

**მულტიმოდალური გადაზიდვების მართვის ავტომატიზებული
სისტემის აგება დაპროექტების CASE- და დაპროგრამების
ჰიბრიდული ტექნოლოგიებით**

გიორგი გოგინაიშვილი, გიორგი სურგულაძე, ნინო თოფურია,
ლილი პეტრიაშვილი, გაია სურგულაძე
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

განიხილება ტვირთების მულტიმოდალური გადაზიდვების მართვის ავტომატიზებული სისტემის მონაცემთა ბაზის დაპროექტების, მისი პროგრამული რეალიზაციის და მომხმარებელთა ინტერფეისების აგების საკითხები დაპროექტების CASE- და დაპროგრამების ჰიბრიდული ტექნოლოგიებით. კერძოდ, შემოთავაზებულია მულტიმოდალური გადაზიდვების (გემი, რკინიგზა, ავტო- და საჰაერო ტრანსპორტი) საპრობლემო სფეროს კონცეპტუალური სქემები კლიენტის (ტვირთის მფლობელი), ტვირთის (გადაზიდვის ობიექტი) და მიმწოდებლის (გადამზიდავი) ცხრილებით, ობიექტ-როლური და არსთა დამოკიდებულების მოდელირების (ORM/ERM) ინსტრუმენტებით Visual Studio.NET გარემოში და Ms SQL Server პაკეტით. აგებულია მონაცემთა ბაზის განახლების ფუნქციების ინტერფეისი დაპროგრამების (WPF, C#, XAML) ინტეგრირებულ გარემოში.

საკვანძო სიტყვები: მართვის ავტომატიზებული სისტემა. მულტიმოდალური გადაზიდვა. კონცეპტუალური მოდელი. ORM. ERM. მონაცემთა რელაციური ბაზა. დაპროგრამების. ტექნოლოგია.

1. შესავალი

მულტიმოდალური გადაზიდვების პროცესი, რომლის ძირითადი მიზანი ტვირთების ტრანსპორტირებაა მიმწოდებლიდან დამკვეთამდე, არის მომსახურების განაწილებული სისტემა. მარტივად რომ წარმოვიდგინოთ, მიმწოდებელი (Supplier_ID) აგზავნის ტვირთს (Freight_ID) დამკვეთის (Client_ID) მისამართზე (Client_Address) [1].

როგორც ცნობილია, კლენტსა (ტვირთის მფლობელი) და მიმწოდებელს (გადამზიდავი) შორის ხელშეკრულებას აფორმებს ექსპედიტორი (შუამავალი), რომელსაც გააჩნია საჭირო ინფორმაცია ადგილობრივი და საერთაშორისო გადაზიდვების აგენტების, მარშრუტებისა და შესაბამისი ფასების შესახებ (ამ უკანასკნელის ცვლილებების შესახებაც) და სხვა.

ქვემოთ მოცემული გვაქვს მულტიმოდალური გადაზიდვების პროცესის ინფრასტრუქტურის ძირითადი ობიექტების და მათი თვისებების (ატრიბუტების) სემანტიკური აღწერა, რაც მომავალში გამოყენებულ იქნება ავტომატიზებული სისტემის მონაცემთა ბაზის ასაგებად [2].

ტვირთი: იდენტიფიკატორი, ტიპი, მდგომარეობა, შეფუთვის ტიპი, ერთეულის ზომები (სიგრძე, სიგანე, სიმაღლე), ერთეულის მოცულობა, ჯამური მოცულობა, ერთეულის წონა, ერთეულის რაოდენობა, ჯამური წონა, უსაფრთხოება, საბაჟო კოდი, გამგზავნი, მიმღები, საწყისი მდებარეობა, საბოლოო მდებარეობა და სხვა.

კლიენტი - იდენტიფიკატორი, დასახელება/ვინაობა, იურიდიული/ფიზიკური პირი, მისამართი, ტელეფონი, ელ-მისამართი და სხვა;

მიმწოდებელი - იდენტიფიკატორი, დასახელება, იურიდიული/ფიზიკური პირი, მისამართი, ტელეფონი, ელ-მისამართი, ფაქსი, ტრანსპორტის სახე და სხვა;

გემი: იდენტიფიკატორი, ტიპი, კრანით/უკრანო, მდგომარეობა, სასაწყობო ლიმიტი, ტვირთამწეობა, ტვირთმოცულობა, ადგილმდებარეობა და სხვა.

თვითმფრინავი: იდენტიფიკატორი, ტიპი, მდგომარეობა, ტვირთმოცულობა, გადასაზიდი ერთეულის დასაშვები ზომები (სიგრძე, სიგანე, სიმაღლე) ადგილმდებარეობა და სხვა.

ავტოტრანსპორტი: იდენტიფიკატორი, ტიპი, მდგომარეობა, ტვირთმოცულობა, გადასაზიდი ერთეულის დასაშვები ზომები (სიგრძე, სიგანე, სიმაღლე), გადასაზიდი ერთეულის დასაშვები წონა, მაქსიმალური დატვირთვა, ადგილმდებარეობა და სხვა.

სარკინიგზო სატვირთო ვაგონი: იდენტიფიკატორი, ტიპი, ტვირთამწეობა, მოცულობა, დასაშვები დატვირთვა, ადგილმდებარეობა, მიმწოდებლის იდენტიფიკატორი, მდგომარეობა და სხვა;

საწყობი: იდენტიფიკატორი, სახე, ფართობი, სართული, დაკავებულობის პროცენტი, დასაშვები დატვირთვა, ადგილმდებარეობა, მისამართი, მიკუთვნება რაიონზე და სხვა;

გადაზიდვის ხელშეკრულება კლიენტთან: იდენტიფიკატორი, საწყისი მდებარეობა, თარიღი-1, საბოლოო მდებარეობა, თარიღი-2, გადაზიდვის ღირებულება, გადახდილი თანხა, გადახდის-თარიღი, მდგომარეობა და სხვა.

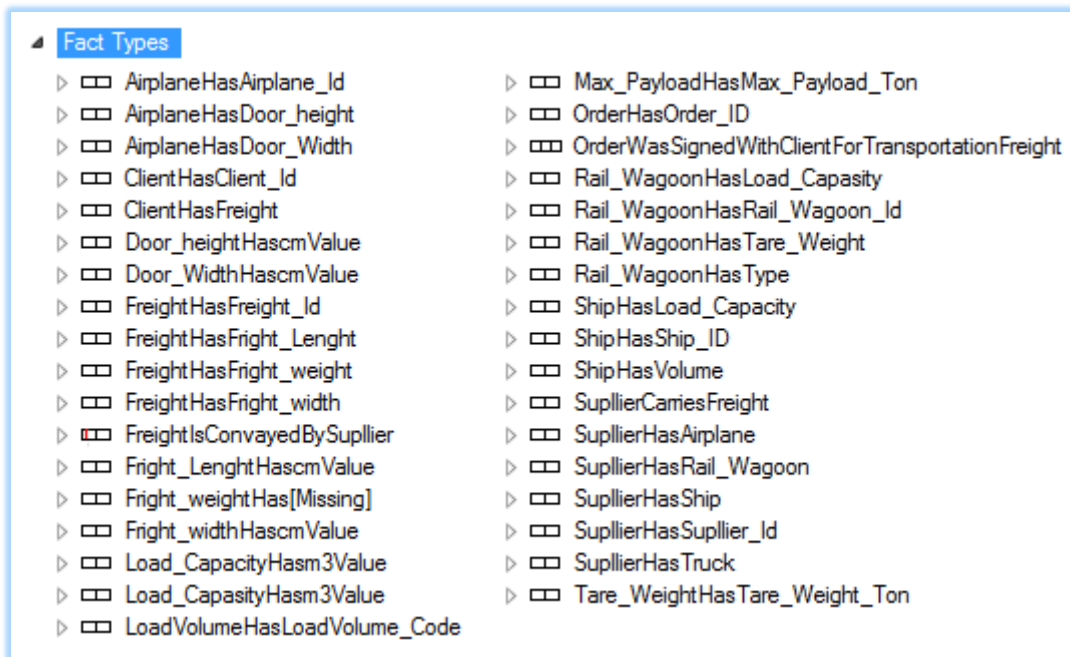
2. ძირითადი ნაწილი

2.1. მონაცემთა ბაზის დაპროექტება და რეალიზაცია

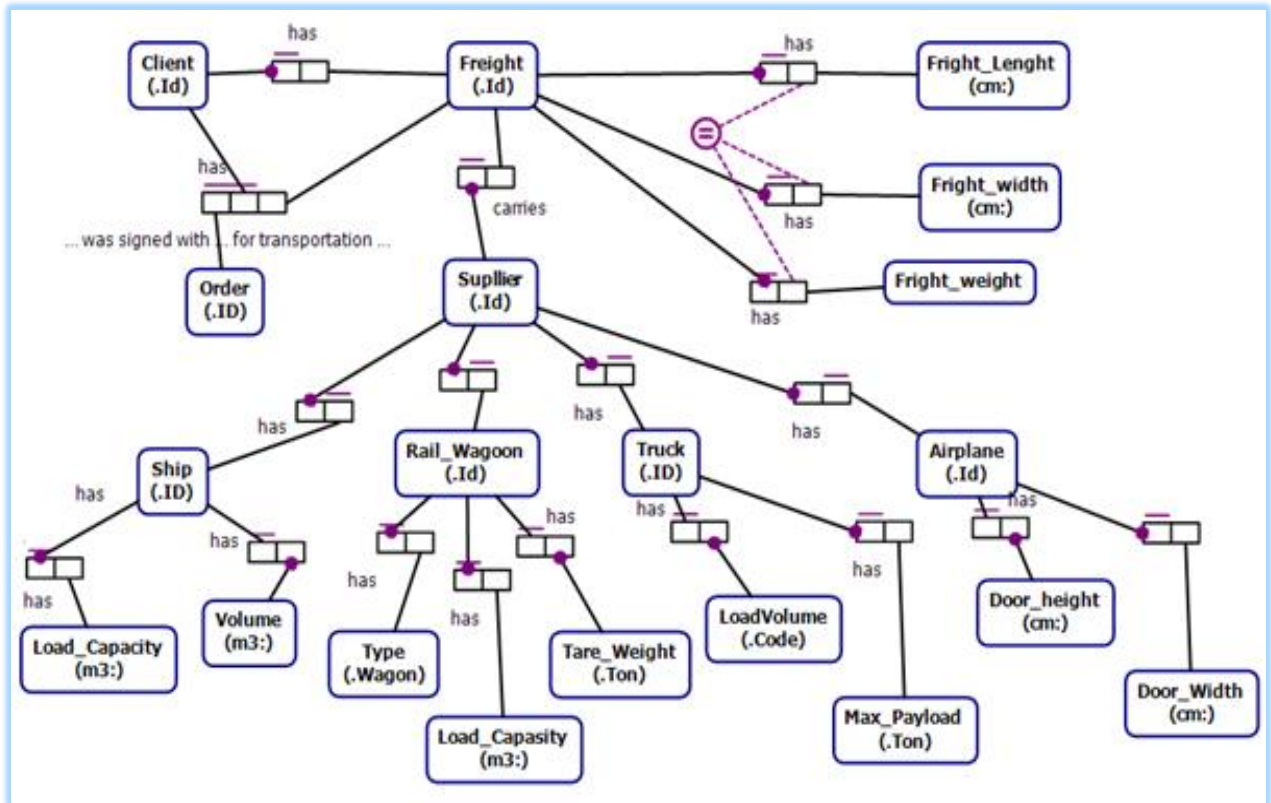
მულტიმოდალური გადაზიდვების საპრობლემო სფეროს მონაცემთა ბაზის ასაგებად საჭიროა მისი ობიექტების, ობიექტთაშორისი რელაციური კავშირების (პრედიკატების) და კონკრეტული ინფორმაციის გადატანა ბაზაში. ამისათვის ჩვენ ვიყენებთ ობიექტ-როლური მოდელირების CASE ინსტრუმენტულ საშუალებას, როგორცაა Natural ORM Architect [3]. ამ სფეროში არსებული ფაქტების აღწერა ხორციელდება კატეგორიალური მიდგომისა (სალაპარაკო ენის გრამატიკული წესები) და ალგებრულ-ლოგიკური თეორიის კანონების საფუძველზე [4,5]. მაგალითად, f1: კლიენტს აქვს ტვირთი; f2: კლიენტს აქვს იდენტიფიკატორი; f3: ტვირთს აქვს იდენტიფიკატორი; f4: ტვირთს აქვს გადასატანი მისამართი; f5: მიმწოდებელს აქვს ტრანსპორტი; და ა.შ. 1-ელ ცხრილში მოცემულია ჩვენი ობიექტების შესაბამისი ფაქტების აღწერის ფრაგმენტი NORMA გარემოში.

აქტის ტიპების (პრედიკატების) ცხრილი

ცხრ.1



მულტიმოდალური გადაზიდვების საპრობლემო სფეროს ფაქტების აღწერისა და მათი Visual Studio.NET + NORMA ინტეგრირებულ სამუშაო გარემოში გადატანის შემდეგ ვლელობთ 1-ელ ნახაზზე ნაჩვენებ კონცეპტუალურ სქემას, რომელიც ობიექტ-როლური მოდელია.



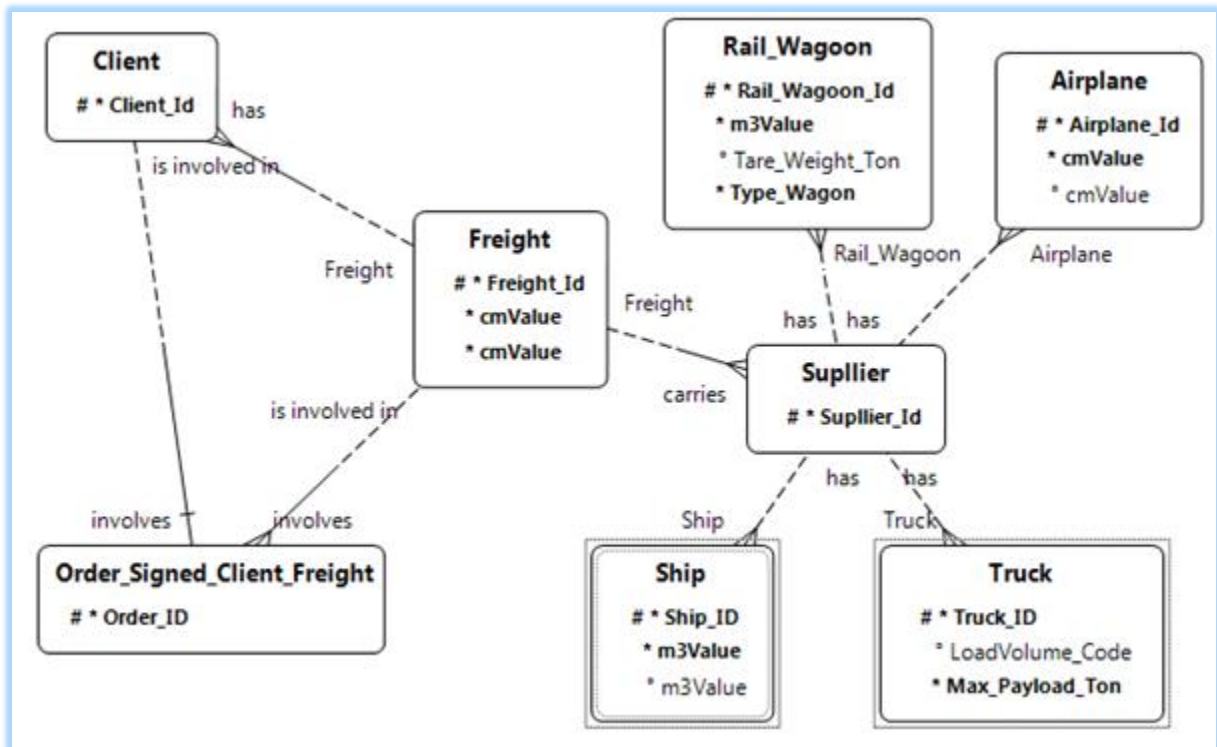
ნახ.1. საპრობლემო სფეროს კონცეპტუალური სქემა:
ობიექტ-როლური მოდელი

დიდი მართკუთხედებით აისახება ობიექტები (მაგალითად, Client, Freight, Airplane და სხვა) და მათი თვისებები (მაგალითად, Load_Capacity, Volume, Freight_Length და სხვა), პატარა მართკუთხედებით კი – პრედიკატები მათ შორის (მაგალითად, „Ship has Load_Capacity” და ა.შ.). აქ „მრგვალთაგა” კავშირის ხაზებით და პრედიკატებზე „ხაზგასმით” გვეძლევა დამატებითი ინფორმაცია ობიექტებს შორის მრავლობითი კავშირების შესახებ, როგორცაა მაგალითად, 1:1, 1:N და M:N.

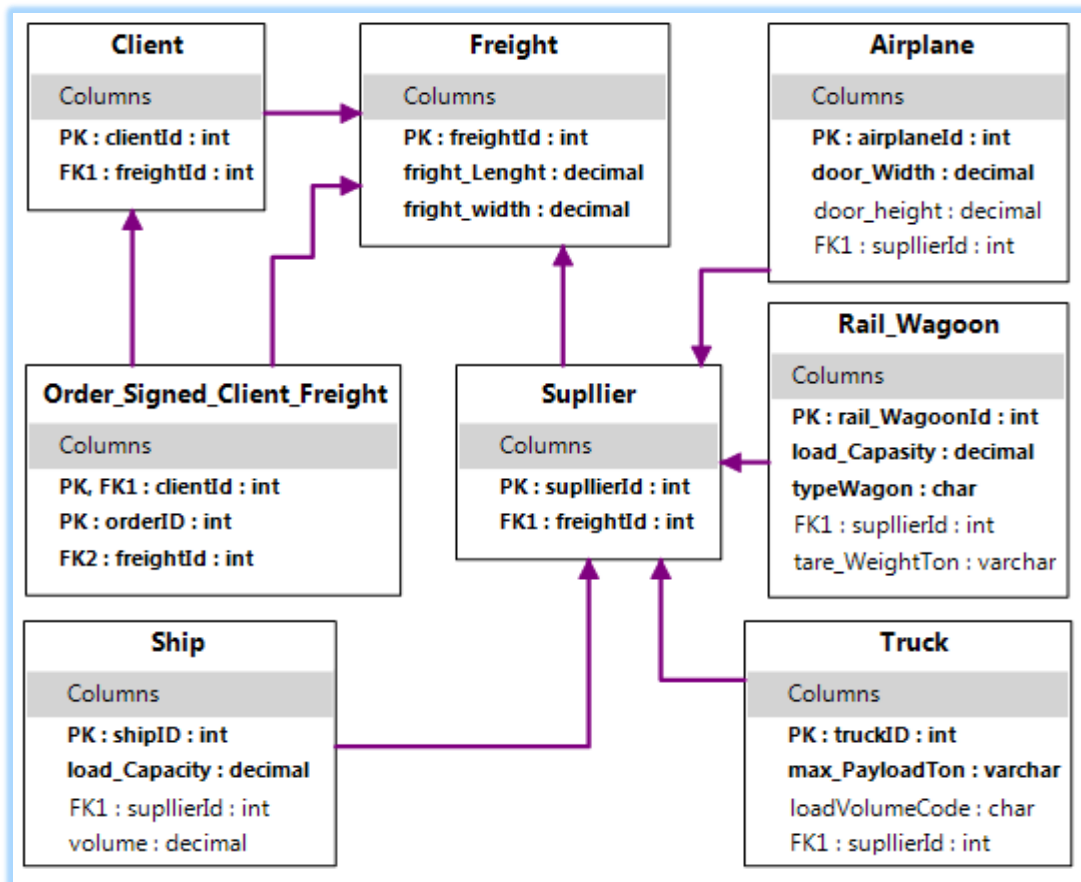
აქვე შესაძლებელია მივიღოთ, ე.წ. რიჩარდ ბარკერის დიაგრამა, რომელშიც ასეთი კავშირები ობიექტებს შორის უკეთესად ჩანს (ნახ.2). იგი ჩვეულებრივი არსთა-დამოკიდებულების მოდელია, რომელსაც იყენებენ ORACLE CASE მიმდევრები [6].

ჩვენი შემდეგი ბიჯი დაკავშირებულია ORM დიაგრამით მიღებული კონცეპტუალური სქემიდან ეკვივალენტური არსთა-დამოკიდებულების მოდელის (ERM) დაპროექტებასთან. ეს პროცესი დიალოგურ რეჟიმში ხორციელდება, სადაც მომხმარებელს შეუძლია სისტემის მიერ შემოთავაზებულ სქემაში შეიტანოს საჭირო ცვლილებები.

მე-3 ნახაზზე ნაჩვენებია მულტიმოდალური გადაზიდვების საპრობლემო სფეროს ER სქემა. იგი ახლოსაა რელაციური ბაზების დაპროექტების ობიექტ-ორიენტირებული მიდგომის ტრადიციულ დიაგრამებთან, რომლებიც აიგება MsVisio, Enterprise Architect, Rational Rose და სხვა ინსტრუმენტებით.



ნახ.2. ბარკერის დიგრამა (Barker ER view)



ნახ.3. კონცეპტუალური სქემა (ER Model)

ჩვენ აღწერეთ კონცეპტუალური სქემის ავტომატიზებული ფორმირების პროცესი ORM->ERM. აქვე შეიძლება აღვნიშნოთ, რომ მე-3 ნახაზზე მოცემულია მხოლოდ ფრაგმენტი და მისი სრული ვერსიის მიღება შესაძლებელია პირველ ეტაპზე ფაქტების თანდათანობით დამატებით.

მე-2 ცხრილში მოცემულია სისტემის მიერ ფორმირებული მულტიმოდალური გადაზიდვების საპრობლემო სფეროს კონცეპტუალური სქემის შესაბამისი ფაქტების ვერბალიზაციის ლისტინგი.

ფაქტების ვერბალიზაციის ცხრილი

ცხრ.2

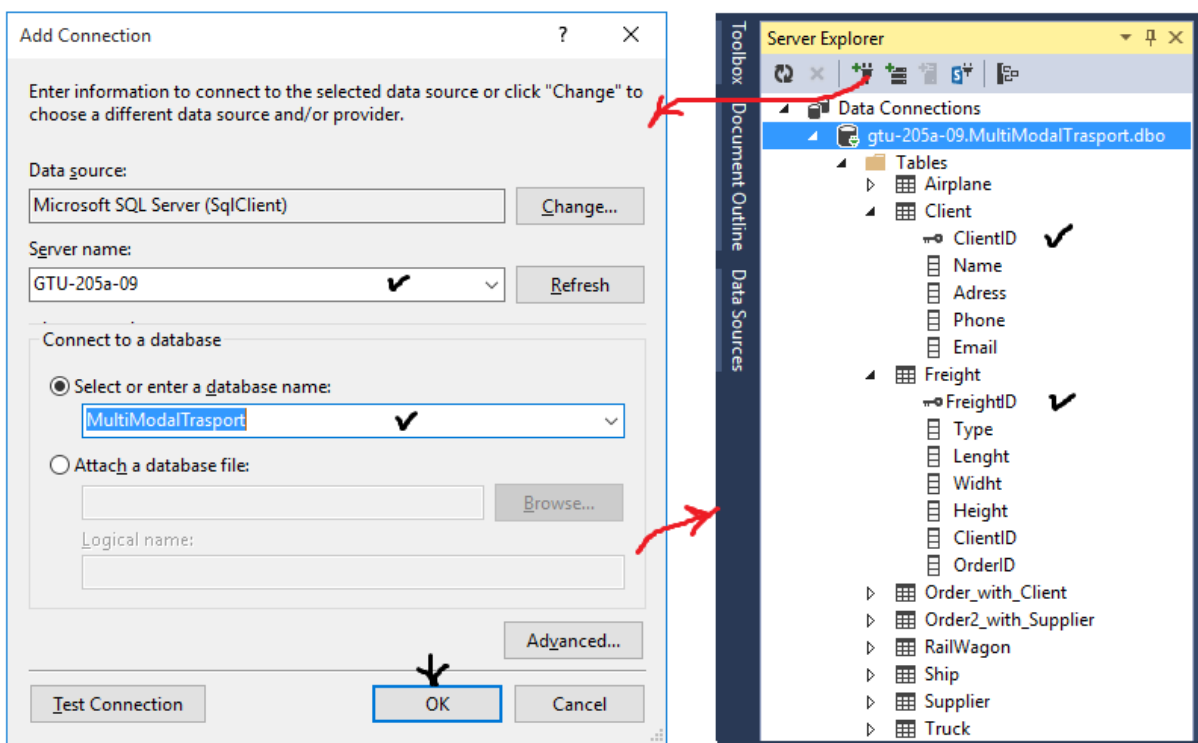
<p>Freight is an entity type. Reference Scheme: Freight has Freight_Id. Reference Mode: .Id. Fact Types: Freight has Freight_Id. Freight is conveyed by Supllier. Supllier carries Freight. Freight has Freight_Lenght. Freight has Freight_width. Freight has Freight_weight. Client has Freight. Order was signed with Client for transportation Freight.</p> <p>Client is an entity type. Reference Scheme: Client has Client_Id. Reference Mode: .Id. Fact Types: Client has Client_Id. Client has Freight. Order was signed with Client for transportation Freight. Each Client has exactly one Freight. It is possible that more than one Client has the same Freight.</p> <p>Supplier is an entity type. Reference Scheme: Supllier has Supplier_Id. Reference Mode: .Id. Fact Types: Supllier has Supplier_Id. Freight is convayed by Supplier. Supplier carries Freight. Supplier has Ship. Supplier has Rail_Wagoon. Supplier has Truck. Supplier has Airplane.</p> <p>Order is an entity type. Reference Scheme: Order has Order_ID. Reference Mode: .ID. Fact Types: Order has Order_ID. Order was signed with Client for transportation Freight. For each Order and Client,that Order was signed with that Client for transportation at most one Freight. This association with Order, Client provides the preferred identification scheme for OrderWasSignedWithClientForTransportationFreight.</p>	<p>Truck is an entity type. Reference Scheme: Truck has Truck_ID. Reference Mode: .ID. Fact Types: Truck has Truck_ID. Truck has LoadVolume. Truck has Max_Payload. Supllier has Truck. Each Supllier has some Truck. For each Truck, at most one Supllier has that Truck. It is possible that the same Supllier has more than one Truck.</p> <p>Airplane is an entity type. Reference Scheme: Airplane has Airplane_Id. Reference Mode: .Id. Fact Types: Airplane has Door_height. Airplane has Door_Width. Airplane has Airplane_Id. Supllier has Airplane. Each Supllier has some Airplane. For each Airplane, at most one Supllier has that Airplane. It is possible that the same Supllier has more than one Airplane.</p> <p>Ship is an entity type. Reference Scheme: Ship has Ship_ID. Reference Mode: .ID. Fact Types: Ship has Ship_ID. Supplier has Ship. Ship has Load_Capacity. Ship has Volume. Each Supplier has some Ship. For each Ship, at most one Supllier has that Ship. It is possible that the same Supllier has more than one Ship.</p> <p>Rail_Wagoon is an entity type. Reference Scheme: Rail_Wagoon has Rail_Wagoon_Id. Reference Mode: .Id. Fact Types: Rail_Wagoon has Rail_Wagoon_Id. Supplier has Rail_Wagoon. Rail_Wagoon has Type. Rail_Wagoon has Load_Capacity. Rail_Wagoon has Tare_Weight. Each Supllier has some Rail_Wagoon. For each Rail_Wagoon, at most one Supllier has that Rail_Wagoon. It is possible that the same Supplier has more than one Rail_Wagoon.</p>
---	--

მომდევნო ბიჯზე, მიღებული ER-მოდელის შესაბამისად, სისტემა გვაძლევს DDL-ფაილს (მონაცემთა აღწერის ენა), რომელიც ჩვენ შემთხვევაში გამოიყენება SQL Server-ბაზის ასაგებად [7].

2.2. მონაცემთა ბაზასთან მუშაობის ინტერფეისის აგება

მულტიმოდალური გადაზიდვების ბაზის დაპროექტებისა და მისი DLL-ფაილით SQL Server 2012-ში რეალიზაციის შემდეგ, საჭიროა აიგოს მომხმარებლის ინტერფეისის პროგრამა Visual Studio.NET 2013 ინტეგრირებულ გარემოში დაპროგრამების ჰიბრიდული ტექნოლოგიის გამოყენებით, როგორცაა WPF (Windows Presentation Foundation) [8,9]. ჰიბრიდული ტექნოლოგიის კონცეფცია გულისხმობს, რომ ამ მეთოდოლოგიით შესაძლებელია როგორც ვინდოუსის, ასევე ვებ-აპლიკაციების აგება (შედეგები აისახება ვინდოუსის ფორმაზე ან რომელიმე ინტერნეტ ბრაუზერში).

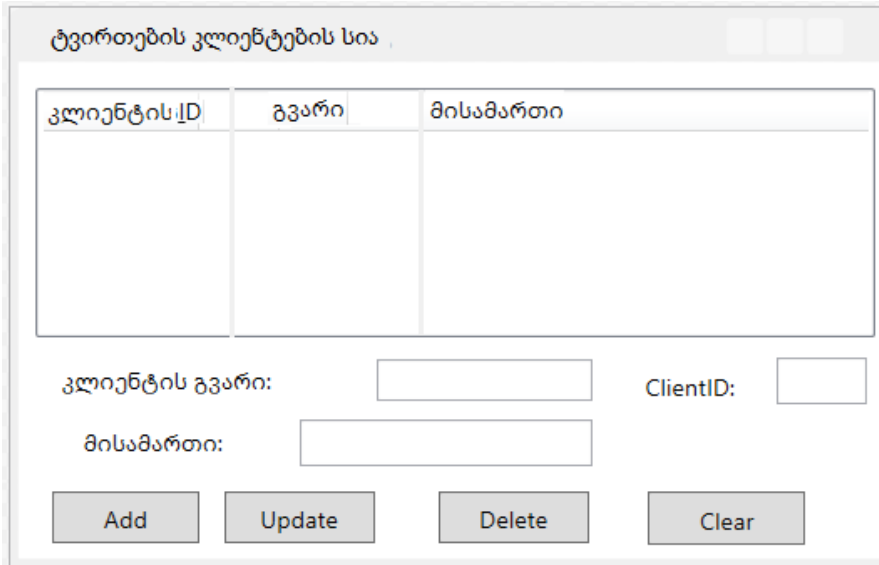
თავდაპირველად საჭიროა ახალი პროექტის შექმნა Visual Studio.NET-ში, შემდეგ კი ჩვენი მონაცემთა ბაზის მიერთება ამ პროექტთან. მე-4 ნახაზზე ნაჩვენებია ბაზის მიერთების (Data Connections) პროცედურა. განახლება სერვერის სახელი (მაგალითად, GTU-205a-09) და აირჩევა მასში მონაცემთა ბაზის სახელი (MultiModalTransport), ბოლოს OK და შედეგად მივიღებთ ნახაზის Server Explorer-ში და Airplane, Client, Freight სხვა ცხრილებს.



ნახ.4. მონაცემთა ბაზის დაკავშირება C#.NET პროგრამასთან

მონაცემთა ბაზის ცხრილებთან მუშაობის ინტერფეისის სქემა წინასწარ შეთანხმებულია პროგრამული სისტემის მომხმარებელთან. მომხმარებელს, რომელმაც არ იცის მონაცემთა ბაზასთან მუშაობა დაპროგრამების ენების გამოყენებით (მაგალითად, DDL და DML), მაშინ მისთვის უნდა აიგოს დიალოგში სამუშაო ინტერფეისი, რომელიც ადვილად შეასრულებს ბაზაში მონაცემების ძებნის (Select) ან ცხრილების განახლების ოპერაციებს (Add, Update, Delete).

მე-5 ნახაზზე ნაჩვენებია ასეთი ინტერფეისის ერთ-ერთი ვარიანტი, რომელშიც შესაძლებელია მომხმარებელმა, როგორც ბაზის ადმინისტრატორმა, განახორციელოს თავისი ფუნქციები (რეალურ მონაცემთა ბაზასთან უშუალოდ წვდომის გარეშე), ამაჯერად, მაგალითად, კლიენტების ბაზასთან.



ნახ.5. ინტერფეისის საილუსტრაციო მაგალითის მაკეტი

ამ ფორმის დიზაინი აიგება Visual Studio.NET გარემოში WPF ინსტრუმენტების პანელის და XAML-ენის გამოყენებით. შესაბამისი კოდის ფრაგმენტი მოცემულია 1-ელ ლისტინგში.

```
<-- ლისტინგი-1: XAML ფაილი ----->
<Window x:Class="WpfAppSQL_ListView.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title ="ტვირთების კლიენტების სია"
  Height="375" Width="520" Loaded="Window_Loaded" Background="White">
  ...
  <GridView>
    <GridViewColumn Header="კლიენტის ID" DisplayMemberBinding=
      "{Binding Path=ClientID}"></GridViewColumn>
    <GridViewColumn Header="გვარი" DisplayMemberBinding=
      "{Binding Path=Name}"></GridViewColumn>
    <GridViewColumn Header="მისამართი" DisplayMemberBinding=
      "{Binding Path=Adress}"></GridViewColumn>
  </GridView>
  ...
</Window>
```

ინტერფეისის ფორმაზე ჩანს LisBox სამი სვეტით კლიენტების სიაში რამდენიმე მონაცემის გამოსატანად, სამი TextBox, საძებნი ან დასამატებელი სტრიქონის მონაცემების ჩასაწერად და ოთხი Button, ბაზაში სხვადასხვა ფუნქციების შესასრულებლად (Add, Update, Delete). Clear ღილაკი გამოიყენება მონაცემთა შესატანი ტექსტობოქსების გასაწმენდად.

საჭიროა პროგრამის მუშაობის ლოგიკის დაპროგრამება C#.NET ენაზე, კერძოდ უნდა დაიწეროს მეთოდები, რომლებიც მოემსახურება შესაბამისი ღილაკების ამოქმედებით ინიციალიზირებული მოვლენების დამუშავებას. განვიხილოთ მაგალითები.

დავამატოთ მონაცემთა ბაზას ახალი კლიენტი (ნახ.ნ-ა). უნდა შევივსოს კლიენტის-გვარი, კლიენტისID და მისამართი.

კლიენტისID	გვარი	მისამართი
100	რეტერი ვადიმ	ვაშინგტონი, აშშ
103	ისაკაძე სეით	ბათუმი, საქართველო
111	ცნობილაძე გია	პარიზი, საფრანგეთი
133	პოჩოვიან სიმონ	ერეზუნი, სომხეთი
150	რაციმორ ვლადიმერ	სან-ფრანცისკო, აშშ
155	ქაჩიბაია ვახტანგ	ტორონტო, კანადა

კლიენტის გვარი: ზალდასტანიშვილი კოტე ClientID: 200

მისამართი: ვენა, ავსტრია

Buttons: Add, Update, Delete, Clear

ნახ.ნ-ა. ახალი ჩანაწერის დამატება (Add ღილაკით)

კლიენტისID	გვარი	მისამართი
100	რეტერი ვადიმ	ვაშინგტონი, აშშ
103	ისაკაძე სეით	ბათუმი, საქართველო
111	ცნობილაძე გია	პარიზი, საფრანგეთი
133	პოჩოვიან სიმონ	ერეზუნი, სომხეთი
150	რაციმორ ვლადიმერ	სან-ფრანცისკო, აშშ
155	ქაჩიბაია ვახტანგ	ტორონტო, კანადა
200	ზალდასტანიშვილი	ვენა, ავსტრია

კლიენტის გვარი: ClientID:

მისამართი:

Buttons: Add, Update, Delete, Clear

ნახ.ნ-ბ. ახალი ჩანაწერი დაემატა და ტექსტბოქსები გაიწმინდა

```
Add – რილაკის პროგრამის ტექსტი მოცემულია მე-2 ლისტინგში.
// --- ლისტინგი_2 – Add -----
private void btnAdd_Click(object sender, RoutedEventArgs e)
{
    string Name = textBox1.Text;
    string Adress = textBox2.Text;
    string ClientID = textBox3.Text;
    SqlConnection con = new SqlConnection(@"Data Source=GTU-205a-09;Initial
        Catalog=MultiModalTrasport;Integrated Security=True");
    con.Open();
    SqlCommand comm = new SqlCommand("insert into Client(Name,Adress, ClientID) values(@Name,
        @Adress, @ClientID)", con);
    comm.Parameters.AddWithValue("@Name", textBox1.Text);
    comm.Parameters.AddWithValue("@Adress", textBox2.Text);
    comm.Parameters.AddWithValue("@ClientID", textBox3.Text);
```



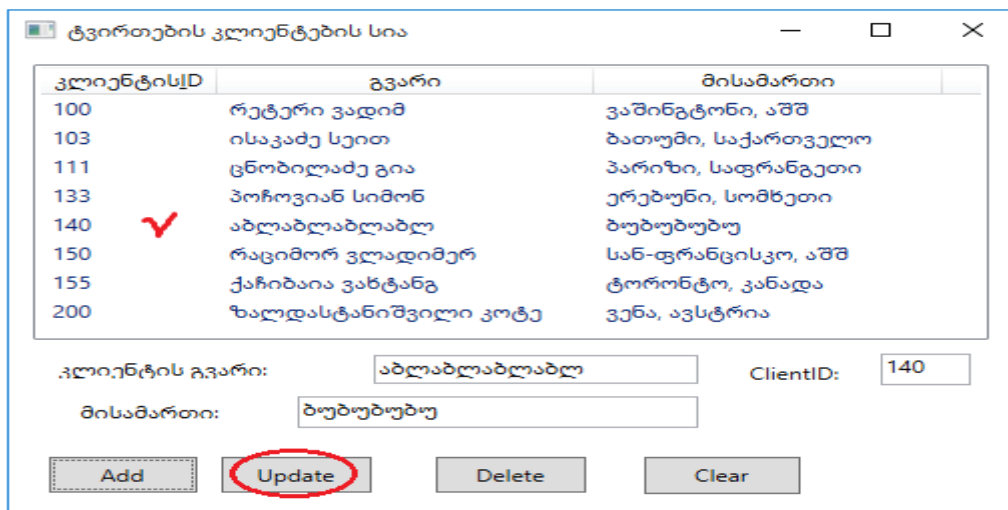
```
comm.ExecuteNonQuery();
con.Close();
ShowData();
}
```

შენიშვნა: SqlConnection სტრიქონით ხდება MultiModalTrasport მონაცემთა ბაზასთან ავტომატური დაკავშირება, ხოლო SqlCommand –ით ხორციელდება Insert ოპერაციის შესრულება. აქ ShowData() მეთოდი, რომელიც უზრუნველყოფს შედეგების ასახვას ინტერფეისის ფორმაზე; მისი შესაბამისი C#-პროგრამის კოდი ნაჩვენებია მე-3 ლისტინგში.

```
//--- ლისტინგი_3 --- ShowData() ----
public void ShowData()
{
    SqlConnection con = new SqlConnection(@"Data Source=GTU-205a-09;Initial
                                        Catalog=MultiModalTrasport;Integrated Security=True");
    con.Open();
    SqlCommand comm = new SqlCommand("Select ClientID, Name, Adress
                                     from Client", con);

    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(comm);
    da.Fill(dt);
    listView1.DataContext = dt.DefaultView;
}
```

მონაცემთა ბაზაში საჭიროა შეიცვალოს ინფორმაცია, შეცდომა 140-კლიენტის ჩანაწერში (ნახ.7-ა). საჭიროა Update ღილაკის დაპროგრამება, რომლის ტექსტი მე-4 ლისტინგშია მოცემული.



ნახ.7-ა. მონაცემების შეცვლა – ბაზის განახლება (Update ღილაკით)

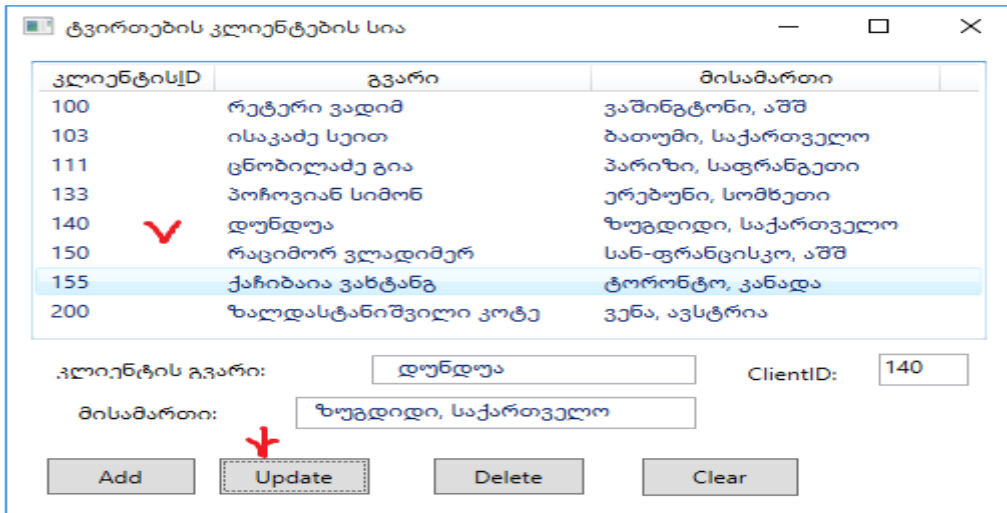
```
//--- ლისტინგი_4 --- Update -----
private void btnUpdate_Click(object sender, RoutedEventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
    {
        DataRowView drv = (DataRowView)listView1.SelectedItem;
        string id = drv.Row[0].ToString();
        SqlConnection con = new SqlConnection(@"Data Source=GTU-205a-09;Initial
                                            Catalog=MultiModalTrasport;Integrated Security=True");
        con.Open();
        SqlCommand comm = new SqlCommand("update Client set
                                         Name=@Name,Adress=@Adress where ClientID=@ClientID", con);
```

```

comm.Parameters.AddWithValue("@ClientID", id);
comm.Parameters.AddWithValue("@Name", textBox1.Text);
comm.Parameters.AddWithValue("@Adress", textBox2.Text);
comm.ExecuteNonQuery();
con.Close();
ShowData();
}
}

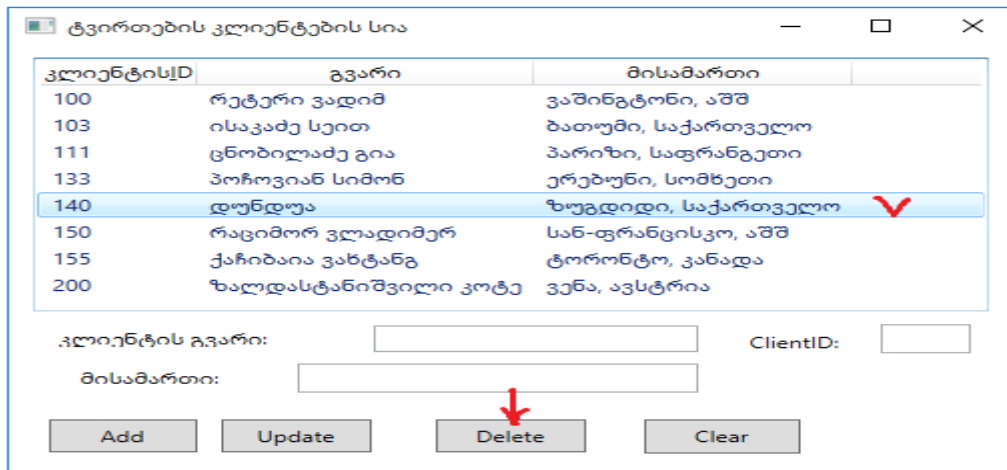
```

შედეგად მივიღებთ 7-ბ ნახაზზე მოცემულ სიტუაციას, ბაზაში 140-ე კლიენტის გვარი და მისამართი განახლებულია.



ნახ.7-ბ. მონაცემები შეიცვალა ბაზაში ჩაწერით

დავუშვათ, რომ გვჭირდება კლიენტის მონაცემების წაშლა, რომლის გვარია „დუნდუა“. უნდა მოვნიშნოთ ეს სტრიქონი (ნახ.8-ა).



ნახ.8-ა. წასაშლელად მომზადებული სტრიქონი

ამჯერად უნდა დაპროგრამირდეს Delete ღილაკი. კოდის ტექსტი მოცემულია მე-5 ლისტინგში.

```

//-ლისტინგი_5 ---Delete -----
private void btnDelete_Click(object sender, RoutedEventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
    {
        DataRowView drv = (DataRowView)listView1.SelectedItem;
        string id = drv.Row[0].ToString();
    }
}

```

```

SqlConnection con = new SqlConnection(@"Data Source=GTU-205a-09;Initial
                                     Catalog=MultiModalTrasport;Integrated Security=True");
con.Open();
SqlCommand comm = new SqlCommand("delete from Client where
                                   ClientID=@ClientID", con);
comm.Parameters.AddWithValue("@ClientID", id);
comm.ExecuteNonQuery();
ShowData();
    }
}

```

ახალი მონაცემების შეტანის დროს Add ლილაკის ამოქმედების შემდეგ თუ ტექსტოქსებში დარჩა უკვე შეტანილი მონაცემები, მაშინ ისინი უნდა გასუფთავდეს, გამზადდეს ახალი მონაცემების შესატანად. ამის ფუნქციას ასრულებს Clear ლილაკი, რომლის კოდის ტექსტი მოცემულია მელისტინგში.

```

/-- ლისტინგი_6 --- Clear -----
private void btnClear_Click(object sender, RoutedEventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
}

```

3. დასკვნა

მცირე და საშუალო ბიზნესის ობიექტებისათვის, რომლის ერთ-ერთი აქტუალური და მზარდი მოთხოვნილების მაგალითია ტვირთების მულტიმოდალური გადაზიდვების ფორმები, აუცილებელია მართვის საინფორმაციო სისტემის მონაცემთა ბაზების აგების და მისი ადმინისტრირების ფუნქციების ავტომატიზაცია. ასეთი ამოცანის ეფექტური გადაწყვეტა შესაძლებელია დაპროექტების CASE- და დაპროგრამების ჰიბრიდული ტექნოლოგიებით. კერძოდ, შემუშავებულია მულტიმოდალური გადაზიდვების (გემი, რკინიგზა, ავტო- და საჰაერო ტრანსპორტი) საპრობლემო სფეროს კონცეპტუალური სქემათა დაპროექტების ობიექტ-როლური და არსთა დამოკიდებულების მოდელის (ORM/ERM) გამოყენება, მისი რეალიზაცია კი - Ms SQL Server პაკეტით. სერვის-ორიენტირებული სისტემების ასაგებად ასეთი ობიექტებისათვის მიზანშეწონილია ჰიბრიდული დაპროგრამების ტექნოლოგიების გამოყენება.

ლიტერატურა:

1. Surguladze Gia, Topuria N., Petriashvili L., Surguladze Giorgi. (2015). Modelling of Designing a Conceptual Schema for Multimodal Freight Transportation Information System. WASET, World Academy of Scientific, Engineering and Technology, v.9, N11. ISSN 1307-6892, Spain, pp. 204-207.
2. გოგიჩაიშვილი გ., სურგულაძე გიორგი. (2014). მულტიმოდალური გადაზიდვების ბიზნეს-პროცესების ავტომატიზებული მართვის კონცეფცია. სტუ-ს შრ.კრ. „მას“ 2(18). გვ.45-50.
3. Halpin T. (2005). ORM 2 Graphical Notation, Neumont University. http://www.orm.net/pdf/ORM2_TechReport1.pdf.
4. სურგულაძე გ., ვედეკინდი ჰ., თოფურია ნ. (2006). განაწილებული ოფის-სისტემების მონაცემთა ბაზების დაპროექტება და რეალიზაცია UML-ტექნოლოგიით. „ტექნიკური უნივერსიტეტი“, თბილისი.
5. სურგულაძე გ., თოფურია ნ., ბაკურია კ., ლომიძე მ. (2014). საინფორმაციო სისტემის დაპროექტება ობიექტ-როლური მოდელირების და სერვისორიენტირებული არქიტექტურის ბაზაზე. სტუ-ს შრ.კრ., „მას“. №1(17). გვ. 32-45.

6. Barker R. (1990). CASE Method: Entity Relationship Modelling. Reading, MA: Addison-Wesley Professional. ISBN 0-201-41696-4.

7. სურგულაძე გიორგი. (2015). მულტიმოდალური გადაზიდვების ბიზნესპროცესების მართვის სისტემის ინფრასტრუქტურა და მისი იმიტაციური მოდელი. სტუ-ს შრ.კრ.,მას“, 2(20). გვ.108-123.

8. სურგულაძე გ. (2014). კორპორაციული მენეჯმენტის სისტემების Windows დეველოპმენტი (WPF ტექნოლოგია). სტუ. „IT-კონსალტინგის ცენტრი“. თბილისი.

9. სურგულაძე გ., თურქია ე., თოფურია ნ. (2015). ჰიბრიდული აპლიკაციების დაპროგრამების ავტომატიზაცია. საერთაშ. სამეცნიერო კონფერენცია. „საინფორმაციო და კომპიუტერ. ტექნოლოგიები, მოდელირება და მართვა“. გვ. 69-73.

CONSTRUCTION OF MULTIMODAL FREIGHT FORWARDING MIS USING CASE DESIGN AND HYBRID PROGRAMMING TECHNOLOGIES

Gogichaishvili Giorgi, Surguladze Giorgi, Topuria Nino,
Petriashvili Lily, Surguladze Gia
Georgian Technical University

Summary

The present article discusses database design for Multimodal Freight Forwarding MIS system, development of software and user interfaces using CASE design and hybrid programming technologies. In particular, the article proposes conceptual schemes for multimodal freight forwarding (ship, rail, motor and air transport modes) problem areas with tables of Customer (owner of cargo), cargo (transportation object) and supplier – freight forwarder, with Object-Role and Entity Relationship modeling (ORM/ERM) instruments, developed using Ms SQL Server package. The article also discusses realization of interface for functional problem of database administrator within environment of hybrid programming (WPF, C#, XAML).

ПОСТРОЕНИЕ АСУ МУЛЬТИМОДАЛЬНЫМИ ПЕРЕВОЗКАМИ ГРУЗОВ НА ОСНОВЕ ТЕХНОЛОГИЙ CASE ПРОЕКТИРОВАНИЯ И ГИБРИДНОГО ПРОГРАММИРОВАНИЯ

Гогичаишвили Георгий, Сургуладзе Георгий, Топурия Нино,
Петриашвили Лили, Сургуладзе Гия
Грузинский Технический Университет

Резюме

Рассматриваются вопросы проектирования и программной реализации баз данных, построения интерфейсов пользователя АСУ мультимодальными перевозками грузов на основе технологий CASE проектирования и гибридного программирования. В частности, предлагаются концептуальные схемы (таблицы клиентов, грузов, поставщиков) проблемной области мультимодальной перевозки грузов (морской, железнодорожный, автомобильный и воздушный виды транспорта) с помощью инструментов Объектно-ролевого и Сущности-связи моделирования (ORM/ERM) в среде Visual Studio.Net и базы данных Ms SQL Server. Построен интерфейс функций системы ведения БД в интегрированной среде программирования (WPF, C#, XAML).