Intelligens Termékkézikönyv
Intelligent Product Manual (IPM)

TDK dolgozat
1998

Készítette:
Lezsák Gergely, Rab Gergely

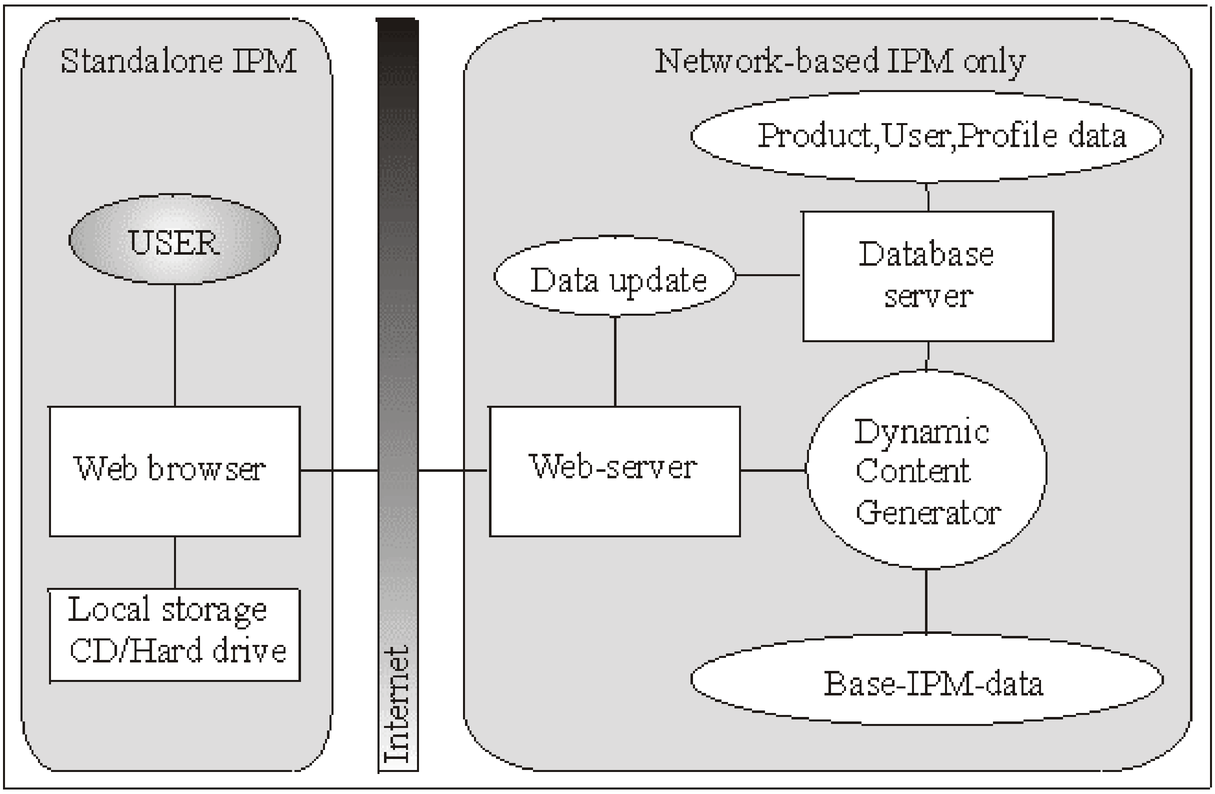
Konzulens: Mészáros Tamás

Tartalomjegyzék

Bevezetõ

Az „Intelligens termékkézikönyvek” (Intelligent Product Manuals -IPM) az Európai Unió egy projektje. Azzal a céllal hozták létre, hogy elkészítsék egy intelligens döntéstámogató rendszer terveit, továbbá egy mûködõképes prototípust. A következõ intézmények mûködnek együtt a projektben:
A brit Walesi Egyetem (Cardiff), a Clausthali Mûszaki Egyetem (Németország), a Rousse-i Egyetem (Bulgária), és a Budapesti Mûszaki Egyetem. A projektet két ipari partner támogatja: az Excel-Csepel Szerszámgépgyár Kft, és a bolgár Sparky PLC.
A Budapesti Mûszaki Egyetem az Excel-Csepel Szerszámgépgyár Kft. számára készít egy IPM prototípust, melyben az Excel által biztosított adatokat használja fel. Az itt bemutatott IPM az EXCEL SL320/600 típusú esztergagép adatait tartalmazza. A jelenlegi prototípuson 4 hónappal ezelõtt kezdtünk dolgozni, az addig összegyûlt elektronikus adatok felhasználásával.
Az Intelligens Termékkézikönyv egy - a hagyományos funkciókat ellátó kézikönyv - korszerû számítógépes platformon implementálva.
Kihasználva a technológia adta lehetõségeket azonban sokkal többet nyújt papír alapú elõdeinél, sõt még a napjainkban forgalomban lévõ elektronikus dokumentációkon is messze túl mutat.
Teljes mértékben megvalósítja a hagyományos kézikönyv funkcióit, így az egyszerû tallózást, a tartalomjegyzék alapján böngészést, szó/téma - jegyzék alapú keresést.
A hipermédia adta lehetõségekkel ez a lista korlátlanul bõvíthetõ. A jelenlegi általunk megvalósított IPM World Wide Web (WWW) alapú dokumentum, kihasználva annak lehetõségeit és rugalmasságát. A HTML-alapú (ld. felépítés) rendszer tökéletesen alkalmas a WWW-n történõ publikálásra, ami biztosítja, hogy az Internet által bárki ingyenesen hozzáférjen az IPM adataihoz. (Természetesen a standalone IPM, egy teljes CD kapacitását felhasználva sokkal nagyobb adathalmazt tartalmazhat - pl. komplett filmeket is). A felhasználói profilok biztosítják, hogy mindenki csak a számára jogosan elérhetõ anyagokhoz férjen hozzá.

A Web alapú lapok a színes képeken túl minden látványos multimédia elemet tartalmaznak, így animációkat, videókat, hangokat, 3 dimenziós modelleket, stb. Ezek a hiperdokumentumok (HTML) sokkal jobban navigálhatók a papír könyveknél, vagy akár az elterjedt, de igen korlátos lehetõségekkel bíró szintén elektronikus (PDF) leírásoknál. Ebben a Web alapú rendszerben egy-egy témát hiperlinkekkel lehet végigkövetni, míg a minket érdeklõ témakört igen jól paraméterezhetõ keresõprogrammal lehet megkerestetni. Megvalósíthatók testre szabott és jogosultságokhoz kötött felhasználói profilok is, melyek elengedhetetlenek az adatok biztonsága érdekében.



Az IPM szerkezeti felépítése

Elvárások:

A rendszert megvásárló cégek és felhasználók jogosan támasztanak elvárásokat az IPM-mel szemben. Igyekeztünk megfelelni az ilyen követelményeknek, ezért a fejlesztés alatt a következõ szempontokat tartottuk szem elõtt:

- Az IPM-nek alkalmasnak kell lennie arra, hogy a felhasználó könnyen kezelje (ld. navigáció), eligazodjon a lapok között, egyszerûen megtalálja a számára érdekes adatokat (ezt biztosítja az ésszerû adatstruktúra, linkek, keresõ), elsajátítsa az új rendszer lehetõségeit (help).
- Látványos multimédia eszközök segítségével meg kell 'gyõznie' a felhasználót a könnyû és áttekinthetõ használatról (ld. Multimédia c. fejezet).
- Tartalmilag teljesen fednie kell a már meglévõ dokumentációk anyagát, hogy ne kelljen 'visszanyúlni' a régi eszközökhöz.
- Ki kell használnia az új platform nyújtotta többletet, hogy valódi elõnyhöz jusson a felhasználó. Ilyenek pl.: interaktív (pl. kérdés-válasz rendszerû) problémamegoldás (ld. Troubleshooter opció), ill. egyéb tudás alapú rendszerek, dinamikusan szerkesztõdõ oldalak (kliens/szerver viszonylatban), aktuális adatokkal kitöltve, rugalmas keresési lehetõségek (search gomb)...
- Biztosítania kell bizonyos felhasználói profilokat, hogy a rendszert kezelõk egyfelõl csak a számukra **jogosan** látható anyagokhoz férjenek hozzá, másfelõl saját ízlésükhöz igazítsák a nap-mint-nap használt eszközt. A profilokról késõbb még bõvebben szólunk.
- Jól hordozhatónak kell lennie, azaz a lehetõ legtöbb hardverplatformon és operációs rendszeren használhatónak kell lennie. Integrálhatónak kell lennie a felhasználók meglévõ eszközeihez (Akár egy vezérlõ funkciót ellátó célszámítógépen is futhasson). Ennek érdekében a legtöbb, amit tehetünk, hogy törekszünk az alacsony hardverigényre és az elterjedt szabványokra (ld. szabványok) építjük fel a rendszert.
- Célszerû támogatnia több nyelvet, mivel elektronikus formában ez nem jelent sok többletkapacitás-igényt, azonban rendkívül nagy segítség a felhasználó számára, ha saját anyanyelvén jut a keresett információhoz.
- Integrálhatónak kell lennie a meglévõ adatforrások formátumaihoz, informatikai rendszereihez, a rugalmas bõvíthetõség érdekében (Ehhez az elterjedt szabványok támogatása szükséges mindkét oldalról).
- Szükség szerint tudjon mûködni önmagában (hálózat nélküli környezetben - Standalone IPM), vagy hálózati környezetben (Intranet/Internet based IPM).
- Hálózati környezetben használja ki az aktuális, friss adatok elérhetõségét a gyártónál. Frissítse a saját (offline) adatbázisát, cseréljen karbantartási/hibakezelési információkat, ha ez szükséges. Maga az IPM is

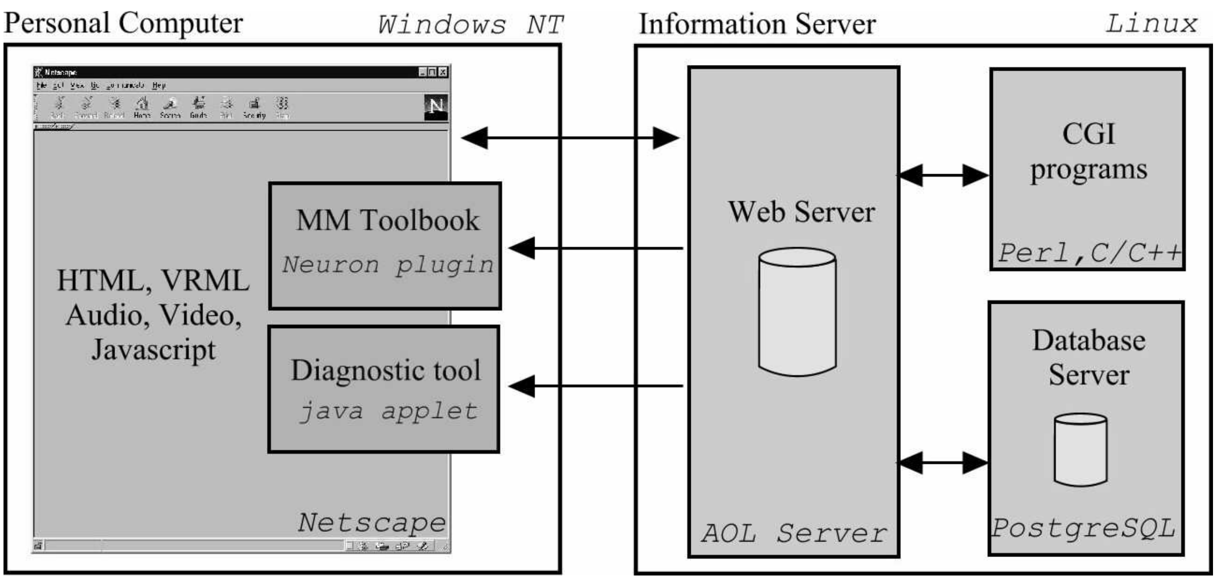
frissítheti önmagát (Automatikus update), ahogy napjainkban a legtöbb új szoftver támogatja ezt a lehetõséget.

- Legyen áttekinthetõ, alkalmazzon a cég / termék image-éhez illeszkedõ designt.
- Legyen elérhetõ mindenki számára. Az Internet –mint média, tökéletesen megfelel ennek az elvárásnak.
- Legyen olcsó! Az elkészült IPM további fejlesztési költségeket nem ró az alkalmazó cégre, csupán a különbözõ termékekrõl szóló tartalmat - az aktuális adatokat kell felvinni (amit természetesen a papír alapú rendszereknél is szükséges). A rendszer alkalmazásakor a telepítés egyszeri költség (az alkalmazott szerver-számítógép ára, vagy a már meglévõ helyi szerver kibõvítésének költsége). A folyamatos gyártási költségek (pl. CD gyártás, Internet elérés) pedig csupán töredékét teszik ki a hagyományos papír alapúaknak (könyvnyomás, postaköltség).

Logikai felépítés

A felhasználó egy hétköznapi papír kézikönyv funkcióit megvalósító felülettel kerül kapcsolatba, mely egy hagyományos könyv átlagos paramétereivel rendelkezik.
Egy könyvnek van tartalomjegyzéke, tartalma és tárgymutatója. A mi interpretációnkban ez tartalomjegyzék, többszintû, bõvíthetõ tartalom és keresõrendszer formájában jelenik meg. Általában lapozni is szoktunk egy könyvben, ezt két féleképpen tehetjük meg: vagy az alfejezet utalásairól, vagy pedig a könyv navigációs gombjai segítségével, amit késõbb fogunk ismertetni. Mielõtt neki kezdtünk volna a megvalósításnak, elgondolkodtunk azon, hogy mi hiányzik egy mai kézikönyvbõl, miért nem lehet egy átlagos kézikönyvet igazán jól kihasználni. Arra a válaszra jutottunk, hogy túl sok idõ alatt lehet megtalálni azokat az információkat, amelyekre szükségünk van, vagy azért, mert nem áll rendelkezésünkre hatékony keresõfeület (pl. arra vagyunk kíváncsiak egy jól dokumentált számítógép használati útmutatójában, hogy hova kell csatlakoztatni a CD-meghajtó kábelét. Erre a hagyományos módszerekkel a kézikönyv méretétõl és a tárgymutató bõségétõl függõen elég hosszú idõ alatt jöhetünk rá, viszont az IPM-be épített keresõrendszernek adott megfelelõ boolean feltételekkel pillanatok alatt megkapjuk a helyes referenciát), vagy azért, mert nem tudunk vagy nem akarunk minden, a lábjegyzetben leírt hivatkozáshoz fordulni, ami a webes változatnál csak pillanatok kérdése a már említett navigációs gomboknak köszönhetõen.

Az elvárások között szerepelt a dinamizmus. Olyan logikai adatstruktúrát kellett tehát kitalálni, amely képes modellezni egy kézikönyv felépítését, az elõbb leírtaknak megfelelõ felülettel tud a felhasználóhoz kapcsolódni, valamint nyitott az adatábrázolási struktúra fejlesztésre, kész a kézikönyv tartalmának bõvítésére, és (majdnem) minden olyan lehetõséget magában hordoz, amelyre szükségünk lehet az IPM tervezett továbbfejlesztésénél. A dokumentumkezelés alapja ezért egy fa szerkezetbe ágyazott objektumhierarchia lett, a fa elágazásain a képernyõn megjelenõ oldalak információival, az adott csomóponttal kapcsolatban álló többi csomópont (alatta, felette, mellette) elérhetõségének adataival. Minden olyan eljárás, amely a fákra vonatkozik, ennél a fánál is felhasználható (keresés, beszúrás, törlés, stb.), viszont azzal a könnyebbséggel, hogy kötött a fa struktúrája (logikailag). Magyarázatképpen képzeljük el a következõ esetet: a fa harmadik szintjére - jelen esetben a tartalomjegyzék fõfejezetei közé - be kell szúrnunk még egy elemet - fejezetet. Ez nem okozhat problémát, mivel a fát tömbben tároljuk, valahova be kell szúrnunk egy tömbelemet, ami triviális feladat, majd a következõ szintre még egyet (mivel egy fejezetnek van legalább egy oldal tartalma), és ezek után átszámozni az arrébbcsúsztatott tömbelemek mutatóit. Kiemelni szintén ilyen könnyû, mivel nem csak csomópontot törlünk, hanem egész részfát, ezért csak a rá vonatkozó hivatkozásokat kell kitörölni, magát a tömbelemet, majd a hierarchiában alatta levõ csomópontokat, és közben folyamatosan átszámozni a tömböt. Már ebben a verzióban lehetõség van általunk virtuális oldalnak nevezett lapok elkészítésére. Ezek valójában nem létezõ oldalak, a rendszer generálja õket. Ilyen például a legtöbb statisztikai kimutatás. Ezek arra használhatóak például, hogy - bár a profil még nincs implementálva - profilfüggõ tartalomjegyzékeket, vagy adatbázisból vett adatok alapján statisztikákat állítsunk össze.



A kliens-szerver architektúra logikai felépítése

Szabványok

Ahhoz, hogy el tudjuk végezni a napi munkánkat, elképzelhetetlen, hogy ne legyen segítségünkre egy olyan rendszer, amely megszabja az adatok egymáshoz való viszonyát, elrendezését. Ez a rendezõ erõ a szabvány. A szabványok megkönnyítik, és egyben nehezítik is munkánkat; könnyítik, mivel nem nekünk kell kitalálni a legjobb konstrukciót, nehezítik, ugyanis olyan kimenetet kell produkálnunk, amely beleilleszkedik e közös megegyezésen alapuló rendszerbe. Könnyûvé válik azonban a dolgunk, ha tudjuk, hogy mit akarunk megvalósítani, például a platformfüggetlenséget, adatnak a futtató alkalmazástól bizonyos mértékig függetlenségét, Internetközeliséget, hálózati topológia függetlenséget, és így tovább. Megkerestük azt a szabványt, amely az összes már leírt és még megvalósításra váró kérdést képes kielégíteni. Ez a World Wide Web.

A Web és a szabványok

A legtöbb ember egyetért abban, hogy a Webet olyan szabványokkal kell leírni, melyek biztosítják, hogy a HTML oldalak bármilyen böngészõtõl, operációs rendszertõl és architektúrától függetlenül ugyanúgy jelenhessenek meg. Ezzel a platformfüggetlen megoldással pénzt lehet megtakarítani, mivel a szerzõnek nem kell a dokumentum különbözõ verzióival törõdni, illetve a felhasználónak nem kell olyan alkalmazást vásárolnia, amelyen kizárólagosan tekinthetõ meg az információ. Azért próbálja minden új HTML verzió az ipar résztvevõinek összhangját tükrözni, hogy a tartalomszolgáltatók befektetései ne vesszenek kárba és dokumentumaik se váljanak olvashatatlanná. A HTML-t úgy fejlesztették ki, hogy sokféle eszközzel elérhetõ legyen a Weben tárolt információ: PC-k különféle felbontásra és színmélységre képes monitorokkal és videokártyákkal, mobiltelefonok, beszédátviteli rendszerek, kis és nagy sávszélességû hálózatokon mûködõ számítógépek egyaránt képesek legyenek elérni a kívánt adatot.

A HTML és a szabványok

A különbözõ szabványok egymáshoz illesztéséhez az ideális választás a HTML-nyelv használata volt, mivel ez a formátum majd' minden eszközön alkalmazható, és sok más elterjedt multimédia szabványt képes magába ágyazni. Erre rendkívül jó példa a mi termékkézikönyvünk:

Adott egy felhasználó, aki a számítógépét, csupán munkaeszközként használja, viszont úgy szeretne vele bánni, mintha a munkatársa lenne, tehát értelmes kérdésekre értelmes választ vár. Értelmes kérdésnek tekinthetjük például: "Nem tudom szétszerelni a fõorsót, mivel nem tudom, hogy milyen sorrendben kell meglazítani a csavarokat". Két dolgot tehet a kérdés feltevõje:

- A keresés funkciót igénybe véve utána néz az anyagban, majd elolvassa a leírást, és ha megértette, megpróbálja megvalósítani mindazt.
Vagy:
- Utánanéz az anyagban, megnézi a csavarkioldás sorrendjét a videoklipen, benéz a fõorsó háta mögé a számítógépes virtuális valóság megjelenítõ program segítségével, majd megszereli a fõorsót.
Nyilvánvaló, hogy a második megoldás sokkal hatékonyabb, fõképp, ha ott áll mellette az elsõ megoldás szövege is. Ezért egy olyan felületet kell megvalósítani, amelybe integrálhatóak más, akár külsõ gyártó szabványai is. Egy HTML oldalban lehet AVI, VRML, MPG, MOV, RA, stb. formátumú média, amely színesebbé, használhatóbbá, multimédiás alkalmazássá teszi az egyszerû lapot. Fontos, hogy a böngészõ fel tudja ismerni ezeket a multimédia szabványokat is. (Netscape-nél a plug-in-ek, Explorernél az aktív komponensek teszik ezt lehetõvé).

A HTML

Mi a HTML?

Ahhoz, hogy valaki a közzétehessen információt, mindenki által értelmezhetõ nyelvre van szüksége, mégpedig olyan publikálási anyanyelvre, amit minden számítógép képes megérteni. A WWW információközlési nyelve a HTML.

A HTML a következõ szolgáltatásokat nyújtja:
- dokumentumok publikálása, minden hozzátartozó részegységgel együtt, pl. szöveg, táblázat, lista, fotó, stb.
- gombnyomásra linkeken keresztül on-line információk lehívása
- formok segítségével távoli szolgáltatásokkal kommunikáció valósítható meg, pl. keresés, megrendelések, vásárlás
- külsõ gyártók termékeinek kimenete (video, hang, különleges adatformátumú dokumentumok) beágyazható

A HTML rövid története

A HTML-t eredetileg a CERN-nél dolgozó Tim Berners-Lee fejlesztette ki, és a Mosaic nevû böngészõvel népszerûsítették. A Web robbanásszerû növekedésével az 1990-es években virágzott, és ez még most is tart. Ez idõ alatt a HTML-t több szempontból is kiterjesztették. A Web azokon a szerzõkön és cégeken múlik, akik mind elfogadják a HTML konvencióit. Ez motiválta az együttes munkát a HTML specifikáció megalkotását.

A HTML 2.0-at 1994 végére fejlesztették ki az IETF (Internet Engineering Task Force) égisze alatt. A HTML+ és a HTML 3.0 sokkal gazdagabb utasításkészlettel dolgozott már, mégsem született a szabványosítási folyamatban egyetértés a különbözõ felek között, tehát nem lehetett látni, merre fog továbbfejlõdni a nyelv. A World Wide Web Consortium HTML munkacsoportja erõfeszítéseinek köszönhetõen 1996-ban lefektették a HTML 3.2 szabványt.

Miért HTML?

Hogy miért éppen a HTML-t választottuk, arra már az elõzõ fejezetben részben választ kaptunk: a multimédiás szabványok illeszthetõsége miatt. Másik indok az volt, hogy a már meglevõ anyagoknak egy olyan könnyen kezelhetõ keretszerkezetet kellett biztosítani, amely Internetközeliségével és dinamizmusával képes megvalósítani mindazon elvárásokat, amelyeket a "Logikai felépítés" címû fejezetben meghatároztunk.
- A Web-es felület könnyen kezelhetõ. Bizonyára mindenki látott már WWW oldalakat. Nem tartalmaznak bonyolult menüsorokat, minden témakör egy-

másfél oldalban elfér, ezekrõl egy egérkattintással eljuthatunk a következõ témakörig, elõre, hátra (esetünkben még több irányban) navigálhatunk, kereshetünk, és ennyi. Az egész megtanulása, megértése nem nehéz, rövid idõ alatt elsajátíthatóak a megfelelõ fogások arra nézve, hogyan jussunk el a keresett információhoz.

- A Webes elérés Internetközeli. Az IPM egyik legnagyobb elõnye, hogy megalkotása után dinamikusan változtatható a tartalma, sõt, maga az egész termékkézikönyv is változtatható. Ennek megvalósításához a legegyszerûbb módszer a webes adatbázis-elérés, illetve szerver oldali alkalmazások futtatása.

JavaScript

Szkriptnyelvek

A HTML segítségével valóban szép oldalakat tudunk készíteni, azonban bizonyos feladatok megoldására önmagában nem alkalmas. Az ilyen specifikus problémákhoz célszerû (a megfelelõ feladatra optimalizált) szkriptnyelvet alkalmazni. Elég sok szkriptnyelv létezik, mindegyik más területen nyújtja a maximumot. Az összes szkriptnyelv megegyezik abban, hogy HTML oldalba építhetõ egy speciális "címke" segítségével és jól ki tudja használni az SGML szabvány adta fa struktúra lehetõségeit. Többé-kevésbé platformfüggetlen és viszonylag browserfüggetlen. Elterjedtségüket, böngészõfüggõségüket, platformfüggõségüket, eszközkészletüket és általános alkalmazhatóságukat figyelembe véve a JavaScript nyelvet választottuk.

A JavaScript nyelv

A nevébõl kiindulva a Javahoz közelítenénk a nyelvet, pedig nem sok rokonság van a kettõ között, legfeljebb a szintaktikájukban, amelynek azonban közös õse a C nyelv. A JavaScript jelenleg megvalósított verziója, az 1.2 már egészen sok funkcióval van ellátva. Próbáltak minden olyan szolgáltatást, amely a dokumentumkezeléssel kapcsolatos ebbe a verzióba sûríteni, vagy legalább is megalapozni. Ilyen például a tömbkezelés (az 1.1 verzióig csak objektumok voltak, amiket különbözõ feladatokra lehetett specializálni, C++ szerû kezeléssel), a mintaillesztés reguláris kifejezésekre, a stringkezelés, és még lehetne sorolni. Minden létezõ tulajdonságot egy fa struktúra bejarásával lehet elérni. Így például ha az ablakban lévõ "Fokeret" nevû keret elérési útjából a paraméterként kapott, kérdõjel utáni részre vagyunk kíváncsiak, akkor arra a top.frames["Fokeret"].location.search kifejezéssel hivatkozhatunk. Az íráshoz szkript részbõl úgy tekintünk a HTML oldalra, mintha egy bizonyos (bármilyen) állapotú papírlapra írnánk, az erre szolgáló write() függvénnyel. A JavaScript hátránya, hogy néhány megoldás túlságosan fejletlen a nyelv átlagosan magas színvonalú eszközeihez képest. Ennek hatására túlságosan elegáns megoldások nem születhetnek. Rovására írható még, hogy nincs jól mûködõ, intelligens debuggere, valamint még az 1.2-es verzió is tartalmaz kisebb, de zavaró hibákat. Mindemellett sok funkciója aluldokumentált.

Biztonsági szempontból a JavaScript nyelv tökéletesen megfelel a mai kor elvárásainak. Megalkotói nagy energiát fektettek a többszintû (lokális gép védelme, felhasználói adatok védelme, rendszerintegritás védelme…) biztonság felállítására. Az adatátvitelt digitális aláírás segítségével lehet védeni, hitelesíteni. Átmeneti biztonsági hézagok csupán az egyes böngészõprogram-implementációk hibáiból fordulhatnak elõ.

Netscape

Az alkalmazott böngészõprogram…

A platformfüggetlenség biztosítása érdekében bizonyos kompromisszumot kellett kötnünk. Jelenleg a különbözõ cégek által gyártott böngészõprogramok (browserek) nem képesek tökéletesen megvalósítani a Java ill. JavaScript specifikációkat. Mivel a jelenlegi IPM prototípus teljes mértékben ezekre épít, a garantált hibátlan mûködést biztosítandó kiválasztottunk egy konkrét böngészõprogramot, a Netscape Navigatort (ill. Communicatort) - továbbiakban Netscape - amely az alábbi szempontok alapján a legjobbnak bizonyult.

1. Sokat kellett szkript nyelven kódolni, és van, ahol Netscape specifikus rész is bekerült a kódba, pusztán azért, mert máshogyan nem volt megvalósítható a kívánt eredmény. Kezdeti idõkbõl származó tapasztalataink szerint a Netscape sokkal megbízhatóbban tudja futtatni a szkripteket, emellett pedig biztosítja a JavaScript specifikáció szerinti támogatottságát, ellentétben a másik nagyon elterjedt (Microsoft Internet Explorer) böngészõvel.
2. A fejlesztések alatt mindvégig Netscape-et használtunk, mivel ennek létezik többféle operációs rendszerre fejlesztett (Unixos/Linuxos, Windowsos, VMS-es, Macintosh-os) verziója is, a többi böngészõgyártó nem büszkélkedhet ekkora termékpalettával. A platformfüggetlenséget a Netscape böngészõprogram választásával megoldottnak tekintjük. Természetesen amint lehetõség nyílik a szoftverpiacon hasonló elterjedtségû, ám Standard Java/JavaScript implementációt alkalmazó böngészõprogram használatára, igyekszünk ahhoz igazítani a jelenleg Netscape-specifikus kódrészeket.
3. Nem volt szükségünk a böngészõ kezelõfelületeire (menüsor, toolbar, kedvencek jegyzeke, stb.), Netscape alatt ezek egyszerûen ki lehet kapcsolni.

Mindazonáltal ha az alkalmazás továbbfejlesztésre érdemesnek bizonyul, valamint a gyártók képesek lesznek megbízhatóan mûködõ operációs rendszereket és böngészõket gyártani, értelemszerûen függetlenítjük a programot a gyártófüggõ részektõl. Nem soroltuk a szempontok közé, de a Netscape forráskódjának nyilvánossága okán elõre látszott, hogy jóval több Netscape dokumentációt találunk a fejlesztéshez, mint más, nyilvánosan nem publikált forráskódú program esetén. A tapasztalat csak részben igazolt: csak elvétve lehet találni néhány szakkönyvben rövid utalásokat igen speciális dolgokra. Viszont az átnézett rengeteg könyvbõl és dokumentációból egyet sem találtunk, amely a Netscape-en kívül más böngészõt ill. böngészõspecifikus részletet tárgyalt volna…

Multimédia

Az IPM egyik fõ eredménye, hogy a számítógépes platform jelenlegi, és jövõbeni lehetõségeit maximálisan ki tudja használni (ui. a rendszer alapját képezõ Internetes technológiák -HTML, Java, Script-nyelvek... - folyamatosan fejlõdnek, mindig kihasználják a kor számítástechnikai eszközei által nyújtotta maximumot).

Természetesen az IPM-ben a tartalom részeként alkalmaztunk színes képeket, amik szemléletessé teszik az anyagot. Technikailag csak elterjedt (a böngészõprogram által is támogatott) képformátumokat használtunk, így a valós színû (16.7 millió szín) képekhez JPG, esetleg PNG formátumot, a 256 színûekhez, a gombokhoz, az átlátszó részekhez és az egyszerûbb animációkhoz a Compuserve-féle GIF képformátumot.

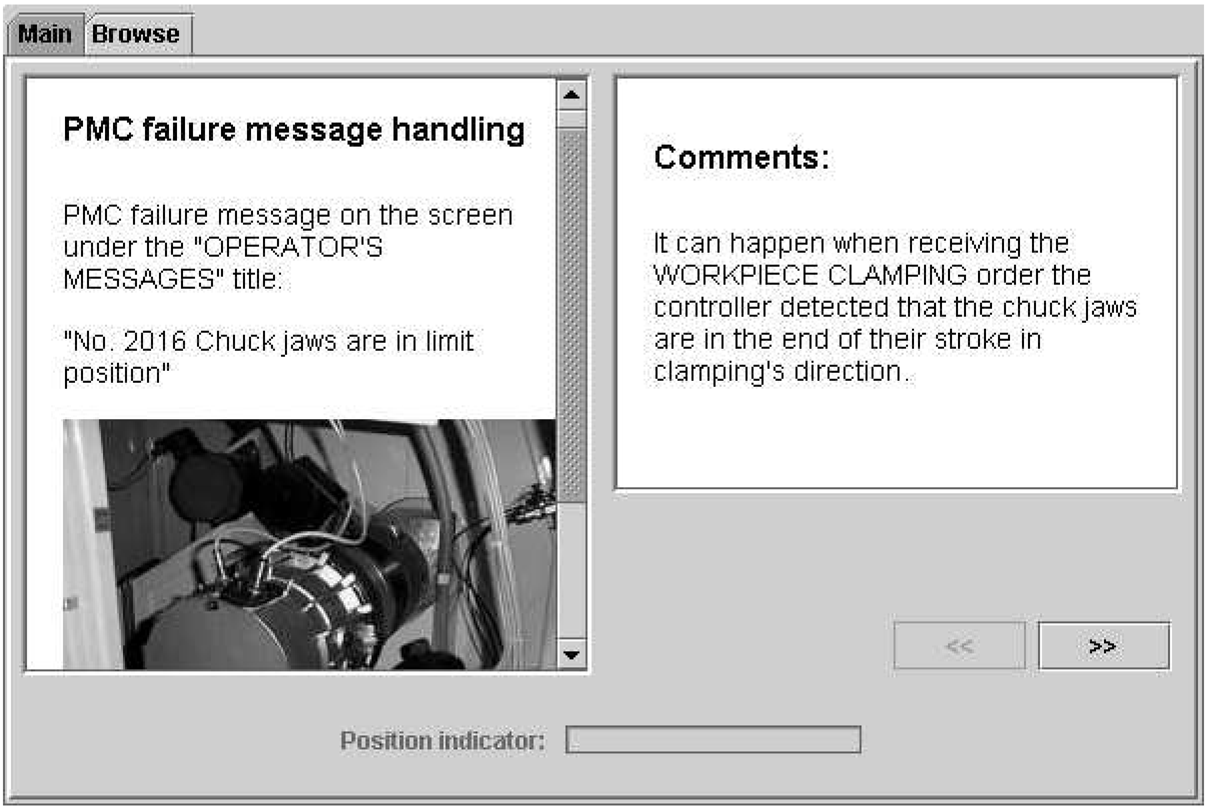
Az oldal arányait és képét minimálisan 800x600-as képernyõfelbontáshoz terveztük. Ez nem különösebben jelentõs elvárás bármely PC-vel szemben. Természetesen jobb megjelenítõvel, nagyobb felbontás mellett egyszerre több részletet látunk a böngészõ ablakában.

Napjainkban egy átlagosnak mondható számítógép számára már nem okoz gondot egy digitális videoklip (vagy animáció) megjelenítése, vagy a CD minõségû hang lejátszása. Ezért az IPM ilyen elemekkel is rendelkezik. A Web-böngészõ (browser - Netscape) ún. plug-in -ek (~ kibõvítések) felhasználásával teszi lehetõvé a különféle médiaklipek (audio és video) lejátszását. A jelenlegi IPM-ben a Microsoft féle AVI formátumban tároljuk a videókat, hanganyagot pedig nem tartalmaz még a prototípus.

Az eszterga bemutatása

Természetesen a megfelelõ browser-plug-in alkalmazásával használhatunk más elterjedt formátumot, mint pl. a Macintosh - környezetben használt MOV, vagy a digitális televíziózásban és videózásban alkalmazott MPEG formátumokat.
Az aktívan kihasznált kétirányú kapcsolat a számítógép és kezelõje között (interaktivitás) további lehetõségeket is nyújt. Például a felhasználó 3 dimenzióban "körüljárhat" egy-egy eszközt ill. annak 3D-s képét (virtuális valóság - virtual reality). Ezt a lehetõséget valósítja meg a VRML vagy WRL formátum, mely szintén egy browser-plug-in által használható. Jelenlegi prototípusunk több ilyen 3D-s modellt is tartalmaz. A jövõben esetleg megvalósítható, hogy a megfelelõ kiegészítõ eszköz alkalmazásával (3D szemüveg, sisak) a virtuális modell valódi 3 dimenziós képet és élményt nyújtson a kezelõnek.
Jelenlegi IPM prototípusunknak egy korábbi változata egy kevésbé rugalmas, ám rendkívül látványos fejlesztõi környezetben készült (Asymetrix Multimedia Toolbook). Ez a verzió is részét képezheti a jelenlegi HTML alapúnak, szintén egy Browser-plug-in (Neuron) segítségével. Szükség szerint az IPM egyes elkülönülõ részei (pl. egy betanító program - tutorial), amelyek számára a látványos kezelõfelület és egyszerû használat fontosabb a hordozhatóságnál*, továbbra is készülhetnek ebben a (TBK) formátumban.
*A Multimédia Toolbook (TBK) állományok csak Microsoft Windows környezetben mûködnek, ill. korlátozott mértékben egy külsõ browser-plug-in használatával. (Ami szintén csak Windows alá létezik)

Problémamegoldás (Troubleshooting) és keresés



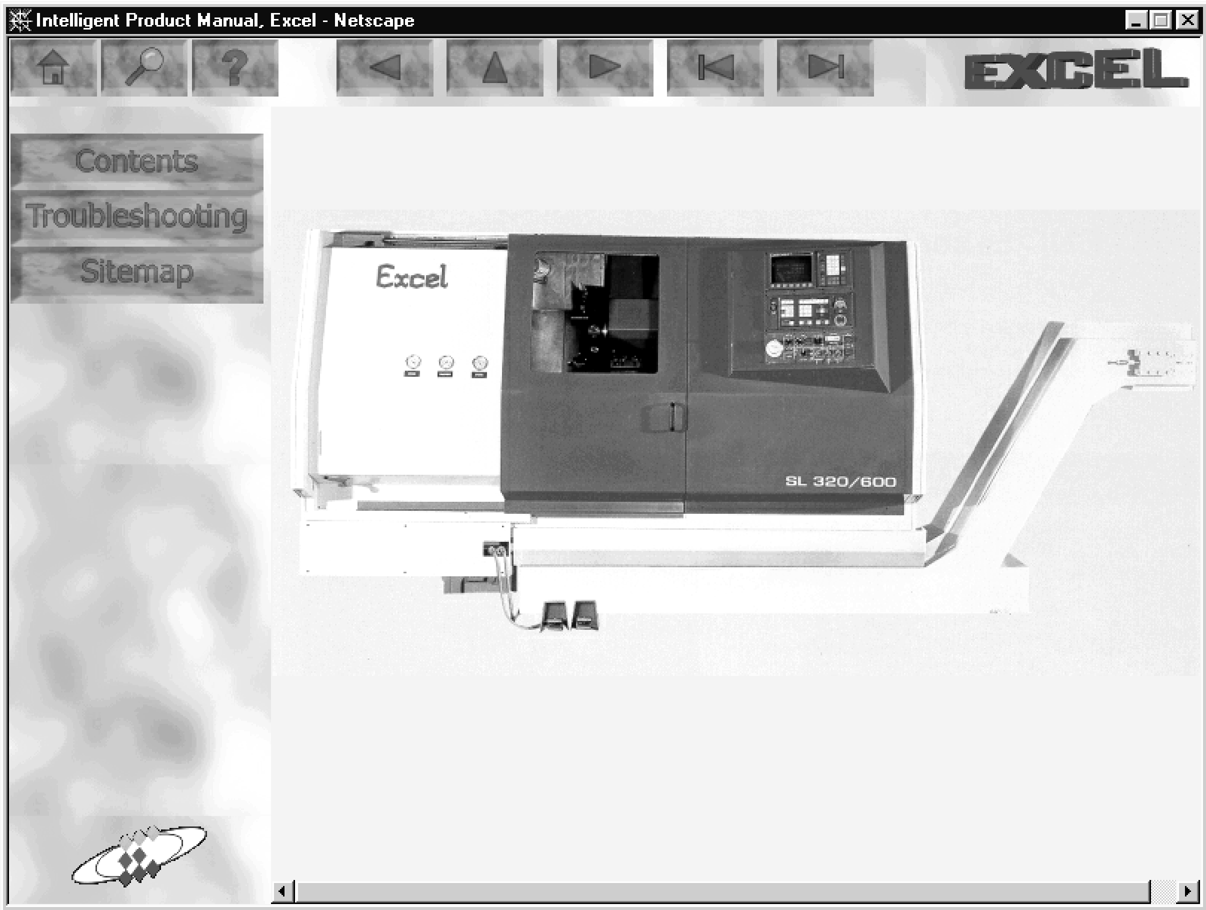
A Troubleshooting ablak

A problémamegoldó (troubleshooter) rendszer az IPM egy szeparáltabb részét képezi. Ez valójában egy normális (számítógép) program, mely a hordozhatóság jegyében JAVA programnyelven íródott (a Java - úgy tûnik- a jövõ technológiája, ezeket a programokat egyre több operációs rendszer, Web-browser, támogatja és néhány számítógép képes native - futtatni). A rendszert Somogyi Csongor fejlesztette, szintén az IPM projekt keretében. A Mi feladatunk a rész rendszerbe történõ integrálása volt.
A troubleshooter egy tudás-alapú rendszer (knowledge based system). Egy elõre meghatározott adathalmaz (fa) elemein 'végiglépkedve' viszi el a használóját a problémától a megoldásig, lépésrõl lépésre. A következõ lépés mindig a felhasználó - aktuális kérdésre adott- válaszától függ. A Troubleshooting rész egy egyszerû szakértõi rendszer, mely a rugalmasan felépített adatbázisa (hibafa) révén szintén egyszerûen bõvíthetõ. Ez a rendszer - a Java nyelv kapcsán- is közel áll a HTML nyújtotta lehetõségekhez, így egy-egy lépés (lap) lehet komplett HTML oldal, képekkel, linkekkel bõvítve.

A keresõ (Search) menüpont ad lehetõséget arra, hogy a felhasználó egy kulcsszó alapján találja meg az õt érdeklõ dokumentumot. A beépített

keresõrendszer hasonlatos bármelyik HTML-keresõhöz, ezért kezelése elég egyszerû:
A bemeneti szövegdobozba írt szót, vagy szavakat keresi. Esetleg megadhatók operátorok is a szavak között (OR, AND), mellyel a keresés eredményét szûkíteni / konkretizálni lehet.

A megvalósított termékkézikönyv



Kezdõképernyõ - a kezelõfelület

Navigáció az adatok között, vezérlõgombok.

A gépkönyv használhatóságának egyik legfontosabb szempontja, hogy kezelõje a lehetõ legegyszerûbben és leggyorsabban jusson hozzá a keresett információkhoz.
Ennek biztosítására az adatok tárolására egy olyan logikai struktúrát alkalmaztunk (ld. az errõl szóló fejezetet), mely az adatok elérését egyszerûvé teszi. Maguk az adatállományok is ezt a struktúrát követik, mind nevükben, mind felosztásukban. - Bár ez az oldal a felhasználó által mindvégig láthatatlan marad, a késõbbi rugalmas homogén fejleszthetõség alapvetõ feltétele.
A felhasználó az IPM elindításakor egy keretrendszerben felépített Weboldallal találja szemben magát. Bízhatunk benne, hogy nem akkor lát elõször ilyen struktúrájú adatokat a számítógépen (korábbi Internetes tapasztalatok), ezért magától is könnyen eligazodik a kezelõfelületen. Ha mégsem, rendelkezésére áll

egy egyszerû segítség, (Help) mely gyors áttekintést ad a gombok jelentésérõl, használatáról.
A bal oldali gombok a fõmenüben a következõ tartalmat takarják:

Contents - tartalomjegyzék
- Az IPM tartalma, a fõ- és alfejezetek címei, sorszáma

Troubleshooting – Hibakeresés
- A gomb megnyomása egy Java alapú, teljesen interaktív step-by-step (lépésrõl lépésre haladó) rendszerû program indítását végzi, mellyel egy adott (elõre betáplált) problémát oldhat meg a felhasználó. Errõl a tudásalapú szakértõi rendszerrõl egy másik TDK dolgozat szól részletesen.

Sitemap - Site-térkép
- Az IPM által fedett adathalmaz strukturális felépítését bemutató sematikus ábra

A bal oldali menü azonban környezet-érzékeny (context-sensitive), azaz tartalma annak tükrében változik, hogy a használó épp melyik menüpont tartalmát nézi. A menü ilyenkor az aktuális almenü további pontjait mutatja, biztosítva ezzel a szorosan kapcsolódó menüpontok áttekinthetõségét, gyors megtalálását.

A felsõ menüsor viszont mindig állandó. Ezek a gombok váltják ki a Web-böngészõ (browser) eredeti navigációs gombjait. Az eredeti gombok lecserélésére több ok miatt is szükség volt. Egyfelõl a kialakításuk, designjuk nem illeszkedett a lapok és a cég image-éhez, ami a megrendelõ cégek szemszögébõl igen fontos szempont, másfelõl funkcionalitásukban sem elégítik ki a jó navigációval szemben támasztott igényeket. Mivel a böngészõ gombjait nem lehet átrajzolni, mûködésüket átdefiniálni, kikapcsoltuk õket, és helyettük saját tervezésû (JavaScript alapú) gombokat tettünk a felsõ sorba. Így kinézetre, és strukturálisan is tökéletesen illeszkedik az IPM belsõ navigációs rendszeréhez.

A felsõ sor gombjai sorra a következõ funkciókat látják el:

Home - Visszavisz a kezdõoldalra, a fõmenübe

Search (nagyító ábra) - A keresõrendszert indítja el, mellyel (kulcsszavak megadásával) lehet az oldalakban ill. az oldalakat keresni.

Help (kérdõjel ábra) - Egy az ittenihez hasonló- rövid áttekintést ad a gombok mûködésérõl és a lapok strukturális elhelyezkedésérõl.

Back (balra nyíl ábra) - Az elõzõ megnyitott oldalra visz vissza (több mélységben).

Forward (jobbra nyíl ábra) - Az éppen aktuális szinten (pl. egy fõfejezetben) lép elõre egy oldalt (egy alfejezetet).

Up (felfele nyíl ábra) - Egy szintet lép felfele az adatstruktúrában (Pl. egy alfejezetbõl egy fõfejezetbe, vagy egy fõfejezetbõl a fõmenübe/tartalomjegyzékbe).

Previous chapter (balra nyíl vonalkával ábra) - Az aktuális strukturális szintrõl (pl. egy alfejezet) az elõzõ fõfejezetre lép.

Next chapter (jobbra nyíl vonalkával ábra) - Az aktuális strukturális szintrõl (pl. egy alfejezet) a következõ fõfejezetre lép.

EXCEL A bal alsó sarokban lévõ ábra (Excel-logo) egy hiperhivatkozás a cég Internetes honlapjára.

A gombok funkcióiról egy kis kirepülõ (flyout) felirat is tájékoztat, ha az egérkurzort a gomb fölé húzzuk és megtartjuk.

Fejlesztési lehetõségek

Lehetõségek

A közeli jövõben a jelenlegi IPM prototípus fejlesztése a hálózat irányába fog eltolódni. A kliens/szerver alapú, Internet bázisú IPM nyit lehetõséget a különféle felhasználói profilok megvalósítására, az adatok hálózati (online) frissítésére, vagy a szerver-oldali adatbázis integrációra.

Egy szerver oldali adatbázisrendszer lehetõvé teszi, hogy a kliensek használati információkat juttassanak el a gyártóhoz (esetünkben pl. az eszterga használódását, meghibásodásait jelentse, alkatrészeket, kiegészítõket rendeljen a gyártótól) és a gyártók viszont.

Egy megfelelõ adatkonverter által akár automatikusan is generálhatók az IPM egyes részei.

A problémamegoldó (Troubleshooting) applet a jövõben valószínûleg már nem csak egyszerû ’hibafákon’ lesz képes végigvezetni a felhasználót, hanem komplexebb feladatok megoldására alkalmas szakértõi rendszerré növi ki magát.

Irodalomjegyzék

- Nagy Péter: JavaScript
- W3C: HTML 4.0 Specification
- Nyékyné G. Judit: Java 1.1 útikalauz programozóknak
- Kris Jamsa: Java
- Kris Jamsa, Suleiman „Sam” Lalani, Steve Weakly: A Web programozása
- Promanual dokumentációk (*proman.manuf.bme.hu*)
- Netscape Communicator dokumentációk (*www.netscape.com*)