

XHELP Programfejlesztési Rendszer

Krafft Walter

Név: XHELP programfejlesztői rendszer HIPO/PASCAL alapon

Kategória: fejlesztő szoftver / software technológiai segédeszköz

Időszak: 1981–1982.

Létrehozó intézmények: MKKE Matematikai és Számítástudományi Intézet, SZKI

Fejlesztők: Abaffy József, Héjjas Attila, Krafft Walter, Manfred Krzikalla

Az XHELP a Számítástechnikai Koordinációs Intézet megbízásából a Marx Károly Közgazdaságtudományi Egyetem Matematikai és Számítástudományi Intézetében 1981-ben kidolgozott kísérleti programfejlesztési rendszer. Célja a teljes programfejlesztési folyamat automatizálására, illetve a különböző fázisok segédeszközökkel való támogatására szolgáló megoldások keresése volt. A prototípus a PDP-11 számítógépen, az RSX-11M operációs rendszer alatt, a PASCAL-2 fordítóprogramot használva működött.

A többlépcsős tervezési folyamat a HIPO módszer ötletén, azaz az Input-Process-Output egyre finomodó megadásán alapult. A tervezői nyelv PASCAL nomenklatúra szerint dolgozott, a generált program PASCAL nyelven.

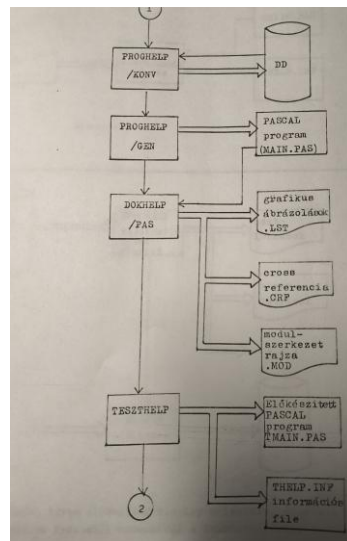
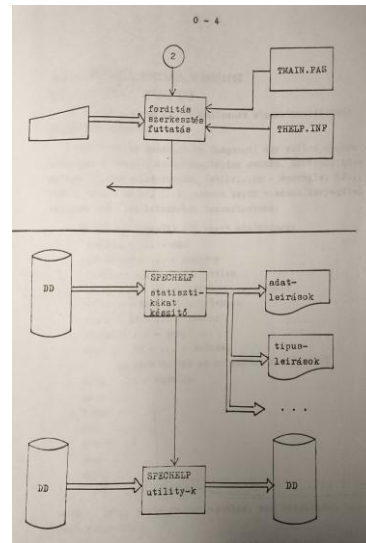
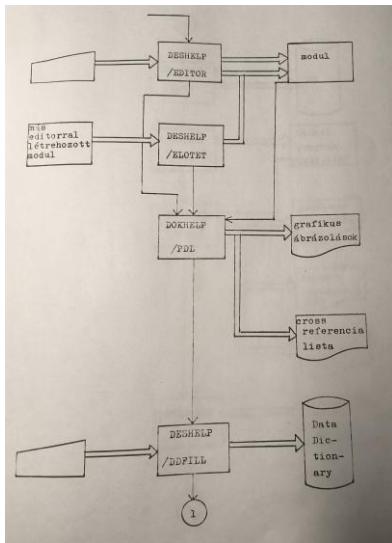
A koncepcióban központi helyet foglalt el egy ún. *Fejlesztői adatszótár* (Data Dictionary), mely a fejlesztés összes lényeges információját tartalmazta. Az XHELP alrendszerei a DD-vel szoros kapcsolatban álltak, adataikat oda töltötték, illetve onnan vették.

Az XHELP-alrendszerei:

- DESHELP: tervezési alrendszer,
- PROGHELP: program generátor alrendszer,
- TESTHELP: teszt processzor,
- SPECHHELP: a *Fejlesztői adatszótárt* kezelő alrendszer,
- DOKHELP: dokumentáló alrendszer.

A DESHELP tovább bontható az EDITOR és a DDFILL (az adatszótárt töltő) funkciókra.

Az alrendszerek összefüggését az alábbi ábrák mutatják (az ábrák az eredeti áttekintő dokumentációból származnak):



Az EDITOR alrendszer: PDL (Program Description Language) formalizmusú algoritmusleírások megadására alkalmas struktúraeditor. A PDL-ben kötött formátumúak a kontroll utasítások (pl. IF...THEN...ELSE...ENDIF, PROCEDURE...ENDPROC stb.), azokon belül viszont lényegében kötetlen leírásokat használhatunk. A nem standard szintaxist követő karaktersorok funkcionálisan kijelölendő modulokként értelmeződnek.

Az adatszótárt töltő alrendszer: kényszerítő párbeszédekben bekéri az egyes modulok input és output információit, s ezeket az adatszótárban tárolja. A DD-ben minden szinten logikailag teljesnek tekinthető információhalmaz található.

A program generátor alrendszer: a program generálásakor a DD-ből kiválasztásra kerülnek a szükséges modulok, adatdefiníciók, a még nem definiált modulok helyére üres rutinok generálódnak, a PDL szintaxis átváltódik PASCAL szintaxisra, a még nem kielégítően tisztázott adatdefiníciók default-okkal helyettesítődnek.

A teszt processzor alrendszer: A dummy modulokkal teletűzdelt program futása általában semmitmondó vagy éppen értelmetlen. Ezért megengedett a rutin funkcionális szimulációja, azaz a modul input- és output paramétereinek a futás közbeni megadása, megállási pontokon. A „megállási pontok” a dummy rutinokba automatikusan belekerülnek, ha a futtatás a teszt processzoron keresztül történik, de bárhol elhelyezhetők programban az ASSERT utasítással.

A dokumentációs alrendszer: A DD révén a fejlesztés során bármikor előállítható az aktuális állapot dokumentációja. Ez részben automatikusan, részben segédeszközökkel támogatott felhasználói közreműködés révén történik.

A dokumentáció áll:

- szöveges leírásokból,
- a rendszert globálisan leíró információkból (modulstruktúra, adattáblák, stb.),
- a modulok leírásaiból – általában három szinten (a funkcionális leírás, modulervezet és programszintű leírás formájában).

Az XHELP számos képszerű ábrázolást tud generálni, amelyekkel áttekinthetővé tehetőek bizonyos információk:

- struktúradiagramot (a programok logikai szerkezetének kiemelésére),
- Leighton-diagramot (egyedi formátumú struktúradiagram),
- keretdiagramot (a programok kontrollszerkezetének a mutatójára),
- fadiagramot (pl. modulok hívási fájának ábrázolására).

Programfejlesztés az XHELP rendszer segítségével

- Az XHELP mint PASCAL fejlesztési környezet: Az XHELP-et ebben a felhasználási módban több, egymástól lényegében független eszköz gyűjteményének tekintjük, s ezen eszközöket használjuk céljainknak megfelelően.
- Az XHELP mint integrált programfejlesztési rendszer: Az XHELP lehetőségeinek, a DD-nek teljes kihasználása akkor érhető el, ha a fejlesztést kezdettől fogva az

XHELP rendszer segítségével végezzük. Így elvileg mindig komplett dokumentáció, módszertanilag helyes vágányon tartott, jól szervezett részek állnak rendelkezésünkre. Ugyanakkor tudnunk kell, hogy az XHELP adminisztrációs igénye nagy, ezért a teljes szabadságról le kell mondanunk, s célszerű két, legfeljebb három szinten finomítani (PDL szint, PASCAL szint, esetleg egy áttekintő szint).

Az XHELP rendszer értékelése

Az XHELP megmaradt ötletthalmaznak. Projektekben alkalmazásra nem került, nem is volt rá alkalmas. Hiányzott az adatbázis háttér, a kötetlen finomítás sem volt életképes.

Ugyanakkor számos elképzelése megvalósult, pl. a C_ASE programfejlesztési környezetben, ami viszont éles és sikeres projektekben, pl. az Országos Felvételi Rendszer létrehozásában bizonyított. A C_ASE funkciói között megtalálhatók a DESHELP, a PROGHELP, a TESTHELP és a DOKHELP megfelelői.

Az XHELP elképzelés az 1990-es évektől kezdve számos, nagy projektekben használt módszertanban visszaköszön. Pl. egy, a svájci biztosítók számára 1999 és 2004 között fejlesztett nagy életbiztosítási projektben a tervezéskor UML modellezést használtak, a modellt tároló META adatbázisból szkripteket és adatbázis definíciókat, a szkriptekből JAVA programstruktúrákat generáltak. A dokumentációk és a help szövegek a META-ból származtak. A tesztelési terv a modell alapján készült. A fejlesztésre egy részletes „Programfejlesztési Kódex” adott útmutatásokat.

A módszer a fejlesztők és a management számára több előnyt biztosított. Külön megemlítendő, hogy a minőségbiztosítás során nemcsak a meglehetősen formális, a fejlesztési folyamatra (de nem a termékre!) fókuszáló ISO modellt lehetett használni, hanem a tényleges szoftvertermék minőségének a mérésére, ellenőrzésre és kiértékelésre is lehetőség nyílt. Ez a minőségbiztosítás számára új távlatokat nyitott, s az ennek keretében készült automatikus kiértékelő és tesztelő programok nagy népszerűsége tettek szert a Megbízóknál(!).

2025. augusztus