

Graful – modalitate de reprezentare a elementelor interfeței cu utilizatorul

Adrian Stănciulescu

Absolvent al Academiei de Studii Economice – București, Facultatea de Cibernetică,
Statistică și Informatică Economică

E-mail: adi_stanci@yahoo.com

Quentin Limbourg

Université catholique de Louvain, School of Management (IAG)

E-mail: limbourg@isys.ucl.ac.be

Jean Vanderdonckt

Université catholique de Louvain, School of Management (IAG)

E-mail: vanderdonckt@isys.ucl.ac.be

Rezumat. Lucrarea descrie modul în care interfețele grafice cu utilizatorul pot fi reprezentate sub forma unui graf astfel încât fiecare element al interfeței (ferestră, buton, meniu, etc.) să fie reprezentat printr-un nod al grafului, iar dependențele dintre elemente să fie reprezentate prin arcele grafului. Structura grafului astfel realizat este codificată cu ajutorul unui limbaj de descriere a interfeței cu utilizatorul (USIXML).

Cuvinte cheie: graf, limbaj de descriere a interfeței cu utilizatorul.

1. INTRODUCERE

De-a lungul anilor, domeniul Interacțiunii Om-Calculator a fost martorul unei curse continue pentru realizarea unui Limbaj de Descriere (LD) a Interfeței cu Utilizatorul (LDIU), limbaj al cărui scop este de a evidenția esența a ceea ce interfața cu utilizatorul ar putea fi sau ar trebui să fie. Multe echipe, atât din lumea industriei cât și din cea academică au realizat LDIU de-a lungul timpului. LDIU a apărut pentru prima oară atunci când oamenii au încercat să specifice aspectul unei IU. Aceasta încercare a fost deosebit de intensă în momentul în care software-ul realizat producea IU pentru multiple platforme, asigurând astfel o formă de portabilitate a IU. Apoi, aceste încercări aproape că au dispărut până în momentul în care XML a fost considerat suficient de complex pentru a iniția o nouă familie de LDIU. Astfel de LDIU pot urmări diferite scopuri [1]:

- asigurarea portabilității IU de la o platformă la alta, păstrând consistența identificarea cerințelor IU pentru a da o definiție abstractă ce rămâne stabilă de-a lungul timpului
- îmbunătățirea reutilizării proiectării IU
- realizarea unei singure IU pentru diferite dispozitive, platforme
- posibilitatea de extensibilitate și adaptabilitatea a IU
- folosirea unei descrieri de IU pentru generarea automată de cod sursă al IU

În zilele noastre, există câteva LDIU dar folosirea lor este limitată la câteva domenii specifice. Chiar dacă au existat câteva inițiative de realizarea a unor standarde nu s-a ajuns la nici un consens, astfel încât identificarea “celor mai bune practici” și a “celor mai bune metode de proiectare” folosind XML pentru dezvoltarea și descrierea unei IU rămâne o problemă de rezolvat.

Pentru a descrie o parte din proprietățile cele mai importante ale Interfețelor Grafice cu Utilizatorul, IU pentru aplicații Web, interfețe bazate pe limbaj, IU de tip multimodal și bazate realitate virtuală, a fost realizat un alt LDIU ce poartă numele User Interface eXtensible Markup Language (USIXML).

2. USIXML – LDIU PENTRU INTERFEȚE CU UTILIZATORUL SENSIBILE LA CONTEXT

În ultimii ani s-a observat apariția pe piață a multor tipuri de calculatoare și dispozitive. Pentru a-și realiza sarcinile, oamenii au acum la dispoziție o largă varietate de dispozitive de calcul ce variază de la telefoane celulare, PDA-uri până la televiziune furnizată prin Internet (WebTV). Utilizatorii doresc să fie capabili să acceseze informația și serviciile indiferent de dispozitivul pe care îl folosesc, chiar și atunci când sistemul sau mediul se schimbă în mod dinamic. Pentru a realiza acest lucru, aplicațiile informatice trebuie realizate astfel încât să poată fi rulate pe o mare varietate de dispozitive și este necesar ca ele să se adapteze precum unameleon la contextul de utilizare aflat într-o continuă modificare. Pentru dezvoltatorii de software, acest lucru cauzează o serie de dificultăți datorate nevoii de realizare a unor versiuni multiple pentru o singură aplicație. Pentru rezolvarea acestor probleme este nevoie ca aceste versiuni să fie capabile să răspundă în mod dinamic la schimbări de context precum conectivitate la rețea, localizarea utilizatorului, tipul de dispozitiv folosit [1].

USIXML are ca scop descrierea IU ce au nivele diferite de detalii și abstractizări, depinzând de contextul de folosire. USIXML ia în considerare o familie de IU cum ar fi (dar nu limitată la acestea): independente de dispozitiv, independente de platformă, independente de context [4].

USIXML permite specificarea unor modele multiple implicate în proiectarea IU, precum: modelul sarcinii, modelul domeniului, modelul prezentării, modelul dialogului, modelul contextului de folosire, care sunt în schimb descompuse în modelul utilizatorului, modelul platformei, modelul mediului. Pentru realizarea legăturilor dintre aceste modele a fost introdus și un model pentru *mappare* inter-modele ce acoperă ingineria inversă și directă, precum și tranzițiile de la un context de folosire la altul.

USIXML este structurat conform celor patru nivele abstracte de bază definite în cadrul proiectului Cameleon, proiect care are ca scop dezvoltarea în cicluri de viață a IU pentru aplicații interactive sensibile la context [4]:

1. Modelul Interfeței Finale cu Utilizatorul (IFU) : se află la baza nivelurilor abstracte și face referire la orice IU ce rulează pe o anumită platformă de calcul, fie prin interpretare (ex., de către un Web browser), fie prin execuție (ex., după compilarea unui cod sursă într-un mediu interactiv de dezvoltare).

2. Modelul Interfeței Concrete cu Utilizatorul (ICU): abstractizează un IFU într-o definiție a unei IU care este independentă de orice platformă de calcul. Deși o ICU definește în mod explicit aspectul final al IFU, ea este deocamdată o machetă ce rulează numai în anumite medii particulare.

3. Modelul Interfeței Abstracte cu Utilizatorul (IAU): abstractizează o ICU într-o definiție a unei IU care este independentă de orice modalitate de interacțiune (ex., interacțiune grafică, interacțiune vocală, recunoașterea vorbirii, interacțiune bazată pe imagine video. În același timp, IAU este un model care reprezintă o expresie canonică a funcțiilor și conceptelor domeniului redat, independentă de orice modalitate de interacțiune.

4. Modelul sarcinii & conceptelor: se află în vârful nivelurilor abstracte ; sarcinile interactive realizate de utilizatorul final sunt definite potrivit punctului lui de vedere, împreună cu diferite obiecte manipulate de aceste sarcini. Aceste obiecte sunt considerate instanțe ale claselor ce reprezintă conceptele manipulate.

Modelul ICU cuprinde o serie de elemente dintre care enumerăm:

cuiModel: model ce reprezintă concretizarea a unui model IAU. Un cuiModel este populat cu obiecte de interacțiune concrete (**cio**) și relații între obiecte de interacțiune concrete (**cui_relationship**).

cio: o entitate a IU pe care utilizatorul o poate percepe și manipula (text, imagine, animație, buton, list box, check box) și care se poate găsi în cea mai mare parte a utilităților grafice (Java AWT/Swing, HTML 4.0, Flash DRK6).

cui_relationship: este o relație ce leagă două sau mai multe cio. Poate face referire la **graphical_Relationship** sau **auditory_relationship**.

graphical_cio: este un element format din elemente grafice ale IU. Poate fi un container sau o componentă individuală.

Exemple de *graphical_cio*: *Text_outputter*, *Label*, *Video_outputter*, *Button*, *Menu*, *EditBox*, *Canvas*, *Checkbox*, *Radiobutton*, *Window* (*dialogBox*, *frame*, *tabbed_dialogBox*).

graphical_Relationship este o relație ce leagă două sau mai multe *graphicalCio*. Elemente ale *graphical_Relationship*:

Topological_placement(*balancing*, *alignment*, *adjacency*, *containment*)

Graphical_container_transition(*close*, *give_focus*, *maximize*, *minimize*, *open*).

AGG – ATTRIBUTED GRAPH GRAMMAR SYSTEM

AGG este un interpretor realizat în limbajul Java utilizat pentru transformarea de grafuri ale cărui noduri au asociate atribute multiple [5].

AGG este un limbaj vizual bazat pe reguli ce suportă o abordare algebrică a transformărilor de grafuri. Are rolul de a specifica și a prototipiza în mod rapid aplicații ce conțin date complexe reprezentate sub formă de grafuri.

Caracteristici ale AGG-ului:

- structurile de date complexe sunt modelate sub forma unor grafuri ale căror tip este precizat prin **tipul grafului** (*type graph*)
- comportamentul sistemului este specificat prin **regulile grafului** (*graph rule*) folosind un tip de descriere de tip **daca-atunci** (*if-then*)
- caracteristicile AGG-ului prevăd reguli ce conțin condiții de aplicare negative pentru a exprima cerințele asociate unei substructuri neexistente
- aplicarea unei reguli transformă structura grafului
- grafurile reprezentate pot conține atribute de tipul obiectelor existente în limbajul Java. Tipurile de date de baza ca și obiectele aparținând claselor disponibile în bibliotecile Java pot fi folosite
- regulile grafului pot fi atribuite prin expresii Java care sunt evaluate pe parcursul aplicării regulii
- regulile pot avea atribute de tip condiție ce sunt expresii booleene Java

Utilitarul oferă editoare grafice pentru manipularea de grafuri și stabilirea de reguli precum și un editor textual integrat pentru expresii Java. Se oferă de asemenea suport pentru interpretare vizuală și validare.

După reprezentarea structurilor de date complexe sub formă de graf, utilitarul oferă posibilitatea salvării structurii sub forma unui fișier cu extensia .ggx. Acest fișier text conține informații referitoare la structurile de date complexe reprezentate sub formă de taguri stabilite prin regulile XML.

Codul sursă scris în limbajul Java este cod deschis ceea ce oferă posibilitatea re(utilizării) AGG-ului pentru realizarea de aplicații ce au la bază grafurile [5].

Astfel a apărut ideea reprezentării elementelor componente ale unei IU și a relațiilor dintre ele sub forma unui graf. Plecând de la această idee și bazându-ne pe codul deschis al utilitarului AGG, acesta a fost îmbunătățit prin implementarea unor noi funcțiuni:

Funcția de export – din formatul XML specific utilitarului AGG în formatul XML descris prin intermediul limbajului USIXML

Funcția de import – din formatul XML descris prin intermediul limbajului USIXML în formatul XML specific utilitarului AGG

4.1. Funcția de export în AGG

Ca urmare a reprezentării IU sub forma unui graf, AGG oferă posibilitatea salvării structurii într-un fișier de tip .ggx. Pentru a putea utiliza acesta structură conform elementelor definite prin USIXML a apărut nevoia implementării unei noi funcții în cadrul AGG. Această funcție realizează exportul unui fișier de tip .ggx într-un fișier în care structura IU este salvată conform specificațiilor USIXML.

În figura 1 se prezintă spre exemplificare următoarea structură a grafului ce reprezintă o IU simplă. Conținutul IU este dat de o fereastră principală care are asociat un meniu și o suprafață de lucru ce include o căsuță de editare (editbox) și un buton de calcul. Elementele IU (nodurile grafului) au asociate atribute și valori ale atributelor caracteristice fiecărui element în parte.

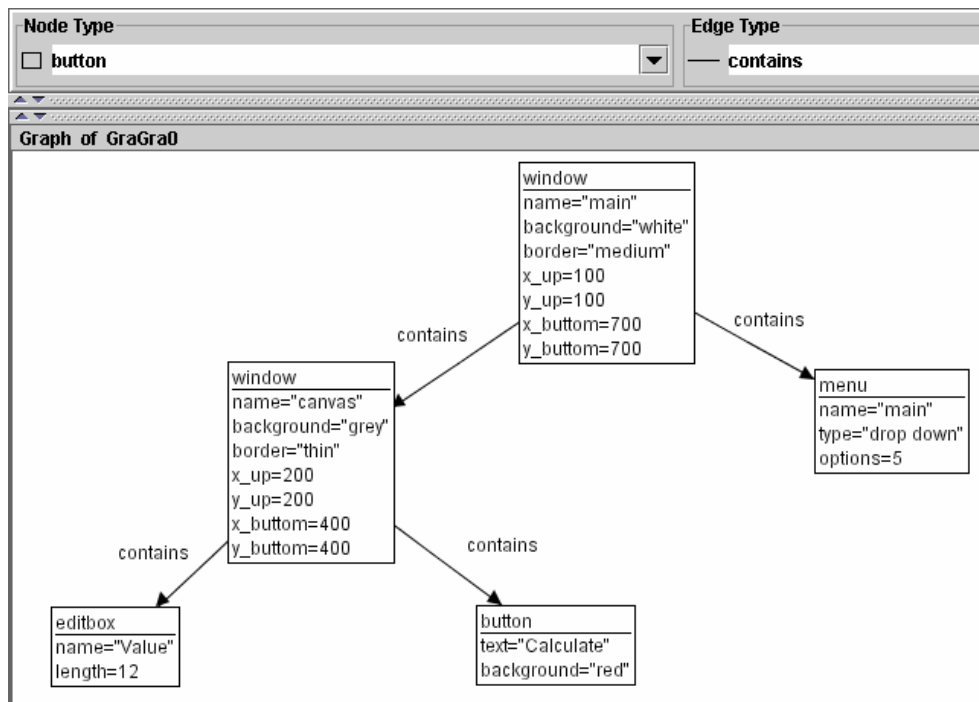


Figura 1. Graful elementelor componente ale IU

Implementarea funcției de export determină o modificare a formatului XML specific utilitarului AGG atât din punct de vedere sintactic cât și din punct de vedere semantic. Se obține astfel un format XML corespunzător specificațiilor limbajului USIXML.

Figura 2 descrie IU reprezentată în figura 1 în formatul XML corespunzător specificațiilor limbajului USIXML:

```

<?xml version="1.0" encoding="UTF-8"?>
<Cameleon_document version="0.33">
  <cui_model ID="I1" name="Graph">
    <cio ID="I3">
      <window background="white" border="medium" name="main"
x_bottom="700" x_up="100" y_bottom="700" y_up="100"/>
      <NodeLayout X="389" Y="73"/>
    </cio>
    <cio ID="I19">
      <window background="grey" border="thin" name="canvas"
x_bottom="400" x_up="200" y_bottom="400" y_up="200"/>
      <NodeLayout X="195" Y="204"/>
    </cio>
    <cio ID="I28">
      <menu name="main" options="5" type="drop down"/>
      <NodeLayout X="576" Y="177"/>
    </cio>
    <cio ID="I36">
      <editbox length="12" name="Value"/>
      <NodeLayout X="67" Y="325"/>
    </cio>
    <cio ID="I42">
      <button background="red" text="Calculate"/>
      <NodeLayout X="356" Y="324"/>
    </cio>
    <cui_relationship ID="I48">
      <contains/>
      <cui_relationship_source source="I3"/>
      <cui_relationship_target target="I19"/>
      <EdgeLayout bendX="292" bendY="139" textOffsetX="-17"
textOffsetY="-27"/>
    </cui_relationship>
    <cui_relationship ID="I50">
      <contains/>

```



```

        <cuirelationship_source source="13"/>
        <cuirelationship_target target="128"/>
        <EdgeLayout bendX="482" bendY="125" textOffsetX="14"
textOffsetY="-23"/>
    </cuirelationship>
    <cuirelationship ID="152">
        <contains/>
        <cuirelationship_source source="119"/>
        <cuirelationship_target target="136"/>
        <EdgeLayout bendX="131" bendY="265" textOffsetX="-39"
textOffsetY="-10"/>
    </cuirelationship>
    <cuirelationship ID="154">
        <contains/>
        <cuirelationship_source source="119"/>
        <cuirelationship_target target="142"/>
        <EdgeLayout bendX="275" bendY="264" textOffsetX="46"
textOffsetY="-14"/>
    </cuirelationship>
</cuirelationship>
</cuirelationship>
</Cameleon_document>

```

Figura 2. IU reprezentată în format USIXML

4.2. Funcția de import în AGG

În zilele noastre IU suferă dese schimbări ale aspectului și conținutului datorate atât necesității trecerii de la o platformă la alta, de la un dispozitiv la altul, cât și nivelului de cunoștințe al diferitelor categorii de utilizatori. De aceea a apărut necesitatea implementării funcției de import în cadrul utilitarului AGG.

Astfel, implementarea funcției de import oferă posibilitatea trecerii de la reprezentarea unui fișier descris conform specificațiilor USIXML la o reprezentare sub forma unui graf (figura 1).

Importanța acestei funcții apare în momentul în care se dorește modificarea elementelor sau a atributelor caracteristice unei IU și se dispune de fișierul salvat în format USIXML, întrucât modificarea este mult mai ușor de realizat pe o structura grafică decât pe una textuală.

Recunoaștere

Suntem recunoscători suportului acordat de către proiectul de cercetare Cameleon (<http://giove.cnuce.cnr.it/cameleon.html>) aflat sub tutela European Fifth Framework Programme (FP5-2000-IST2) și rețelei de excelență SIMILAR (<http://www.similar.cc>), o forță Europeană de cercetare în domeniul dezvoltării de interfețe om-mașină similare comunicării om-om, din cadrul European Sixth Framework Programme (FP6-2002-IST1-507609).

Referințe

1. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., *CXML V1.2.0 Definition*, Cameleon Deliverable D1.3-1, 19 February 2004.
2. Limbourg, Q., Vanderdonckt, J.: *Transformational Development of User Interfaces with Graph Transformations*. In: Jacob, R., Limbourg, Q., Vanderdonckt, J. (eds.): Proc. of 5th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2004 (Madeira, January 14-16, 2004). Kluwer Academics Pub., Dordrecht (2004).
3. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L. and Lopez, V., *USIXML: a Language Supporting Multi-Path Development of User Interfaces*. PreProc. of 9th IFIP Working Conf. on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems EHCI-DSVIS'2004 (Hamburg, 11-13 July 2004). IFIP, Hamburg (2004).
4. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., Florins, M. and Trevisan, D., *USIXML: A User Interface Description Language for Context-Sensitive User Interfaces*. Proc. of the AVI'2004 Workshop "Developing User Interfaces with XML: Advances on User Interface Description Languages" USIXML'04 (Gallipoli, 25 May 2004). EDM-Luc, Diepenbeek (2004), 55–62.
5. Michael, R., Gabriele, T., *Introduction to the Language Concepts of AGG* (10 November 1999).