**PÉCSI TUDOMÁNYEGYETEM   
MŰSZAKI ÉS INFORMATIKAI KAR  
VILLAMOSMÉRNÖKI SZAK**

**Devossa Bence**

**Komplex folyamatirányítási rendszer   
template alapú generálása**

**Pécs, 2016**

Tartalomjegyzék

[Előszó 4](#_Toc451809996)

[1. Bevezetés 5](#_Toc451809997)

[1.1 Az alapgondolat 5](#_Toc451809998)

[1.2 A sablonosítás 5](#_Toc451809999)

[2. A technológia alapjai 7](#_Toc451810000)

[2.1 Az XML 7](#_Toc451810001)

# Előszó

Szakdolgozatom témája egy olyan szoftver megalkotása volt, mellyel komplex folyamatirányítási rendszereket generálhatunk sablonok (angolul: template) alapján. A jelenlegi technika rohamos fejlődése és piaci versenyképesség fenntartása miatt szükséges, hogy ne ragadjunk le a gépies, mechanikus fejlesztési folyamatoknál, mint például egyes paraméterek vagy egyszerű logikai vizsgálatok kézi kikeresése és megváltoztatása. A helyzet kritikusabbnak tekinthető mikor a fent említett műveleteket a fejlesztőnek nem csak több fájlon át kell vezetnie, hanem esetleg rövid intervallumon belül érkezik rendkívül hasonló felépítéssel rendelkező rendszer iránt igény, ekkor ciklus újraismétlődik.

Idő és tesztelhető eszközök hiányában 3 platformra készítettem el a generálást minimális eszközszámmal, amivel ha még nem is generálok azonnal éles helyzetben alkalmazható programot, az elvet tökéletesen prezentálni tudom, és minimális módosításokkal, illetve az elérhető eszközök számának bővítésével rövid időn belül vállalati szinten is alkalmazható lenne a szoftver.

A téma több szempontból is szimpatikus volt számomra. Elsősorban azért, mert több helyen is alkalmazom az alappillérként alkalmazott XML nyelvet és ezáltal bővíthettem az ismereteimet benne, amely nem csak további tanulmányaim során lesz hasznos, hanem gyakornoki feladataim során is igen gyakran előkerül. A második fő szempont az ismereteim bővítése és a lehetőségeim számának növelése volt, ugyanis beágyazott mikroszámítógépes rendszerek szakirányon a programozható logikai vezérlők programozását nem oktatják olyan részletesen, viszont így ebbe a témába is sikerült kicsit jobban belelátnom és amennyiben olyan munkakörben alkalmaznának, amelyben hasonló technológiával kellene dolgoznom, könnyebben meg tudnám állni a helyem.

*\*\*\**

Köszönet konzulensemnek a téma javaslatáért és tanácsaiért, melyekkel nem kis mértékben gyarapítottam tudásomat, illetve családomnak és barátaimnak, akik vég nélkül bíztattak minden egyes pillanatban tanulmányaim során.

Pécs, 2016. május 16.

*Devossa Bence*

# Bevezetés

## Az alapgondolat

A technológiával, mint definícióval kezdeném. Napjaink egyik legáltalánosabb, legtöbbször hivatkozott internetes enciklopédiája szerint:

„A technológia az ember által készített olyan célszerű, az egyéni (emberi) képességeit megnövelő eszközökről (például gépek, anyagok és eljárások) valamint azok alkalmazásáról szóló ismeretek gyűjtőneve, amelyek segítségével az emberiség egyre többet tud megismerni, megváltoztatni, megőrizni stb. az őt körülvevő világból.”

A szó, amit leginkább kiemelnék az fenti idézetből: célszerűség. Célszerű-e megkönnyíteni az embernek, a saját dolgát? Célszerű-e ezt megtenni, mikor a technológia rendelkezésére áll és minimális erőfeszítések révén munkáját nem csak percekkel, de hosszú távon akár órákkal is megrövidítheti? Egy olyan világban, ahol az ember pénzből él, az idő pénz és szeret az ember minél kevesebb idő alatt minél több pénzt keresni a válasz véleményem szerint triviális.

Mind a programozás, a számítástechnika és a mérnöki munkakör rendelkezik olyan ismétlődő, manuálisan megoldandó feladatokkal, mely során feltehetjük a fent említett kérdést. Az alábbi munkakörök viszont eltérnek az átlagtól, mivel mindháromra tanult, szakképzett embereket alkalmaznak, akiknek birtokában áll a tudás és a technológia ismerete ahhoz, hogy meg is valósítsák ezeket a könnyítéseket, kiskapukat. Egy ilyen probléma elemzéséről és egy lehetséges megoldásáról szól ez a szakdolgozat.

## A sablonosítás

Legyen szó bármilyen tevékenységről, csapatmunkáról, eszközről, a sikeresség, vagy eszköz esetében használhatóság nagyrészt a résztvevők, részegységek, részfolyamatok sikerén múlik. Amennyiben a folyamat egy résztvevős vagy egy tényleg igazán egyszerű, primitív feladat elvégzéséről van szó, az iménti állítás csak részben helyes, viszont jelen helyzetben nem is ezen van a hangsúly. Ha minden egység tökéletesen végzi a dolgát, úgy egy jól működő rendszerről beszélhetünk.

Az előforduló feladatok megoldására szánt eszközök felépítése nagymértékben hasonló. Általában egy (legalább egy) vezetőre és több alárendelt egységre oszlanak fel a feladatok. Nyilván itt nem érintek rendkívül bonyolult rendszereket, viszont ezek előfordulása nem mindennapos, továbbá tapasztalataim alapján a fejlesztők rossz rálátása illetve a feladatok nem megfelelő megközelítése sokkal bonyolultabbá teszi feladatot, mint az valójában. A vezetők feladata leírva viszonylag egyszerű: a részegységek számára a megfelelő instrukciók megadása illetve az általuk végezett feladatok folyamatos megfigyelése. A részegységek munkája értelemszerűen a rájuk bízott feladatok megvalósítása, a megfelelő feltételek között.

Ha erre a szimpla példára úgy gondolunk, mint például egy próbanyákra és egy érdeklődő fiatalra, van egy realizálható példánk a folyamat személtetésére. Az érdeklődő személyében a folyamat lelkét, a vezérlő egységet látjuk, a nyákra felhelyezett elemekben pedig a részegységeket, melyek pontosan azt a feladatot látják el, amire tervezték őket. Komplexebb, szoftveres megközelítésből nézve a dolgot, ugyanez az élethelyzet realizálható például egy egyszerű while ciklussal, ami a megfelelő feltételek mellett folyamatosan fut. Definiálhatunk akármennyi függvényt az egyes feladatok elvégzésére, nekünk csak a megfelelő paramétereket kell biztosítani számukra és azok maguktól végzik a dolgukat. A helyzet akkor sem bonyolultabb amennyiben a feladatok egymástól függenek, átadják a paramétereiket, esetleg a végterméküket, vagy egyazon változót vizsgálják. Kizárólag a vezérlő egységnek (jelen esetünkben az általunk megírt például main függvénynek) kell gondoskodnia arról, hogy az egyes feladatok mindenképp megkapják a szükséges információkat.

A PLC-k, illetve az általuk vezérelt modulok esetére is vázolható az alábbi sablon. Habár egy probléma megoldására rendkívül sok féle módszer elképzelhető, különösen, ha szoftverről beszélünk, bizonyos konvenciók lefektetésével nemcsak leszűkítjük a lehetőségek számát, de a programozók dolgát is egyszerűbbé tehetjük azzal, hogy amennyiben egy már meglévő, de számukra újnak számító projektbe kell becsatlakozniuk, nem kell eltérő megoldásokat átvenniük, így a helyi szabvány szerint megírt kódok megismerése lényegesen kevesebb időbe fog telni nekik. Az általam fejlesztett megoldás annyiban könnyíti meg a dolgot, hogy nem csak a projekten belüli, de a projektek közötti modulok átfedésére is megoldást nyújt. Amennyiben külön tároljuk a vezérléshez szükséges feltételeket és paramétereket illetve a vezérelt modulokat, máris van egy tökéletes sablonunk.

Amennyiben újszerű feladattal vagy eszközzel szembesülünk, nyilván először megoldást kell találni rá, hogy aztán sablont készíthessünk belőle, viszont ez az élet minden újra előforduló problémájára igaz, ezért úgy gondolom, hogy ez sem feltétlen számít a programom Achilles-sarkának.

# A technológia alapjai

## 2.1 Az XML

Az Extensible Markup Language (röviden: XML, magyarul: Bővíthető Leírónyelv/Jelölőnyelv) egy általános, könnyen olvasható leírást biztosít alkalmazás specifikus információk számára. A W3C (World Wide Web Consortium) által megalkotott technológia az SGML (Standard Generalized Markup Language, magyarul: Szabványos Általánosított Jelölőnyelv) egy egyszerűsített megoldása, mely segítségével különböző adattípusokat írhatunk le. Egyéb SGML alkalmazás példák még például a HTML és a DTD.

Az XML egyik elsődleges célja a könnyen olvashatóság mellett a jól strukturált információ továbbítása, mely a programok számára is egyszerűen kezelhető. Egyik legnagyobb alkalmazási területe például az interneten keresztül történő adattovábbítás. Többféle adatstruktúra reprezentálására is tökéletes, emellett előre definiált illetve magunk által megalkotott sémák segítségével validációra is képes, ezáltal biztosak lehetünk, hogy a megadott adatok helyesek, illetve a rendszer által feldolgozhatóak. Mivel egy lightweight, egyszerű szöveges formátumról beszélünk, így nagy mennyiségű adat tárolására is alkalmas.

Igaz, amennyiben terjedelmes fájlról van szó, nem a legegyszerűbb átlátni, azonban mivel, mint azt később részletezem, a modulok leírása és paraméterei pár sort igényelnek csupán, így ezzel a problémával nem kell szembesülnünk. Habár lényegesen egyszerűbb egy fejlesztőkörnyezetben dolgozni vele, írása megoldható egy egyszerű szövegszerkesztő segítségével. Validálásra is van lehetőség, ám ehhez egy kicsit bonyolultabb szövegszerkesztőhöz kell nyúlnunk, mint például Windows alatt a Notepad++ vagy Linuxon a Kate. Ez is mutatja, hogy mennyire egyszerű és gyors a vele végzett munka.