

ASISTENTI VIRTUAL NË AUTOMATIZIMIN E PROCESIVE

SIDORELA LAZI, ENDRI XHINA.

Universiteti i Tiranës, Fakulteti i Shkencave të Natyrës, Departamenti i Informatikës

e-mail: sidorela.lazi@gmail.com

Përmbledhje

Aplikacionet për automatizimin e proceseve po zgjerohen çdo ditë e më tepër duke thjeshtuar jetën e njerëzve në përgjithësi. Automatizimi i proceseve të biznesit ul shpenzimet, redukton kohën dhe zgjidh probleme të vështira. Megjithatë ato konsiderohen aplikacione komplekse, të cilat kërkojnë nivel të lartë ekspertize si proces dhe aplikacion software. Për më tepër përdorimi i këtyre aplikacioneve shoqërohet me nivel të lartë përgjegjësie, për shkak se shpesh ato duhet të garantojnë zbatimin e procedurave formale, të cilat përcaktojnë rrjedhën e punës, rolet që marrin pjesë, të drejtat që ka çdo rol, si dhe dokumentacionin që duhet të shoqërojë procesin në çdo hap. Përdoruesit fundorë shpesh kërkojnë garanci, që software i përmbahet dhe zbaton me përpikmëri procedurat e përcaktuara. Këto konkluzione rrjedhin nga një periudhë disa vjeçare në zbatimin e software-ve të kësaj natyre në institucione të ndryshme. Për këtë arsye, në këtë punim do prezantojmë 2 mjete të ndërtuara mbi software-n e automatizimit të proceseve "Smart Processes".

Fjalëkyçe: Automatizimi i proceseve, ndërfaqe miqësore, asistenti virtual.

Abstract

Process automation applications are expanding every day, simplifying people's lives in general. Automating business processes cuts costs, saves time, and solves difficult problems. However, they are considered as complex applications that require a high level of expertise like software process and application. The use of these applications is associated with high level of responsibility, because they often have to guarantee the implementation of formal procedures, which determine process workflow, participating roles, access rights for each role, and the documentation that should accompany the process at every step. Users often require assurance that the program strictly adheres to and complies with the defined rules. These conclusions come from a period of several years in the implementation of such software at different institutions. For this reason, in this paper we will present 2 tools built on process automation software "Smart Processes".

Key words: Process automation, user-friendly interface, virtual assistant.

Hyrje

Automatizimi i proceseve ndihmon në përmirësimin dhe efikasitetin e rrjedhës së punës. Në vend që të harxhohet kohë për të ndjekur hap pas hapi zhvillimin e procesit, mund të arrihet një menaxhim i mirë i procesit i bazuar në teknologji. Kjo qasje ndihmon të vendoset më shumë kontroll mbi procese të ndryshme në një kohë të reduktuar dhe me pak kosto ekonomike.

Për të përmbushur presionet kohore dhe afatet e nevojshme për të drejtuar një proces kompleks, softweri Smart Processes, mund të automatizojë rrjedhën e punës dhe të ndihmojë në menaxhimin e të dhënave hyrëse për të maksimizuar rezultatin përfundimtar të procesit.

Duke përdorur Smart Processes, ofrohet mundësia të ndërtohen flukse pune komplekse të cilat shkojnë shumë përtej funksionalitetit standart dhe gjithashtu japin fleksibilitet për të pasur vetë kontroll mbi proceset.

Asistenti virtual duhet të mundësojë automatizimin e proceseve duke ofruar:

- Vizualizimin e metadata-ve të proceseve me terminologji jo teknike dhe lehtësisht të kuptueshme nga përdoruesi.
- Ndërtimin e një ndërfaqeje miqësore të organizuar në formën e një interviste të asistuar për konfigurimin e një procesi.
- Ndërtimin e një ndërfaqeje miqësore për përcaktimin e seksionit të të dhënave. Ndërtimin e një ndërfaqeje miqësore për menaxhimin e nivelit të aksesit.

Materiali dhe metodat

Për realizimin e këtyre objektivave u ndërtua një modul i ri në .Net, i cili përfshin funksionalitete për çdo njëri nga objektivat.

Në përfundim të krijimit të një procesi, përdoruesi drejtohet tek një udhëzues për konfigurimin e procesit, i cili përveç të dhënave bazë trajton konfigurimin e disa parametrave më të ndërlikuar.

Më poshtë, përshkruhen me rradhë konceptimi i ndërfaqeve dhe realizimit të tyre duke theksuar disa çështje teknike, të cilat ilustrjnë disa zgjidhje të ripërdorshme.

Ndërtimi i ndërfaqes për konfigurimin e seksionit të të dhënave

Nga vlerësimet e mbledhura nga përdorues të Smart Processes, në rolin e dizenjuesit dhe konfiguruesit të procesit, hedhja e të dhënave nëpërmjet dizenjuesit të formave, konsiderohet si një mënyrë e cila ndikon në humbjen e pamjes panoramike mbi të gjithë procesin.

Krijimi i seksioneve me anë të dizenjuesit të formave shoqërohet me krijimin në të njëjten kohë të kontrolleve për hedhjen e të dhënave dhe kjo e bën një proces kompleks.

Për këtë ndërfaqja e re i lejon dizenjuesit të procesit të përcaktojë seksionet e hedhjes së të dhënave, pa përcaktuar detaje mbi përmbajtjen e tyre, duke e lejuar mbajtjen e fokusit në pamjen panoramike të procesit.

Seksionet e të dhënave referohen në momentin e përcaktimit të të drejtave të aksesit në përmbajtjen e procesit dhe në përcaktimin e vendosjes së butonave që mundësojnë avancimin e praktikës në rrjedhën e punës.

Për ndërtimin e kësaj ndërfaqeje janë përdorur .Net, Css, JavaScript.

Rrjedha e punës

Konfiguratori është konceptuar në formën e një pyetësoi, i cili i kërkon përdoruesit: hapat në të cilat do të kalojë procesi, rolet që kanë të drejtë të ekzekutojnë këto hapa, në cilat seksione do të vendosen kontrollet për ekzekutimin e hapave.

Gjithashtu është realizuar dhe një kontroll i cili bën bredhjen e pemës së gjëndjeve për të evidentuar degëzime të pa finalizuara, në të cilat asnjë rol nuk ka të drejtë, të avancojë praktikën më tej, situata kur mungon kontrolli për ekzekutimin e një hapi në ndërfaqe.

Për të bredhur pemën e gjëndjeve janë përdorur komandat rekursive në një bazë të dhënash relacionale si SQL Server.

Hapat ruhen në një tabelë me relacion të natyrës prind-bir ku përdoret një funksion rekursiv për listimin e përmbajtjes së kësaj tablele sipas niveleve të pemës, duke realizuar bredhjen në gjerësi.

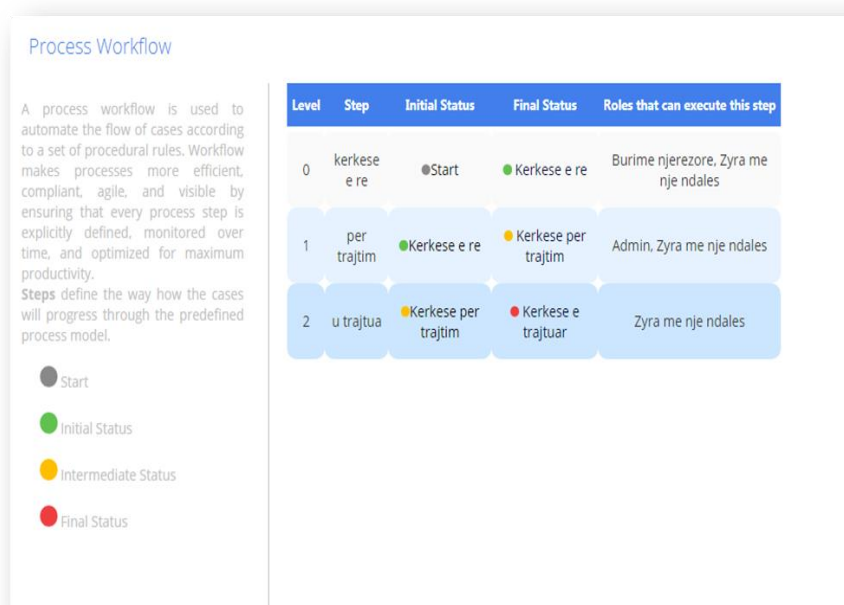


Figura 1: Ndërfaqja grafike për realizimin e rrjedhës së punës

```

create function [utils].[processWorkflowLayered]
(
  @processId INT
)
RETURNS TABLE

AS
RETURN

WITH workflow
(processID,StepID,StepName,InitialStatus,InitialStatusName,FinalStatus,FinalStatusName,
InitialStatusduration,FinalStatusDuration,initialstatustype,finalstatustype,
initialstatusdescription,finalstatusdescription, Level)
AS
(
  -- Anchor member definition
  SELECT e.ProcessId, e.StepID, e.StepName, e.InitialStatus,e.InitialStatusName,
  e.FinalStatus,e.FinalStatusName,e.InitialStatusduration,e.finalstatusduration ,
  initialstatustype,finalstatustype,initialstatusdescription,finalstatusdescription, 0 AS Level
  FROM utils.ExtendedSteps e
  where InitialStatus IS NULL and ProcessId=@processId
  UNION ALL

  -- Recursive member definition
  SELECT e.ProcessId, e.StepID, e.StepName, e.InitialStatus,e.InitialStatusName,
  e.FinalStatus,e.FinalStatusName,e.InitialStatusduration,e.finalstatusduration ,
  e.initialstatustype,e.finalstatustype,e.initialstatusdescription,e.finalstatusdescription,
  Level + 1
  FROM utils.ExtendedSteps AS e
  INNER JOIN
  workflow AS w
  ON w.FinalStatus= e.InitialStatus
)
-- Statement that executes the CTE
SELECT *
FROM workflow
where processID=@processId

```

Figura 2: Funkzioni rekursiv për bredhjen në gjerësi të pemës

Rrjedha e punës përcaktohet nga hapat në të cilat kalon procesi. Një hap përbëhet nga gjëndja fillestare ku ndodhet procesi dhe gjëndja fundore ku do të kalojë procesi pas ekzekutimit të hapit përkatës.

Bredhja fillon nga niveli zero ku procesi ndodhet në gjëndjen fillestare (statusi null). Më pas vazhdon duke bredhur gjëndjet jo fundore në nivelin e parë, dhe në mënyrë rekursive bredh të gjithë nivelet e tjera të pemës, deri sa të mos ketë më gjëndje jo fundore dhe komanda të mos kthejë më rreshta.

Funksionet rekursive janë të veçantë pasi ata i referohen vetes. Ky funksion është përdorur për shkak të kësaj aftësie të veçantë, për të punuar me të dhëna hierarkike siç është bredhja e pemës në gjerësi, e cila nuk mund të realizohej me komanda të tjera.

Përcaktimi i nivelit të aksesit

Ndërtimi i modulit të aksesit është një funksionalitet i kërkuar nga dizenjuesit e procesit për shkak se volumi i punës është i konsiderueshëm dhe shpesh bëhet burim gabimesh. Përdoruesi duhet të klikojë shumë herë për të përcaktuar nivelin e aksesit në disa dritare që hapen e mbyllen.

Smart Processes kërkon që të përcaktohet niveli i aksesit mbi rastet e një procesi, për çdo rol, në çdo gjëndje të procesit, për çdo seksion të përmbajtes. Bazuar në 8 parimet e dizenjimit të ndërfaqes së përdoruesit (Shneiderman *et al.*, 2015), është krijuar një ndërfaqe miqësore, e thjeshtë për tu përdorur dhe nuk e lejon përdoruesin të humbasë fokusin.

Ndërfaqja e re paraqet një hartë të të drejtave në një pamje të vetme, e cila i lejon konfiguruesit të procesit ta shikojë këtë lehtësisht. Për realizimin e kësaj faqeje u përdor një teknikë tjetër që kombinon disa gjuhë.

Konkretisht të dhënat merren nga baza e të dhënave në format XML, të cilat përshtaten nëpërmjet XSLT në një tekst HTML duke vendosur dëgjues ngjarjesh në JavaScript.

```

<Statuses>
  <Status processid="9200" StatusID="9842" StatusName="Kerkese e re">
    <Role RoleId="9257" RoleName="Zyra me nje ndales" PPT="6">
      <Pages>
        <Page PageId="15053" PageName="Informacion personal" PLA="4" />
        <Page PageId="15054" PageName="Dokumentat per aplikim " PLA="2" />
        <Page PageId="15055" PageName="Eksperienat e punesimit " PLA="4" />
        <Page PageId="15056" PageName="Arsimi dhe trajnimet " PLA="4" />
        <Page PageId="15057" PageName="Aftesite personale dhe kompetencat " PLA="2" />
      </Pages>
    </Role>
    <Role RoleId="9258" RoleName="Burime njerezor" PPT="6">...</Role>
    <Role RoleId="10297" RoleName="Admin" PPT="2">...</Role>
  </Status>

```

Figura 3: Struktura XML e nivelit të aksesit

```

<xsl:for-each select="Statuses/Status">
  <xsl:variable name="processid" select="@processid"></xsl:variable>
  <h4>
    Access rights when case is in status <i>
      <xsl:value-of select="@StatusName"/>
    </i></h4>
  <xsl:for-each select="Role">
    <br/>
    <table class="processrights" >
      <tr>
        <td class="RoleAccess">
          <xsl:value-of select="@RoleName"/>
        </td>
      </tr>
      <tr>
        <td>
          <table>
            <tr>
              <xsl:variable name="ProcessPermission" select="@PPT"></xsl:variable>
              <td>
                <xsl:variable name="roleid" select="@RoleId" > </xsl:variable>
                <xsl:variable name="statusid" select="..@StatusID" > </xsl:variable>
                <td>
                  <xsl:variable name="aright" select="@PPT"></xsl:variable>
                  <div class="rights" id="r{$roleid}_{$statusid}_{$aright}" style="display: none;" onClick="load2({$roleid},{
                    <xsl:value-of select="@PPT"/>
                  </div>
                  <a id="a{$roleid}_{$statusid}_0" onclick=" updateAccesspermission({$roleid},{$statusid},{0},{$processid})"
                </td>

```

Figura 4: Struktura XSLT për vizualizimin e nivelit të aksesit

Për lehtësi është përdorur një librari e JavaScript-it, JQuery për selektimin e elementeve të ndërfaqes nëpërmjet klasave dhe id-ve si dhe për ndërveprimin me AJAX, Handlers në Server-Side.

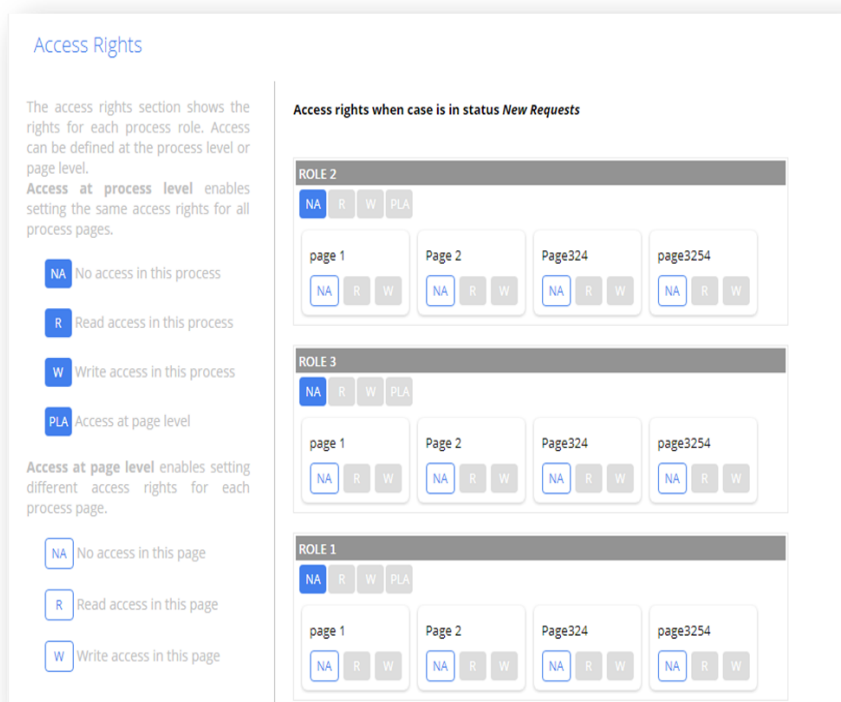


Figura 5: Ndërfaqja për përcaktimin e nivelit të aksesit

Përdoruesi duhet të përcaktojë të drejtat në nivelin e rolit, për secilën gjëndje në të cilën kalon procesi, për çdo seksion të të dhënave.

Në rastin më sipër përcaktohen të drejtat e aksesit kur procesi ndodhet në gjëndjen Start, për rolin 1, rolin 2 dhe rolin 3.

Secili nga këto role mund të mos ketë asnjë të drejtë, mund të ketë të drejta shkrimi, leximi ose të drejta specifike për çdo seksion.

Akësi i të drejtave ndryshon për çdo rol:

- Kur nuk ka asnjë të drejtë, zgjidhet NA (No Access).
- Kur ka vetëm të drejtë leximi në çdo seksion, zgjidhet R (Read Access)
- Kur ka vetëm të drejtë shkrimi në çdo seksion, zgjidhet W (Write Access)
- Kur ka të drejta specifike për çdo seksion, zgjidhet PLA (Page Level Access).

Duke patur një pamje të plotë të përcaktimit të të drejtave të aksesit, përdoruesi mund t'a përcaktojë lehtësisht atë.

Përfundimet

Përmirësimi i ndërfaqes grafike vjen si rrjedhojë e dëgjimit të kërkesave të përdoruesve të aplikacionit, duke ofruar lehtësi në përdorim dhe efikasitet në rrjedhën e punës. Kjo sjell dhe reduktimin e kohës dhe gabimeve të cilat ndodhin shpeshherë si pasojë e fluksit të madh në zhvillimin e procesit të punës.

Për ndërtimin e ndërfaqeve të reja janë përdorur disa teknika ku do përmendim: Përdorimi i komandave rekursive në një bazë të dhënash relacionale si SQL Server, XML për strukturimin e të dhënave, XSLT për transformimin e të dhënave XML, klasat e .NET për përpunimin e tyre (System.Xml, System.Xml.Xsl), JQuery, CSS, AJAX.

Literatura

Jesse James Garrett's, (2011): The Elements of User Experience: User-Centered Design for the Web and Beyond (2nd Edition)

Peter Morville and Louis Rosenfeld's. (2007): Information Architecture for the World Wide Web: Designing Large-Scale Web Sites

Ben Shneiderman. (2015): Eight Golden Rules of Interface Design
User Interface Principles (microsoft.com)