantys nustatymai: **Triangular**, Peak:50, Falloff:35. Atlikus minėtus veiksmus nustatymai turėtų atrodyti kaip pavaizduota 2.14 paveiksle.[5]

Ant sukurto elemento paspaudus dešiniu pelės klavišu parenkama **Order|Send To Back**. Elementui nustatomos koordinatės X:162, Y:68, plotis 276, aukštis 34. Atlikus minėtas procedūras gaunamas 2.15 paveiksle pavaizduotas simbolis.



2.15 pav. Nubraižytassimbolis

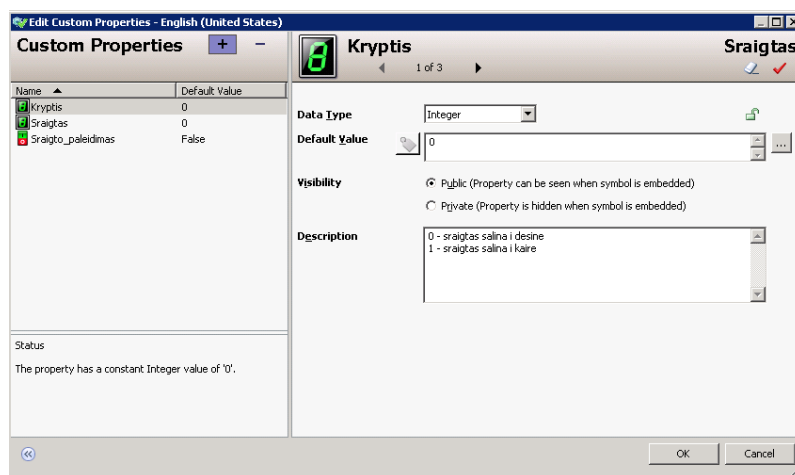
Sukurtam simboliui sukuriama paprogramė (toliau – script‘as). Script‘o rašymo langas išskviečiamas komanda *Special\Scripts...* . Atsivėrusiame lange parenkamas tipas (**type**) **While Showing** ir script‘o iškvietimo laikas (**Every**) 100ms. Žemiau pateiktas script‘o kodas.

```

ifSraigto_paleidimas == truethen
ifKryptis == 0 then
  Sraigtas=Sraigtas+1;
  ifSraigtas>= 3 then
    Sraigtas=0;
  endif;
else
  Sraigtas=Sraigtas-1;
  ifSraigtas<= -1 then
    Sraigtas=2;
  endif;
endif;
endif;

```

Toliau aprašomi kintamieji, kurie naudojami script‘e ir bus prieinami, kai simbolis bus panaudotas InTouch aplinkoje. Komanda *Special\Custom Properties...* atveriamas langas skirtas aprašyti kintamiesiems. Nuspaudus mygtuką „+“ atsiranda naujas kintamasis, kuris pavadinamas „Kryptis“. Laukelyje **Data Type** parenkamas kintamojo tipas **Integer**, pradinė reikšmė (**Default Value**) 0, matomumas InTouch aplinkoje (**Visibility**) **Public** ir trumpas kintamojo reikšmių aprašymas laukelyje **Description** („0 - sraigtas salina i desine 1 - sraigtas salina i kaire“). Vadovaujantis tomis pačiomis procedūromis sukuriamas simbolis „Sraigtas“ (Integer, 0, Private) ir „Sraigto_paleidimas“ (**Boolean**, *False*, **Public**). Atlikus kintamųjų aprašymas, langas turi atrodyti kaip parodyta 2.16 paveiksle.



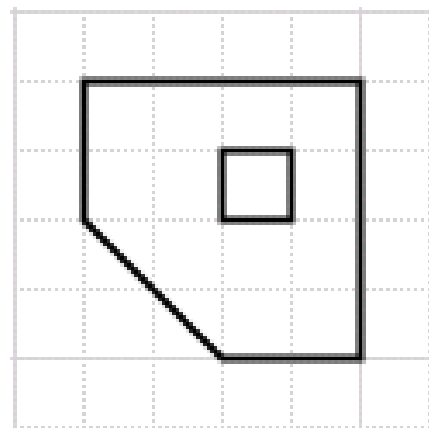
2.16 pav. Kintamųjų aprašymo langas

Simbolių matomumui nustatyti yra dvi galimybės: **Public** ir **Private**. Naudojant pasirinktą **Public**, leidžia kintamąjį naudoti kitoms grandims. Tai reikalinga, kai norima simbolį surišti su script‘u ar elementais už simbolio ribų. Pasirinktis **Private** naudojama, kai reikalinga aprašyti vidiniu simbolio kintamuosius, naudojamus script‘e ar pan.

2.4.2 Šablono „Kranas“ kūrimas

Pagal 2.4 skyriaus aprašymą sukuriamas naujas simbolio šablonaspavadinimu „Kranas“.

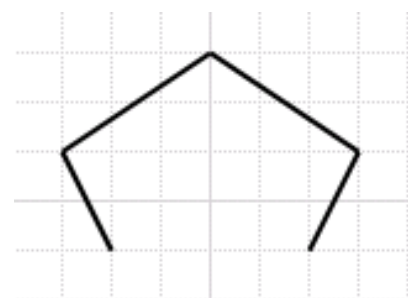
Vadovaujantis 2.4.1 skyriumi brėžiama linija, kurios koordinatės X:0, Y:0, plotis (*Width*) 230 ir aukštis (*Height*) 0. Minėtas elementas 2 kartus dubliuojamas klaviatūros klavišų CTRL+D kombinacija. Dubliuotų elementų koordinatės X:0, Y:5 ir X:0, Y:10. Panaudojant brėžimo įrankį **PolyLineir Rectangle** nubrėžiama 2.17 paveiksle pavaizduota figūra, vaizduojanti kraną kabiną. Panaudojant nuorodą **Fill Color|More Solid Colors...** elementui **Rectangle1** pakeičiama spalva. Keičiant spalvą priskiriami komponentai: Red:131, Green:131, Blue:255, Hue:170, Sat:255, Lum:193. Pažymimas



2.17 pav. Kranas kabinas

elementas (**PolyLine1**) ir paspaudus dešiniu pelės klavišu ant simbolio, pritaikoma komanda **Path|Combine**. Pastarąjį elementą apibrėžiamas kaip vienas elementas, kuriam galima pritaikyti fono spalvos pakeitimus. Naudojant nuorodą **Fill Color|More Gradients...** elementui **Path1** pakeičiama fono spalva. Laukelyje **Colors** parenkamas parametras **One, Direction – Custom, Variants** – pirmas simbolis iš kairės, **Color Distribution Shape – Triangular**. Elementai **Rectangle1** ir **Path1** sugrupuojami juos pažymint ir nuspaudus klavišą F3. Sukurtai grupei priskiriama animacija **Location Horizontal** su parametrais: **Analog – Kranas_X, At Left End – 0, At Right End – 100, To Left – 0, To Right – 170**. Brėžiama linija, kuriai priskiriami tokie parametrai - koordinatės X:40, Y:50, plotis 0, aukštis 100, pradžios koordinatės (**Start**) 40:50, pabaigos koordinatės, (**End**) 40:150. Jai priskiriama animacijos **Location Horizontal** ir **Height**. Animacijai **Location Horizontal** priskiriami parametrai kaip ir prieš tai aprašyti grupei. Animacijai **Height** priskiriami parametrai yra: **Analog – Lyno_ilgis, At Min Size – 0, At Max Size – 100, Min% – 30, Max% – 100, Anchor - Top**.

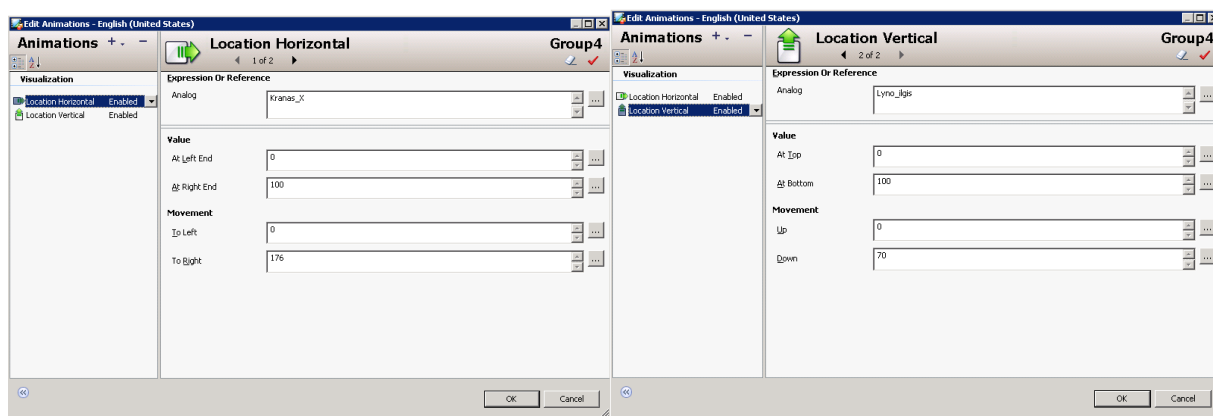
Nubrėžiama linija, kurios koordinatės X:10, Y:80, plotis 30, aukštis 20, pradžios koordinatės 40:80, pabaigos koordinatės 10:100. Brėžiama kita linija, kurios koordinatės X:10, Y:100, plotis 10, aukštis 20, pradžios koordinatės 10:100, pabaigos koordinatės 20:120. Nubrėžtos linijos sugrupuojamos. Šalimais nubrėžiamos kitos dvi linijos sudarant veidrodinį variantą Y ašies atžvilgiu. Pastarosios linijos taip pat sugrupuojamos. Gautas vaizdas turi atitikti 2.19 paveiksle pavaizduotą figūrą. Ant kairiosios žnyplių pusės 2



2.18 pav. Žnyplės

kartus paspaudus kairiuoju pelės klavišu ant elementų grupės išskviečiamas animacijų priskyrimo

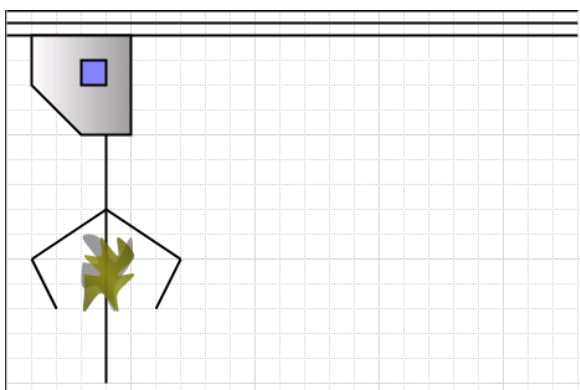
langas. Nuspaudus mygtuką „+“ priskiriama animacija **Orientation** ir žemiau išvardinti parametrai: **Analog** – *Znyples_K*, **At CCW End** – 10, **At CW End** – 0, **CCW** – 15, **CW** – 0, **Use Current as Offset Orientation** – 0, **Current Realtive Origin** – dX:15, dY:-20. Dešiniajai žnyplių pusei priskiriama tokia pati animacija su parametrais: **Analog** – *Znyples_D*, **At CCW End** – 0, **At CW End** – 10, **CCW** – 0, **CW** – 15, **Use Current as Offset Orientation** – 0, **Current Realtive Origin** – dX:-15, dY:-20. Abi grupės dar kartą sugrupuojamos gaunant vieną elementą. Naujai sugrupuotam elementui priskiriamos animacijos **Location Horizontal** ir **Location Vertical**. Animacijoms priskiriami kintamieji ir parametrai kaip parodyta 2.19paveiksle.



2.19 pav. Žnyplių judėjimo parametrai

Panaudojant brėžimo įrankį **Curve** nubrėžiama netaisyklinga figūra. Panaudojus nuorodą **Path\Combine** figūra apjungiama į vientisą. Pakeičiamas figūros fonas taip, kad figūra primintų nešamą kurą. Jai priskiriamos animacijos **Location Horizontal**, **Location Vertical** ir **Visibility**. Animacijoms **Location Horizontal** ir **Location Vertical** priskiriami parametrai pavaizduoti 2.19 paveiksle. Animacijai **Visibility** laukelyje **Boolean** priskiriamas kintamasis „Kuras“.

Atlikus minėtus veiksmus gaunamas simbolis pavaizduotas 2.20 paveiksle.



2.20 pav. Krano simbolis

Iškvietus langą **Custom Properties** sukuriami kintamieji su parametrais pagal 2.1 lentelę.

2.1 lentelė. Simbolio kintamųjų aprašymo lentelė

Kintamasis	Kintamojo tipas (Data Type)	Pradinė reikšmė (Default Value)		Matomumas (Visibility)
Ciklas	Integer	0		Private
Greitis	Integer	5		Public
Kranas	Boolean	False		Public
Kranas_X	Integer	0		Private
Kuras	Boolean	False		Private
Lyno_ilgis	Boolean	False		Private
Znyples_D	Boolean	False		Private
Znyples_K	Boolean	False		Private

Iškvietus script'o rašymo langą parametras **Trigger type** parenkamas **While Showing**, **Every:1000ms** ir įkeliamas žemiau pateiktas script'as.

```

ifKranas == truethen
  IfCiklas == 0 then
    ifLyno_ilgis < 100 then
      Lyno_ilgis = Lyno_ilgis + 5 * Greitis;
    else
      ifLyno_ilgis > 100 then
        Lyno_ilgis = 100;
      endif;
      Ciklas = 1;
    endif; endif;
    IfCiklas == 1 then
      IfZnyples_K < 10 then
        Znyples_K = Znyples_K + 0.5 * Greitis;
        Znyples_D = Znyples_D + 0.5 * Greitis;
      else
        ifZnyples_K > 10 then
          Znyples_K = 10;
          Znyples_D = 10;
        endif;
        Ciklas = 2;
        Kuras = true;
      endif;
    endif;
    ifCiklas == 2 then
      ifLyno_ilgis > 0 then
        Lyno_ilgis = Lyno_ilgis - 5 * Greitis;
      else
        ifLyno_ilgis < 0 then
          Lyno_ilgis = 0;
        endif;
        Ciklas = 3;
      endif;
    endif;
    ifCiklas == 3 then
      ifKranas_X < 100 then
        Kranas_X = Kranas_x + 5 * Greitis;
      else
        ifKranas_X > 100 then
          Kranas_X = 100;
        endif;
        Ciklas = 4;
      endif;
    endif;
  endif;

```

```

ifCiklas == 4 then
ifLyno_ilgis<= 100 then
    Lyno_ilgis = Lyno_ilgis + 5 * Greitis;
else
    ifLyno_ilgis> 100 then
        Lyno_ilgis = 100;
    endif;
    Ciklas = 5;
endif;
endif;
ifCiklas == 5 then
ifZnyples_K> 0 then
    Znyples_K = Znyples_K - 0.5 * Greitis;
    Znyples_D = Znyples_D - 0.5 * Greitis;
else
    ifZnyples_K< 0 then
        Znyples_K = 0;
        Znyples_D = 0;
    endif;
    Ciklas = 6;
    Kuras = false;
endif;
endif;
ifCiklas == 6 then
ifLyno_ilgis> 0 then
    Lyno_ilgis = Lyno_ilgis - 5 * Greitis;
else
    ifLyno_ilgis< 0 then
        Lyno_ilgis = 0;
    endif;
    Ciklas = 7;
endif;
endif;
ifCiklas == 7 then
ifKranas_X> 0 then
    Kranas_X = Kranas_x - 5 * Greitis;
else
    ifKranas_X< 0 then
        Kranas_X = 0;
    endif;
    Ciklas = 0;
endif;
endif;
endif;

```

2.4.3 Šablono „Talpa“ kūrimas

Simbolių bibliotekoje sukuriamas naujas simbolio šablonas ir pavadinamas „Talpa“. Brėžiamos trys linijos. Pirmosios linijos pradžios koordinatės (*Start*) 0:0 ir pabaigos koordinatės (*End*) 36:120, antrosios *Start* 36:120, *End* 14:120, trečiosios *Start* 144:120, *End* 180:0. Pažymėjus visus tris elementus naudojantis komanda **Arrange|Path|Combine** elementai apjungiami į vientisą elementą.

Sukurtam elementui priskiriama fonas spalvos keitimo įrankiu **Fill Color** ir jo komanda **More Gradients...**. Atsivėrusio lango laukelyje **Colors** pasirenkama **Two. Color 1** komponentai: Red:77, Green: 77, Blue: 77, Hue:0, Sat:0, Lum: 77. **Color 2** komponentai:

Red:239, Green: 239, Blue:239, Hue:0, Sat:0, Lum: 239.

Laukelyje **Direction** parenkama pasirinkti **Vertical**.

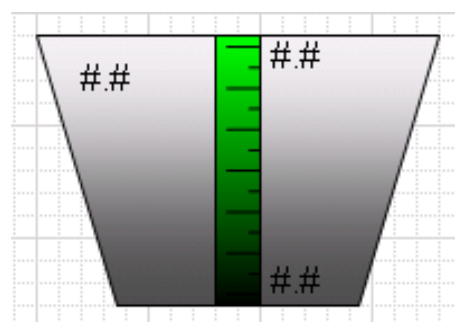
Laukelyje **Variants** pasirenkamas antras iš kairės fono

užpildymo būdas. Laukelyje **Color Distribution Shape**

parenkamai sekantys nustatymai: **Bell**, **Peak:100**,

Falloff:100. Toliau brėžiamas kvadratas, kurio koordinatės

X:80, Y:0, plotis (*Width*) 20 ir aukštis (*Height*) 120. Du



2.21 pav. Talpa

kartus spragtelėjus kairiu pelės klavišu ant elemento, mygtuku „+“ priskiriama animacija „% Fill

Vertical“. Priskirtos animacijos parametrai yra: **Analog** – *Talpa*, **At Min Fill** – *Min*, **At Max**

Fill – *Max*, **Min%** - 0, **Max%** - 80, **Direction** – Up, **Orientation** – Relative to Graphic.

Pridedama kita animacija **Fill Style** (žr. 2.21 pav.). Parenkamos parametrų reikšmės: **States** –

Truth Table, **Expression Or Reference** – *String* ir įrašomas kintamasis *Indikatoriaus_spalva*,

Element Fill Style – žalia spalva. Laukelyje **Truth Table** – mygtuku „+“ sukuriami iš viso trys

laukai, stulpelyje **Color** atitinkamai parenkamos žalia, geltona ir mėlyna spalvos, stulpelyje

Operator – parenkamas „=“ ženklas, **Value or Expression** – atitinkamai *Zalia*, *Geltona*, *Melyna*.

Kiekvienoje eilutėje stulpelyje **Value or Expression** simbolis  pakeičiamas į simbolį .

Brėžiama tiesė, kurios *Start* 85:5, *End* 100:5. Klavišais CTRL+D tiesė dubliuojama. Dubliuotos

tiesės *Start* 85:23, *End* 100:23. Dubliuota tiesė dubliuojama dar 5 kartus. Brėžiama linija, kurios

Start 95:14, *End* 100:14. Minėta tiesė dubliuojama ir jai priskiriama *Start* 95:32, *End* 100:32.

Dubliuota tiesė dubliuojama dar 4 kartus.

Iš įrankių srities pasirenkamas įrankis **Text** ir kurioje vietoje pasinaudojus įrankiu įrašoma „#.#“. Elementui priskiriama koordinatės X:20, Y:10, plotis (*Width*) 23 ir aukštis (*Height*) 18.

Analogiškai sukuriami dar 2 lementai, kuriems priskiriami elementų laukelyje priskiriami vardai

Text2 ir *Text3*. Naujiems elementams atitinkamai priskiriama koordinatės X:105, Y:0, plotis

(*Width*) 23, aukštis (*Height*) 18 ir koordinatės X:105, Y:100, plotis (*Width*) 23, aukštis (*Height*)

18. Elementui *Text1* priskiriama animacija **Value Display**. Laukelyje **States** parenkama *Analog*,

Analog – *Talpa*, **Text Format** - *#.#*. Elementams *Text2* ir *Text3* priskiriama analogiška

animacija, tik atitinkamai laukelyje **Analog** priskiriami kintamieji *Max* ir *Min*.

Klaviatūros klavišais CLTR+M iškviečiamas kintamųjų aprašymo langas, kuriame aprašomi kintamieji. Mygtuku „+“ suformuojami kintamieji pagal 2.2 lentelę.

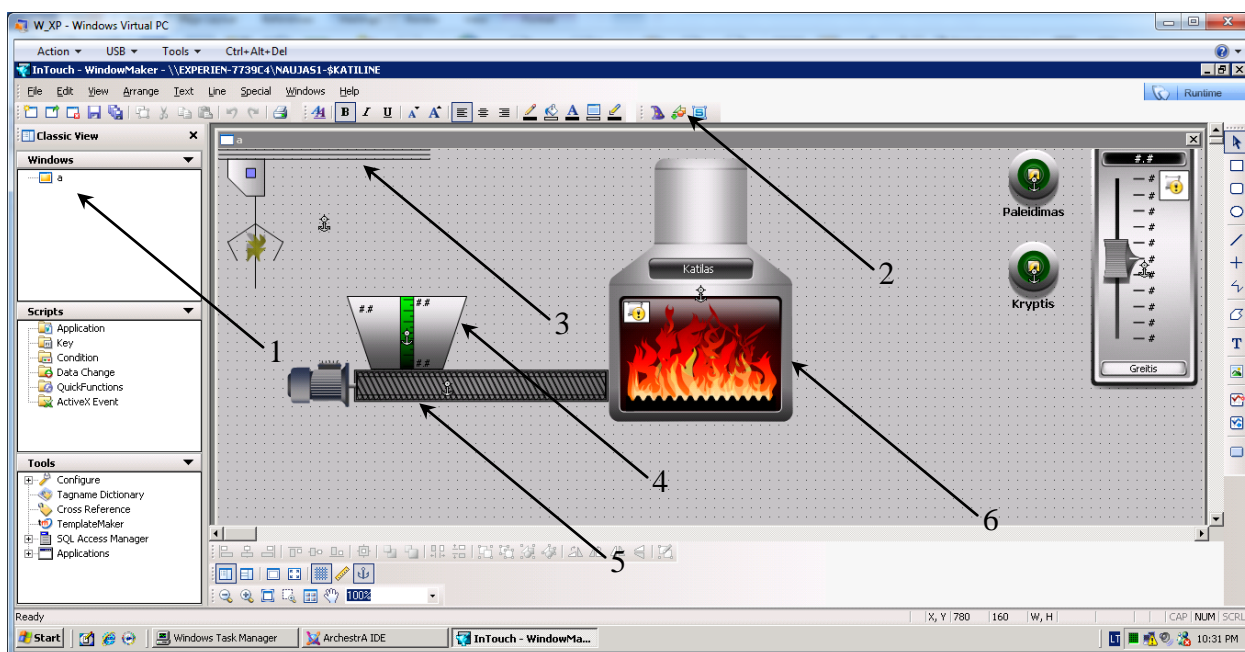
2.2 lentelė. Simbolio kintamųjų aprašymo lentelė

Kintamasis	Kintamojo tipas (Data Type)	Pradinė reikšmė (Default Value)	Matomumas (Visibility)
Indikatoriaus_spalva	String	Zalia	Public
Max	Double	0	Public
Min	Double	0	Public
Talpa	Integer	---	Public

2.5 Aplikacijos kūrimas

Skyriuje „Aplikacijų kūrimas“ aprašoma kaip sukurti žmogus-mašina sąsają InTouch aplikacijoje. Vadovaujantis aukštesniuose poskyriuose pateiktu aprašymu sukuriama kitas modelis pavadinimu **\$Katiline**. Atvėrus modelį, pasirenkama **Create new InTouch application** ir atveriamas InTouch aplinka

Pasinaudojus komanda **File\New Window...** sukuriama naujas langas pavadinimu „Katiline“. Iš Archestra IDE bibliotekos įkeliami ankstesniuose skyriuose sukurti simboliai ir vienas iš esamos bibliotekos: „Kranas“, „Sraigtas“, „Talpa“ ir „BoilerWithFlames“. Taip pat iš tos pačios bibliotekos yra paimami papildomi mygtukai ir slankmatis. 2.22 paveiksle rodyklėmis nurodyti elementai ir jų buvimo vietos. Skaičiumi „1“ pažymėta InTouch aplikacijoje sukurtų langų skaičius ir pavadinimai. Skaičiumi „2“ pažymėta Archestra IDE simbolių biblioteka. Skaičiumi „3“ pažymėtas ankstesniuose skyriuose sukurtas simbolis „Kranas“. Skaičiumi „4“ pažymėta „Talpa“. Skaičiumi „5“ pažymėta „Sraigtas“. Skaičiumi „6“ pažymėta „Katilas“.



2.22 pav. InTouch aplikacijos langas

Išdėliojus simbolius, kaip parodyta 2.22 paveiksle, reikia priskirti kintamuosius. Du kartus paspaudus kairiu pelės klavišu ant kranas, atveriamas animacijų kūrimo lentelė. Parametrai „Greitis“ sukuriama ir priskiriamas kintamasis „Greitis“, o parametrai „Kranas“ – „Paleidimas“. Bandant patvirtinti, pasirodanti lange nurodytas kintamojo vardas ir tipas. Reikiamus pausti „Save“ po to „Close“.

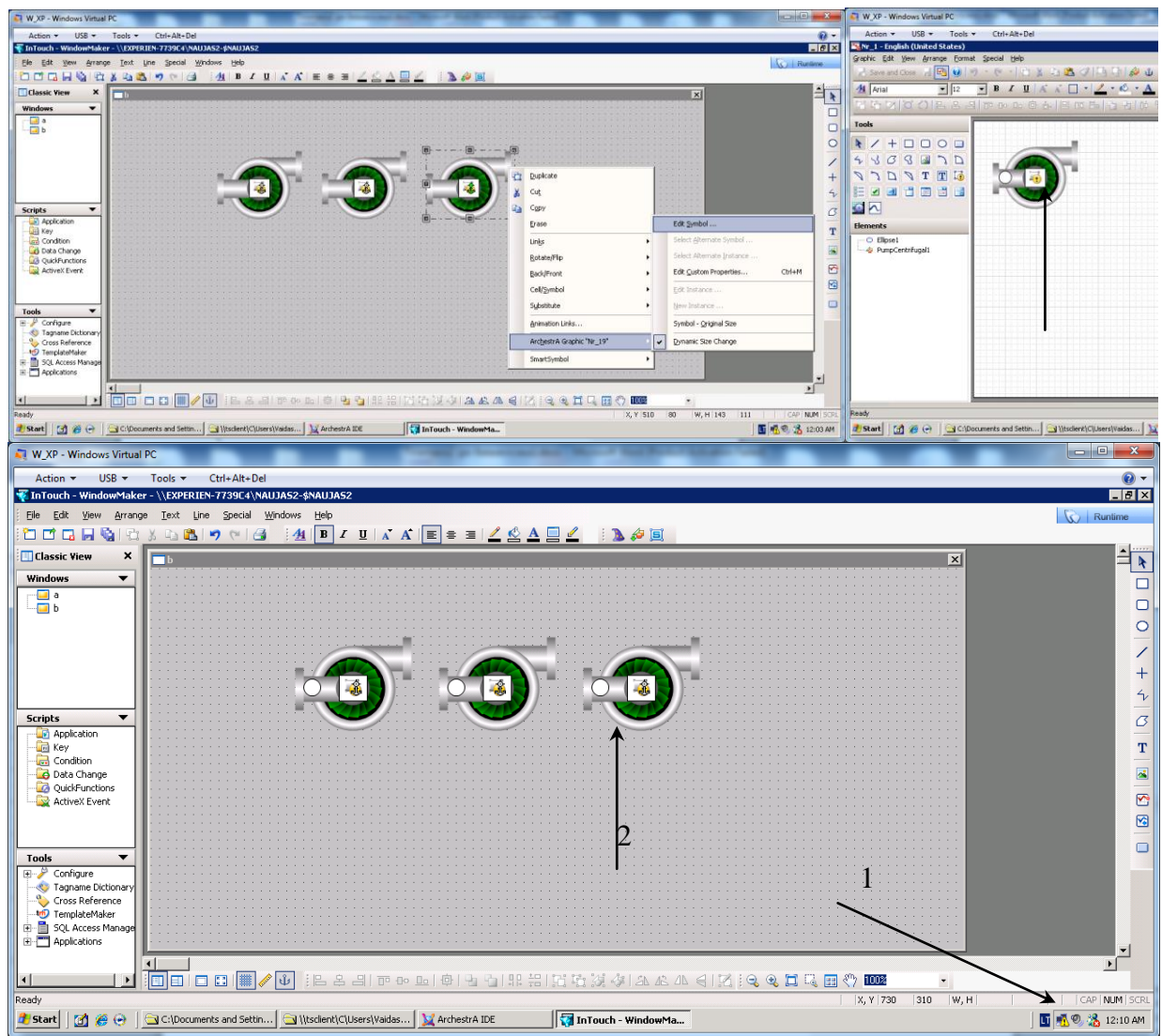
Simboliui „Talpa“ priskiriamas kintamasis *Talpa*, kurio tipas **Memory Integer**. Sraigtui taip pat priskiriami kintamieji: „Kryptis“ – **Kryptis** (tipas Memory discrete) ir „Sraigto _paleidimas“ – **Paleidimas**.

Katilo parametrui „Value“ priskiriamas kintamasis „Paleidimas“. Katilas pagal nutylėjimą pavadintas „Label“. Norint pakeisti pavadinimą reikia dešiniu klavišu paspausti ant katilo ir vadovaujantis komanda *Substitute\Substitute Strings...* pakeičiamas pavadinimas. Mygtukas taip pat priskiriamos animacijos: žaliajam – **Paleidimas**, raudonam – **Kryptis**. Pagal anksčiau minėtas procedūras reguliatoriui reikiapakeisti vardą ir parametrų reikšmes: **Value** -*Greitis*, „Min“ – 0, „Max“ – 10.

Atlikus anksčiau minėtus veiksmus gauname aplikaciją, kuri paruošta naudoti imituojant katilinės darbą. Aplikacijai sukurti buvo panaudoti ArchestrA IDE sukurti simboliai. Kadangi simboliai turėjo suformuotas vaizdines priemones ir viduje parašytus script'us, darbas InTouch aplinkoje tapo žymiai paprastesnis - užteko tiesiog susikurti kintamuosius ir juos priskirti simboliams. Šis aplikacijų metodas labai pasiteisintų dirbant vienoje srityje, kur reikalingi panašūs simboliai. Taip susidarius simbolių biblioteką galima sutaupyti projektavimo laiko.

2.5.1 Globali simbolių korekcija

Panaudotus simbolius galima keisti globaliai. Taip suteikiama galimybė sutaupyti laiko atliekant pakeitimus vizualizacijoje. Pakeitimai atliekami simbolių bibliotekoje taip pakeičiant visus tame projekte panaudotus simbolius. Pavyzdžiui, 2.23 **Error! Reference source not found.** paveiksle pateikti trys vienodi siurbliai patalpinti InTouch aplinkoje. Norint atlikti pakeitimą reikia dešiniu klavišu spustelėjus ant simbolio pasirinkti grafą *ArchestrA Graphic/Edit Symbol...* atsiradusiame lange koreguojamas simbolis. Esamu atveju pakeičiama siurblio išvaizda papildant jį baltu apskritimu. Atikus pakeitimus simbolio pakeitimai išsaugomi ir uždaromas langas. InTouch aplinkos lange nuspaudžiamas atsiradęs simbolis pažymėtas 1 rodykle ir visi trys siurbliai papildomi apskritimu (2 rodyklė)[6].



2.23 pav. Siurbliai InTouch aplinkoje

3. Intouch grafinės sąsajos panaudojimas Archestra IDE aplinkoje

Archestra IDE programinis paketas pasižymi ganėtinai patogiomis savybėmis norint panaudoti jau esamus simbolius iš Intouch ir Archestra IDE programinių paketų. Pastarųjų panaudojimą galima pastebėti skyriuje „Sraigtas“.

Viena jų galima paminėti kaip esamų Intouch aplikacijų pavertimas Archestra IDE simboliais. Kaip pavyzdys žemiau pateikta esamos Intouch aplikacijos dalies pavertimas Archestra IDE simboliu. Sukūrus Archestra IDE simbolį iš Intouch aplikacijos bus dvi identiškos aplikacijos, kurių dėka bus galima atlikti eksperimentus įvertinant aparatūros darbą naudojant skirtingas programines įrangas.

Kad negadinti originalios Intouch aplikacijos reikia padaryti jos kopiją. Vadovaujantis nuoroda **C:\Documents and Settings\All Users\Application Data\InTouchDemos** padaroma aplanko „demoapp1_1280“ kopija. Jos pavadinimas turėtų būti „Copy of

demoapp1_1280“.Paleidus programinį paketą InTouch gali automatiškai nerasti aplikacijos kopijos, todėl ją reikia surasti panaudojantįrankį *Tools\Find Application...* ir anksčiau minėtą direktoriją. Atidaroma originalios aplikacijos kopija, kurios pavadinimas turėtų būti „**Demo Application 1280 x 1024(2)**“.

Atsivėrusiame aplikacijos lange pasirenkama langas **Conveyor**. Toliau iš laukelio **Windows** ištrinami visi langai, išskyrus **Conveyor**. Kadangi buvo panaikinti kai kurie aplikacijos langai, reikia pakoreguoti script'ą. Panaudojus *Special\Scripts\Application Scripts...* ištrinamas visas script'as. Panaudojus *Special\Scripts\Window Scripts...* laukelyje **Condition Type** pasirinktis pakeičiamaj **While Showing**. Script'o laukelyje įrašomas žemiau pateiktas script'as.

```
IFCycle == 0 THEN  
VerticalMove = 0;  
HorizontalMove = 0;  
IFAutoTHEN  
Cycle = 1;  
ENDIF;  
ENDIF;
```

```
IFCycle == 1 THEN  
VerticalMove = VerticalMove + Speed;  
IFVerticalMove >= 41 THEN  
VerticalMove = 41;  
IFAutoTHEN  
Cycle = 2;  
ENDIF;  
ENDIF;  
ENDIF;
```

```
IFCycle == 2 THEN  
HorizontalMove = HorizontalMove + Speed * 3;  
IFHorizontalMove >= 134 THEN  
Cycle = 3;  
ENDIF;  
ENDIF;
```

```
IFCycle == 3 THEN  
HorizontalMove = HorizontalMove + Speed * 3;
```

```
IFAutoTHEN  
IFHorizontalMove >= 275 THEN  
Ejector = 1;  
IFHorizontalMove >= 289 THEN  
HorizontalMove = 289;  
Cycle = 4;  
ENDIF;  
ENDIF;
```

```
ELSE{AUTO}
```

```
IFEjectorTHEN
```

```
IF (HorizontalMove > 250) AND (HorizontalMove < 289) THEN  
Cycle = 7;  
ENDIF;
```

```

IF (HorizontalMove> 272) AND (HorizontalMove<310) THEN
HorizontalMove = 289;
Cycle = 4;
ENDIF;

ELSE
IFHorizontalMove> 310 THEN
Cycle = 6;
ENDIF;
ENDIF;
ENDIF; {Auto}

ENDIF; {Cycle == 3}

IFCycle == 4 THEN
VerticalMove =VerticalMove - Speed;
HorizontalMove = HorizontalMove - Speed;
IFVerticalMove<= 30 THEN
Ejector = 0;
ENDIF;
IFVerticalMove<= 0 THEN
IFAutoTHEN
Cycle = 1;
ELSE
Cycle = 0;
ENDIF;
HorizontalMove = 0;
VerticalMove = 0;
ENDIF;
ENDIF;

IFCycle == 5 THEN
VerticalMove = VerticalMove + 1;
Cycle = 51;

ENDIF;

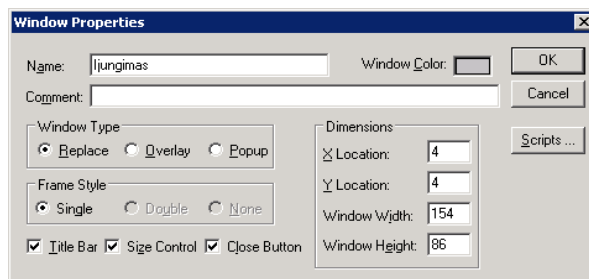
IFCycle == 7 THEN
Ejector = 1;
IFHorizontalMove< 272 THEN
HorizontalMove = HorizontalMove + Speed;
ELSE
HorizontalMove = 272;
Cycle = 5;
ENDIF;
ENDIF;

```

Panaudojus įrankį „**Button**“ nubrėžiamas mygtukas.

Panaudojus komandą *Special\Substitute Strings...* mygtuko pavadinimas pakeičiamas į „Ijungimo langas“.

Du kartus paspauduskairiu pelės klavišu ant mygtuko iškviečiamas animacijos pasirinkčių



3.1pav. Naujai kuriamo lango nustatymai

langas. Pasirenkama funkcija „**Action**“ ir atsivėrusiame lange įrašomas trumpas script'as „**Hide "Conveyor"; Show "Ijungimas";**“. Sukuriamas kitas mygtukas, kuris pavadinimas „Avarija“ ir jam priskiriama ta pati animacija su script'u „**HorizontalMove = 0; VerticalMove = 0; Cycle = 0; HideSelf;Show "Conveyor";**“. Laukelyje **Windows** paspaudus dešiniu klavišu sukuriamas naujas langas pavadinimu „Ijungimas“. Lango nustatymai parenkami taip, kaip parodyta **Error! Reference source not found.** paveiksle. Naujame lange nubrėžiamas naujas mygtukas, kuris pavadinimas „Reaktorius“ ir jam priskiriama ta pati animacija su script'u „**Hide "Ijungimas"; Show "Conveyor";**“. Taip buvo sukurti mygtukai, kurių vieno pagalba grįžtama į langą „Ijungimas“, kito atstatoma išsiliejusi statinė, trečiojo – grįžtama į langą „Ijungimas“. Atlikus minėtus veiksmus gaunamas vaizdas nurodytas **Error! Reference source not found.** paveiksle.

3.2pav. InTouch aplikacija po pertvakymo

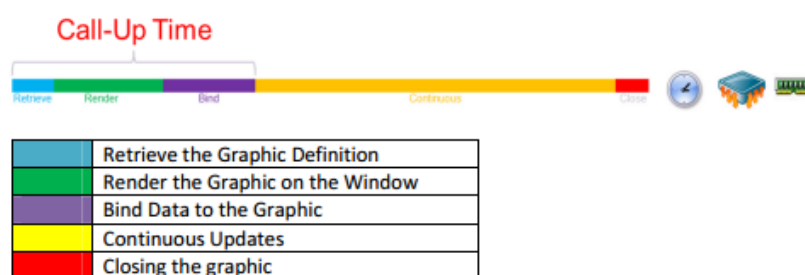
Grįžus į langą „**Conveyor**“ panaudojamas klavišas F2 taip pažymimi visi lange esantys elementai. Spustelėjus klaviatūros klavišą F3 sukuriamas vientisas elementas (**Cell**). Panaudojus komandą *Special\SmartSymbol\Generate Smart Symbols* sukuriamas „protingas“ simbolis, t.y. simbolis su viduje pasislėpusiomis animacijomis. Sukurtas simbolis pavadinamas „**Reaktorius**“. Įsijungus ArchestrA IDE simbolių biblioteką sukuriamas naujas simbolis pavadinimu „Reaktorius“. Atsivėrusiame lange iš įrankių juostos pasirenkamas komanda *Special\Import InTouch SmartSymbol...* Laukelyje **Application path:** nurodoma prieš tai sukurtos aplikacijos buvimo direktorija. Nurodžius direktoriją spaudžiama **Next >** ir

pasirenkamas aplankas pavadinimu „**InTouch Symbols**“. Atsivėrusiame lange surandamas prieš tai sukurtas „protingas“ simbolis“ (angl. *SmartSymbol*) pavadinimu „Reaktorius“ ir įkeliamas į simbolių kūrimo langą. Spustelėjus klavišą F10 atveriamas script‘o rašymo langas, kuriame įkeliamas toks pat script‘as kaip InTouch kuriamoje aplikacijoje. Taip buvo sukurtas ArchestrA IDE aplinkos simbolis panaudojant esamą InTouch aplikaciją.

4. InTouch ir ArchestrA IDE grafinei sąsajai reikalingų resursų palyginimas

ArchestrA IDE grafika suteikia šiuolaikinius grafikos ir animacijos kūrimo įrankius. Grafika paremta vektoriais, todėl leidžia nepriklausomai keisti simbolių dydį ir raišką. Nepriklausomos savybės ir script‘ai suteikia galimybę stiprinti funkcionalumą grafikos moduluose. Kadangi grafika lanksti ir galinga, ArchestrA IDE grafika gali naudoti daugiau procesoriaus laiko ir užimti daugiau atminties nei tradicinė InTouch grafika.

Neracionaliai naudojantis ArchestrA gali būti bereikalingai perkraunamas procesorius ir operatyvioji atmintis. 4.1 paveiksle matyti kaip panaudojamas procesoriaus darbas ir laikas.



4.1 pav. Procesoriaus darbo atvaizdavimas [7]

Procesas suskirstytas į 5 pozicijas, kurios aprašytos 4.1 paveikslo lentelėje.

Tam, kad nustatyti ArchestrA IDE grafiką nenaudojant tradicinės InTouch grafikos yra galimybė panaudoti spalvų gradientą eilutėms tekstui, ir figūrų užpildymui. Nors gradientai gali tikrai pagerinti spalvų kokybę ir išplėsti paletę, procesas išilgėja, tuo apkraunamas procesorius, todėl sulėtėja proceso vykdymas vykdymo metu. Naudojant gradientus svarbu, jog vartotojas tikrai pajustų poveikį [7].

Kompiuterio procesoriaus apkrovą galima matuoti panaudojus vieną iš vidinių ArchestrA IDE atributų. Funkcija iškviečiama ArchestrA IDE pagrindiniame lange dešiniu klavišu pažymėjus aplikacijos **ViewEngine_001** ir panaudojant komandą **View in Object Viewer**. Atributų sąrašė surandamas atributas **Engine.ProcessCPUload**. Dešiniu klavišu pažymėjus pasirenkama komanda **Add to Watch**. Lange **Watch List 1** atsiranda atributas ir laike kintanti jo

vertė. Kitas galimas būdas stebėti procesoriaus atminties apkrovimą yra iškviečiant funkciją **System.Diagnostics.PerformanceCounter** per žemiau pateiktą script'ą [8].

```
dim cpuCounter as System.Diagnostics.PerformanceCounter;
cpuCounter = new System.Diagnostics.PerformanceCounter();
cpuCounter.CategoryName = "Processor";
cpuCounter.CounterName = "% Processor Time";
cpuCounter.InstanceName = "_Total";
cpuCounter.NextValue();
System.Threading.Thread.Sleep(1000);
LogMessage("CPU usage: " + cpuCounter.NextValue()+"%");
```

Duomenys stebimi **System Management Console** lange. Atvėrus langą reikia išskleisti aplanką **Log viewer/(Galaxy pavadinimas)**. Pateiktas script'as patalpintas simbolio script'e. Procesoriaus apkrovimo stebėjimo langas pateiktas 4.2 paveiksle.

No	Date	Time	Process ID	Thread ID	Log Flag	Component	Message
109617	5/31/2015	8:59:10 PM	4572	5688	Info	view	Backa21: CPU usage: 4.040404%%
109618	5/31/2015	8:59:11 PM	4572	5688	Info	view	Backa21: CPU usage: 1%%
109619	5/31/2015	8:59:12 PM	4572	5688	Info	view	Backa21: CPU usage: 3%%
109620	5/31/2015	8:59:13 PM	4572	5688	Info	view	Backa21: CPU usage: 0%%
109621	5/31/2015	8:59:14 PM	4572	5688	Info	view	Backa21: CPU usage: 3%%
109622	5/31/2015	8:59:14 PM	4572	5688	Info	WWSLS	SUSPENDED SuiteLink Updates
109623	5/31/2015	8:59:15 PM	4572	5688	Info	view	Backa21: CPU usage: 0%%
109624	5/31/2015	8:59:16 PM	4572	5688	Info	view	Backa21: CPU usage: 1%%
109625	5/31/2015	8:59:17 PM	4572	5688	Info	view	Backa21: CPU usage: 2%%
109626	5/31/2015	8:59:18 PM	4572	5688	Info	view	Backa21: CPU usage: 1%%
109627	5/31/2015	8:59:19 PM	4572	5688	Info	view	Backa21: CPU usage: 2%%
109628	5/31/2015	8:59:21 PM	4572	5688	Info	view	Backa21: CPU usage: 5%%
109629	5/31/2015	8:59:22 PM	4572	5688	Info	view	Backa21: CPU usage: 5%%
109630	5/31/2015	8:59:23 PM	4572	5688	Info	view	Backa21: CPU usage: 2%%
109631	5/31/2015	8:59:24 PM	4572	5688	Info	view	Backa21: CPU usage: 5%%
109632	5/31/2015	8:59:25 PM	4572	5688	Info	view	Backa21: CPU usage: 1%%
109633	5/31/2015	8:59:26 PM	4572	5688	Info	view	Backa21: CPU usage: 3%%
109634	5/31/2015	8:59:27 PM	4572	5688	Info	view	Backa21: CPU usage: 3%%
109635	5/31/2015	8:59:28 PM	4572	5688	Info	view	Backa21: CPU usage: 1%%
109636	5/31/2015	8:59:29 PM	4572	5688	Info	view	Backa21: CPU usage: 2%%
109637	5/31/2015	8:59:30 PM	4572	5688	Info	view	Backa21: CPU usage: 1%%
109638	5/31/2015	8:59:31 PM	4572	5688	Info	view	Backa21: CPU usage: 3%%
109639	5/31/2015	8:59:32 PM	4572	5688	Info	view	Backa21: CPU usage: 1%%
109640	5/31/2015	8:59:33 PM	4572	5688	Info	view	Backa21: CPU usage: 3%%
109641	5/31/2015	8:59:34 PM	4572	5688	Info	view	Backa21: CPU usage: 3%%
109642	5/31/2015	8:59:35 PM	4572	5688	Info	view	Backa21: CPU usage: 1%%
109643	5/31/2015	8:59:36 PM	4572	5688	Info	view	Backa21: CPU usage: 4%%
109644	5/31/2015	8:59:37 PM	4572	5688	Info	view	Backa21: CPU usage: 0%%

4.2 pav. System Management Console langas

Atlikus stebėjimus, pastebėta, jog aplikacijoms vykdyti sunaudojama nuo 1 % iki 2 % procesoriaus darbo.

Pasinaudojant 3 skyriuje sukurtomis aplikacijomis, atliktas tyrimas, kuriuo siekiama išsiaiškinti atminties naudojimą, naudojant InTouch aplinkoje sukurtą aplikaciją ir tą pačią aplikaciją perdarius *Archestra IDE* simboliu, script'ą paslepiant simboliuje. Sistemos apkrovai matuoti naudojamas įrankis **Task Manager**, imant informaciją iš kortelės **Processes**.

Siekiant užtikrinti tyrimo rezultatų objektyvumą, panaudota keletas kompiuterių su skirtingais parametrais ir skirtingomis operacinėmis sistemomis. Kompiuterių parametrai pateikti 4.1 lentelėje.

4.1 lentelė. Kompiuterių palyginimas

	Kompiuteris Nr. 1	Kompiuteris Nr. 2
Procesorius	Intel (R) Core (TM) i5-3210M @ 2,50GHz	AMD Phenom II X2 545 780MHz
Įdiegta atmintis (RAM)	6,00 GB	3,25 GB
Operacinės sistemos tipas	64bit	32bit
Operacine sistema	Win7	Win XP

Pradedant tyrimą paleidžiamas **Task Manager**. Paleidus InTouch programinį paketą jau yra sukurtos aplikacijos „**Demo Application 1280x1024 (2)**“ ir „**Reaktorius**“. Naudojant komandą **WindowViewer** paleidžiama aplikacija pavadinimu „**Demo Application 1280x1024 (2)**“ ir **Task Manager** lange stebimas „**view.exe**“ proceso atminties panaudojimas (**Mem Usage**). Aplikacijose yra sukurti du langai – „Ijungimas“ ir „Reaktorius“. Langas „Ijungimas“ naudojamas paleisti patį procesą, esantį lange „Reaktorius“. Vykstant tyrimui fiksuojami duomenys, kai: paleidžiamas **WindowViewer**, atveriamas langas „Ijungimas“, atveriamas langas „Reaktorius“, grįžtama atgal į langą „Ijungimas“ ir t.t. Atlikus tyrimą su viena aplikacija, tyrimas pakartojamas su kita aplikacija.

Pirmiausia tyrimas atliekamas inžinerinėje stotyje, tada toks pat tyrimas pakartojamas valdymo stotyse (žr. 2.1 pav.). Inžinerinėje stotyje aplikacija paleidžiama Archesrtra IDE aplinkoje atvėrus **WindowMaker**, o iš jo **WindowViewer**. Valdymo stotyje **WindowViewer** paleidžiamas per **Intouch Manger**.

Trijų bandymų duomenų vidurkiai pateikti 4.2, 4.3, 4.4 ir 4.5 lentelėse.

4.2 lentelė. Kompiuteris Nr. 1. Tyrimas, naudojant inžinerinę stotį

	Demo Application 1280x1024 (2)	\$Reaktorius	Naudojamos atminties santykis
Paleidus programinį paketą	31832 KB	62752 KB	1,97
Atvėrus langą „Ijungimas“	32036 KB	63224 KB	1,97
Atvėrus langą „Reaktorius“	35620 KB	73296 KB	2,06
Atvėrus langą „Ijungimas“	35620 KB	73128 KB	2,05
Atvėrus langą „Reaktorius“	35620 KB	73468 KB	2,06
Atvėrus langą „Ijungimas“	35620 KB	73144 KB	2,05
Atvėrus langą „Reaktorius“	35620 KB	73496 KB	2,06

4.3 lentelė. Kompiuteris Nr. 2. Tyrimas, naudojant inžinerinę stotį

	Demo Application 1280x1024 (2)	\$Reaktorius	Naudojamos atminties santykis
Paleidus programinį paketą	32204 KB	63263 KB	1,96
Atvėrus langą „Ijungimas“	32482 KB	64284 KB	1,98
Atvėrus langą „Reaktorius“	36012 KB	75561 KB	2,10
Atvėrus langą „Ijungimas“	36012 KB	74962 KB	2,08
Atvėrus langą „Reaktorius“	36012 KB	75731 KB	2,10
Atvėrus langą „Ijungimas“	36012 KB	74971 KB	2,08
Atvėrus langą „Reaktorius“	36012 KB	75712 KB	2,10

Atlikus pirmąją dalį tyrimo, iš 4.2 ir 4.3 lentelių matyti, jog, išvedus stulpelio „Naudojamos atminties santykis“ duomenų aritmetinį vidurkį, Kompiuteryje Nr. 1 ArchestrA IDE aplikacijos sunaudojamas atminties kiekis yra 2,03 karto didesnis nei InTouch aplikacijos. Kompiuterio Nr. 2 duomenys parodė, jog ArchestrA IDE aplikacijos sunaudojamas atminties kiekis yra 2,06 karto didesnis nei InTouch aplikacijos.

4.4 lentelė. Kompiuteris Nr. 1. Tyrimas, naudojant valdymo stotį

	Demo Application 1280x1024 (2)	\$Reaktorius	Naudojamos atminties santykis
Paleidus programinį paketą	32 348KB	52 682KB	1,63
Atvėrus langą „Ijungimas“	32 632KB	52 708KB	1,62
Atvėrus langą „Reaktorius“	36 256KB	59 504KB	1,64
Atvėrus langą „Ijungimas“	36 256KB	59 292KB	1,64
Atvėrus langą „Reaktorius“	36 256KB	58 500KB	1,61
Atvėrus langą „Ijungimas“	36 256KB	58 144KB	1,60
Atvėrus langą „Reaktorius“	36 256KB	58 524KB	1,61

4.5 lentelė. Kompiuteris Nr. 2. Tyrimas, naudojant valdymo stotį

	Demo Application 1280x1024 (2)	\$Reaktorius	Naudojamos atminties santykis
Paleidus programinį paketą	32476 KB	53672 KB	1,65
Atvėrus langą „Ijungimas“	32713 KB	54109 KB	1,65
Atvėrus langą „Reaktorius“	36507 KB	61983 KB	1,70
Atvėrus langą „Ijungimas“	36507 KB	61430 KB	1,68
Atvėrus langą „Reaktorius“	36507 KB	61791 KB	1,69
Atvėrus langą „Ijungimas“	36507 KB	61109 KB	1,67
Atvėrus langą „Reaktorius“	36507 KB	61816 KB	1,69

Atlikus bandymus pastebėta, jog užkrovimo metu rezervuojama skirtingas atminties kiekis. Iš 4.4 ir 4.5 lentelėse pateiktų duomenų matyti, jognaudojant ArchestrA IDE sunaudojama atitinkamai 1,62 ir 1,68 karto daugiau atminties nei naudojant aplikaciją, sukurtą InTouch programiniu paketu.

Atvėrus langą „Ijungimas“ sunaudojamas atminties kiekis padidėja nedaug dėl to, jog langas parinktas toks, kad būtų sunaudoti minimalūs resursai, t.y. lange patalpintas vienas mygtukas. Pirmą kartą atvėrus langą „Reaktorius“, sunaudojamos atminties kiekis padidėja

žymiai daugiau. Toliau tarpusavyje keičiant langus, sunaudojamos atminties kiekis, naudojant InTouch programinį paketą nesikeičia. Taip yra dėl to, jog reikalingi duomenys užkraunami į „buferį“, iš kurio programinis paketas naudoja duomenis, reikalingus aplikacijosvykdymui.

Naudojant aplikaciją su programinio paketo ArcestrA IDE simboliu, atminties sunaudojama daugiau ir netolygiai dėl to, kad kartu aktyvuojamas programinio paketo grafikos variklis, kurio dėka galima naudoti ArcestrA IDE simbolius.

Palyginus 4.2, 4.3, 4.4 ir 4.5 lenteles, galima teigti, jog daugiau atminties rezervuojama aplikaciją paleidžiant inžinerinėje stotyje (4.2 ir 4.3 lentelės) nei valdymo stotyse (4.4 ir 4.5 lentelės).

Išvados

Atlikus ArchestrA IDE grafinės sąsajos kūrimo galimybių analizę galima teigti, kad

1. ArchestrA IDE aplinka leidžia kurti supervizorinės aplikacijos struktūrą pagal požiūrį „iš apačios į viršų“. ArchestrA IDE simbolis (šablonas) nėra tiesiog grafikos elementas. ArchestrA sukurtame šablone apibrėžiama įėjimai/išėjimai, smulkesni grafiniai elementai, šablono dokumentacija, aliarmai ir įvykiai, apsauga, *Historian* informacija, šablono veikimo logika ir script'iai. Toliau apjungiant šablonus į sudėtingesnius arba panaudojant sukurtus elementarius Wonderware WindowMaker sukuriama proceso žmogus-mašina sąsaja. Aplikacija vykdoma InTouch programa WindowViewer.

2. ArchestrA IDE priešingai nei InTouch turi kintamųjų tipus laiko ir datos nurodymui ir palaiko specialiuosius simbolius, kurie naudojami kuriant daugiakalbes aplikacijas ; ArchestrA IDE nepalaiko InTouch esamos galimybės slėpti ir rodyti langus, tačiau, kuriant simbolių šablonus, ši funkciją nėra būtina; ArchestrA IDE nepalaiko script'o funkcijos **Application Script**, tačiau leidžia kurta script'us kiekvienam simboliui atskirai.

3. Sukurta aplikacija panaudojant ArchestrA IDE sukurtus simbolius parodė, kad simbolių bibliotekos sukūrimui reikia laiko, tačiau vėliau tai supaprastina aplikacijų kūrimą, nes kiekvienas simbolis su savo logika gali būti naudojamas tiesiogiai arba lengvai adaptuojamas kitose aplikacijose.

4. Atlikus sukurtų InTouch ir ArchestrA IDE aplikacijų grafinės sąsajos naudojamos atminties apkrovimo tyrimą pastebėta, jog ArchestrA kurta aplikacija gali sunaudoti nuo 1,60 iki 2,10 kartų daugiau atminties nei InTouch kurta aplikacija. Taip pat pastebėta, jog užkrovus aplikaciją „\$Reaktorius“ inžinerinėse stotyse suvartojama atitinkamai 1,26 ir 1,22 kartų daugiau atminties nei valdymo stotyse.

Literatūros sąrašas

1. *Iveta Zolotová, Roman Mihal', Ratislav HOŠÁK. Object for Visualization of Process Data in Supervisory Control.* Aspects of Computational Intelligence: Theory and Applications e-ISSN 2193 51-61psl. ©Springer-Verlag Berlin Heidelberg 2013
2. http://global.wonderware.com/EN/PDF%20Library/Wonderware_System_Platform3_1_data_sheet_spread.pdf
3. *Creating and Managing ArchestrA Graphics User's Guide*, 2012
4. <http://ctp.wonderware.com/GetStartIT10/InTouchPages/eM1S1b-Lab01.htm> (žiūrėta 2015-01-18)
5. <http://www.everdyn.com/why-png-files-rule-part-2-how-to-build-really-cool-wonderware-graphics-fast/> (žiūrėta 2014-04-18)
6. <https://www.youtube.com/watch?v=z95j5Unfrdc> (žiūrėta 2015-01-20)
7. *ArchestrA Graphics Development Guidelines pdf*
8. https://insource.mindtouch.us/Wonderware_Application_Server/Tech_Notes/TN_AppSvr10_4_Capturing_CPU_load_in_Archestra_script(žiūrėta 2015-04-20)