

Döntéstámogató rendszerek és adaptáció

Döntéstámogató rendszerek

Számtalan döntést kell életünk során meghozni, amelyek minden esetben kihatással vannak a jövőre nézve. Célunk, hogy az adott feltételek mellett optimális döntést tudjunk hozni. Az optimális döntés feltételezi, hogy teljes körű információ birtokában vagyunk, a lehetséges alternatívák között sorrendet tudunk felállítani. Az összes cselekvési változat és a hozzá kapcsolható várható eredmények ismerete mintegy előfeltétel jelenik meg a döntési folyamatokban. Ha ezek közül bármelyik alapfeltétel sérül, vagyis nem ismerjük az összes alternatívát, vagy a lehető összes információval nem rendelkezünk, akkor korlátozott racionális döntést tudunk csak hozni.

A döntésekhez rendelhető bekövetkezési valószínűségek eltérők. Ha a bekövetkezés számszerűsíthető, akkor kockázatról beszélünk, ha nem tudjuk megadni, akkor bizonytalanságról beszélünk. A valószínűségi változó 0 és 1 közé esik. Legtöbb esetben a környezeti körülmények befolyásoló hatása miatt a rendszer számos bizonytalansággal terhelt. Ilyen esetekben parametrikus döntésemletről beszélünk. Viszont ha adott a valószínűségi változó és kockázati érték, akkor könnyedén képezhető a várható hasznosság. A döntési szabályok ekkor a következők lehetnek:

Laplace-kritérium: Nem ismerjük a végbemeneteli valószínűségeket, ezért mindegyiket egyenlőnek tekintjük. Egyszerűsített a számítás, mivel a változatok várható eredménye közvetlenül adja, hogy mit kell választani

Maximax kritérium: A lehetséges legnagyobb eredmény közül a legjobb eredményt adó alternatívát választjuk. Ehhez a döntési szabályhoz eléggé optimistának kell lenni, hiszen a lehető legjobb eredményt se folytonosan hozzuk, de azok közül mégis a legjobbat választjuk.

Maximin kritérium: Az alternatívák közül azt választjuk, ahol a legkisebb eredmény a többi alternatívához képest a legnagyobb. Tehát a legrosszabb időszakban (például szennyvíztisztítási technológiában télen a többi alternatíva közül a legjobban működik a telep).

A legkisebb megbánás elve: Itt nem eredményeket, teljesítést kell nézni, hanem az elmaradt hasznot. Azt választjuk, ahol a legkisebb az elmaradt haszon. Más néven „minimum regret” elvnek is nevezik ezt a döntési szabályt.

Döntéstámogató rendszernek nevezünk minden olyan rendszert, amely a döntések meghozatalában segíthet. Ez alapján ilyen rendszer lehet például egy egyszerű Excelben elkészített táblázat, egy honlap, egy keresőprogram (például Google) vagy akár a LinkedIn-profil is, ha éppen humán erőforrást keresünk egy adott projekthez. Látszik, hogy széles körű eszköztár tartozik a döntéstámogatáshoz, amelyet a használhatóság érdekében tovább kellene pontosítani, hiszen ebben a formában túl általános, és a következő elemeket kellene tartalmaznia:

- interaktivitás,
- számítógép-alapú,
- adatbázisokat, modelleket alkalmaz,
- nem jól strukturált probléma megoldásában segít.

Az interaktivitási feltétel egyértelműnek tűnik; például nagy adathalmazból strukturált módon kinyert és feldolgozott értékek alapján olyan formában jeleníti meg a döntéshozónak a számára

releváns információt, hogy azt könnyen fel tudja dolgozni. Lényeges elemként tekinthető a vizualizáció, vagyis felhasználni azt a tényt, hogy az emberi észlelés elsősorban a látás segítségével történik. Az adathalmazokból elkészített ábrák, grafikonok rávilágíthatnak trendekre, amiből a jövőre vonatkozó becsléseket elkészíthetjük. Mindez számítástechnikai eszközökkel hatékonyan megvalósítható.

A döntéstámogató rendszer gyakran nagy mennyiségű adatot használ fel, esetlegesen modellekkel közelítéseket, becsléseket végez. A matematikai modell feladata az adatelemek közötti összefüggések felállítása, kezelése. A nem jól strukturált probléma definíciójában szerepel, hogy nem ismerjük az összes megoldási alternatívát, az egymáshoz viszonyított preferenciákat.

A döntéstámogató rendszerek bemutatását a vezetői információs rendszerek relációjában érdemes vizsgálni és összehasonlítani az egyes elemeket.

Főleg taktikai döntésekhez alkalmazandók.

- Támogatják az egyes döntéshozatali folyamatok elvégzését, mint a tervezés és adatgyűjtés, választás az alternatívák közül.
- Egyedi, specifikus problémákat oldanak meg.
- Interaktív felülettel rendelkeznek, a formátumuk rugalmas.
- Az információt matematikai modellek, szimulációk hozzák létre.
- Rugalmasan alakítható rendszerek.

Ellenben a vezetői információs rendszerek:

Elsősorban operatív feladatokhoz dolgozták ki őket.

- Strukturált döntéseket készítenek elő.
- Gyakori problémák megoldásához nyújtanak támogatást.
- Előre meghatározott formával rendelkeznek.
- Meglévő adatok konverziójával állítják elő az információt.
- Nehezen módosítható rendszerek.

A döntéshozatali rendszerek felépítésében láthatjuk, hogy a kiindulási pont az adatbázis, majd van egy belső, modellezési réteg és a user számára látható front-end alkalmazás.

1. Adatbázis: A döntéstámogató rendszer szerves része az adatbázis, egyes esetekben azonban külső forrású is lehet. Az adatbázis már szűrt állapotban, a döntés meghozatalához szükséges mértékben előre fel van dolgozva. Nem kizárólag a tartalom, hanem a szerkezet is meghatározó a döntés előkészítésében.
2. Modellező réteg: A modellező réteg feladata, hogy az adatokból információt állítsunk elő. Ehhez különféle függvényeket, algoritmusokat alkalmaz. Itt kell definiálni a szabályokat és az információ-előállítási metódust.
3. Front-end alkalmazás: A döntéshozók ezzel a felülettel találkoznak, amelyen keresztül az eredményeket megtekinthetik. Ezen eredmények szűrtek, csak a releváns információt használja fel. A megjelenés általában sok vizualizációs elemet tartalmaz.

A döntéstámogató rendszerek szoftvertechnológiáját tekintve a következő lehetőségek jöhetnek számításra:

- adattárház-technológia,
- többdimenziós adatbázis-kezelés (OLAP),
- oszlopalapú adatbázis-kezelők,
- adatbányász-technológiák,
- riportkészítő eszközök.

Adattárházak (Data warehouse)

Az adattárház tárgyorientált, integrált, tartós és időfüggő adatgyűjtemény a vezetői döntéstámogatásra. A tárgyorientáltság jelentése, hogy az alkalmazások funkcióit és feladatait helyezi előtérbe. Általában az adatcsoportosítás a felhasználó és annak preferenciái köré szerveződik, ezeket egy helyre és szabványosítva gyűjti. Törekedni kell az adatok változatlanságára. Amennyiben mégis szükség van a forrásadat megváltoztatására, akkor megfelelő időbélyegzők használata szükséges, hiszen a reprodukálhatóság kulcskérdés. Az időt minden esetben az adathoz kell rendelni, hiszen általában az elemzések elmúlt időszakok adatain alapulnak.

OLAP- (On Line Analytical Processing) rendszerek

Szabványosított elemző rendszer, amely a következő tulajdonságokkal rendelkezik:

- többdimenziós nézet,
- transzparencia,
- jogosultságok beállíthatósága,
- lekérdezések kezelése,
- kliens-szerver architektúra,
- dimenzió fogalma,
- többdimenziós modell tárolásánál dinamikus ritkamátrix-kezelés (elsősorban parciális differenciálegyenletek numerikus analízisének költségkímélő tárolási technika),
- konkurens felhasználó támogatása,
- korlátozás nélküli dimenzióműveletek,
- end-user számára intuitív adatkezelés,
- flexibilis riportkezelés,
- korlátlan dimenziószám és aggregációs szint szám, amely a skálázási problémákon segít.

Oszlopalapú adatbázis-kezelők

Az adatbázis-kezelők alkalmazásakor gyakori a dimenziós adatmodell használata, ahol tény- és dimenziós adatokat lehet elkülöníteni. A ténytáblához lehet a dimenziókat kapcsolni. A ténytáblák általában kevés oszlopot, de sok rekordot tartalmaznak, a dimenziótáblák ezzel ellentétesek; kevés sorból, azonban sok oszlopból állnak.

Adatbányász-technológiák

Óriási adathalmazból (például másodpercenkénti szenzoradatokból) feldolgozható információt alkotunk, akár jellemzők átlagolásával, léptékek helyes megválasztásával vagy releváns adatok kiemelésével. Az adatbányászat az a folyamat, amellyel hasznos információ fedezhető fel automatikus módon nagy adattárakban. A bemenő adatokat különböző formátumokban tárolhatjuk (szövegfájlok, táblázatok vagy relációs táblák), amelyek lehetnek központi adattárolón vagy több hely között elosztva.

Az előfeldolgozás során a nyers bemenő adatokon formátumkonverziót hajthatunk végre, a zajtól megtisztíthatjuk, a lényeges adatokat kiemelhetjük. Az adatok előkezelése a legidőigényesebb folyamat, mivel az egységesítés során a sokfajta nyers adatból indulunk ki.

A rendszerben megjelenik a visszacsatolás, amely az adatbányászat eredményeinek a döntéstámogató rendszerekbe való implementálását jelenti. Az utófeldolgozási lépés biztosítja, hogy csak az érvényes és hasznos eredmények épüljenek be a döntéstámogató rendszerbe. Gyakran az utófeldolgozás részeként statisztikai vizsgálatokat és/vagy hipotézisvizsgálatot is el kell végezni annak érdekében, hogy a félrevezető adatbányászati eredményeket eltávolítsuk.

Riportkészítő eszközök

Elterjedt az üzleti intelligencia (BI: Business Intelligence) területén, de más területeken is alkalmazható, javítja az adatelérhetőséget, könnyebb, gyorsabb hozzáférést biztosít releváns információkhoz, többfajta lehetőséggel:

- jelentések, kivonatok, jegyzőkönyvek elkészítése,
- adatvizualizáció,
- tervezés, előrejelzés, modellezés – What-if scenáriók lefuttatása,
- időszorelemzés,
- mutatók, jellemzők kivonatolása (Balanced Scoreboard),
- dashboardok elkészítése.

A döntéstámogató rendszerek másfajta csoportosítása is lehet, amely inkább elméleti jelentőségű, azonban a megfelelő terminológia elsajátításához elengedhetetlen az ismerete.

- adatvezérelt adatkezelés (például OLAP),
- modellvezérelt adatkezelés (például optimalizációs modellek),
- ismeretvezérelt adatkezelés (például szakértő rendszerek),
- dokumentumvezérelt adatkezelés,
- kommunikációvezérelt adatkezelés (user által bevitt adatok alapján).

Az 5.1. táblázatban láthatjuk az egyes döntéstámogató rendszertípusokhoz rendelhető folyamatokat és alkalmazott technológiákat, amely a kliens/szerver kapcsolatot, az önálló (stand-alone) PC-t és a webes felületet öleli fel.

5.1. táblázat

Döntéstámogató modelltipusok folyamatai és alkalmazott technológiái (saját szerkesztés a <http://real.mtak.hu/10316/1/1212879.pdf> alapján)

	Folyamat	Alkalmazott technológia
Adatvezérelt	lekérdezés az adattárházból	kliens/szerver
Modellvezérelt	döntéselemzés	kliens/szerver, PC, web
Ismeretvezérelt	belső információk gyűjtése	kliens/szerver, PC
Dokumentumvezérelt	weboldalak, dokumentumok	web, kliens/szerver
Kommunikációvezérelt	együttműködés, találkozások	web, kliens/szerver

Döntési folyamat

Lineáris programozás

A döntéstámogatás során optimumra törekszünk, azaz a rendelkezésre álló információk alapján a kritériumok figyelembevételével opciók között választunk. A rendszer olyan alapállapottal rendelkezik, amelyben az egyes változatokhoz hasznosság (utility) társul. A hasznosságfüggvény a változatok és a hasznosság együttes súlyozásával áll elő. A függvények közül megkülönböztetünk feltételi függvényeket és célfüggvényt. Általában a célfüggvény minimumának vagy maximumának keresése a feladatunk valamilyen korlátozó feltételek között. Ha a feltételi függvények és a célfüggvény lineáris, akkor a probléma lineáris programozást igényel. Ha bármely függvényünk nem lineáris, akkor a probléma is nem lineáris természetű. Ha a feltételi függvények és a célfüggvény felírhatók egyváltozós függvények összegeként, akkor dinamikus programozást alkalmazunk.

A programozási feladat ezenkívül lehet folytonos, diszkrét vagy hibrid a függvénytípusoktól függően. Ha valószínűségi változók is szerepelnek a feladatban, akkor sztochasztikus programozási feladatról van szó, viszont valószínűségi változók hiányában a probléma determinisztikus. Általánosan felírva a célfüggvény maximalizációjához az alábbi rendszert kell megoldani:

$$\begin{aligned} A \cdot x &\leq b \\ x &\geq 0 \end{aligned}$$

Ahol az A mátrix a technológia mátrixa, elemei a technológiai együtthatók. A mátrix n sorból és m oszlopból áll:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix}$$

A b vektor a kapacitásvektor, a c vektor pedig a célfüggvény, az x pedig az ismeretlen.

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}, \quad c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

a célfüggvényre felírható:

$$c^T x \rightarrow \max$$

A megoldás során alkalmazhatunk grafikus módszert, amelynek előnye, hogy kisszámú korlátozó feltételnél és kevés változónál kézzel is hamar eredményre jutunk. Azonban ha a rendszerünk sokváltozós, és az automatizálás is fontos, akkor érdemes a simplex módszert alkalmazni. A feladat megoldása gyakorlatilag a lineáris egyenletrendszerek megoldásából következik.

Szimplex módszer

A szimplex módszer kezdőtáblája olyan táblázat, amelynek a bal felső sarkában található a technológiai együttható mátrix, jobb felső sarkában a kapacitásvektor, az alsó sorban pedig a célfüggvény együtthatói szerepelnek.

5.2. táblázat

Szimplex módszer kezdőtáblája (saját szerkesztés)

	x_1	x_2	...	x_m	b
u_1	a_{11}	a_{12}	...	a_{1m}	b_1
u_2	a_{21}	a_{22}	...	a_{2m}	b_2
...	...	...	...	...	...
u_n	a_{n1}	a_{n2}	...	a_{nm}	b_n
$-c$	c_1	c_2	...	c_m	0

A szimplex módszer lépései a következők:

1. A generálóelem kiválasztása: A generálóelem kiválasztása csak abból az oszlopból tehető meg, ahol a célfüggvény együtthatója nem negatív, tehát $c_1, c_2, \dots, c_n$ közül csak a zérus és nagyobb értékek jöhetnek szóba. Elvi lehetőség van minden nem negatív elem kiválasztására, de célszerű a $\max(c_1, c_2, \dots, c_m)$ érték megkeresése és kiválasztása. A generálóelem oszlopa legyen j ! A generálóelem sorának kiválasztásához azt a sort kell tekinteni, amelyben a b_i/a_{ij} a legkisebb. Ez az ún. legkisebb keresztmetszet.
2. A generálóelem segítségével elemi bázistranszformációval új bázisra áttérés: A bázisvektorcserét úgy hajtjuk végre, hogy a generálóelem helyére írjuk a reciprokát, majd a generálóelem oszlopát szorozzuk a generálóelem reciprokának -1 -szeresével és a generálóelem sorát szorozzuk a generálóelem reciprokával.
3. Az eljárás iteratív módon való ismétlése.

Az eljárás addig folytatandó, amíg a táblázat jobb alsó cellájában tovább nem csökken az érték. Ekkor értük el a maximumot.

A következő egyenletrendszer megoldásához használjuk a szimplex módszert:

$$x_1 + 2x_2 + x_3 \leq 3$$

$$3x_2 + x_3 \leq 5$$

$$2x_1 + 3x_2 + 3x_3 \leq 7$$

ahol minden változó pozitív,

keressük a $c(x) = 3x_1 + x_2 + 4x_3$ függvény maximumát.

Első lépésben a kiinduló táblázatot írjuk fel:

	x_1	x_2	x_3	b
u_1	1	2	1	3
u_2	0	3	1	5
u_3	2	3	3	7
$-c$	3	1	4	0

Ezután keressük meg a generálóelemet! Az alsó sorban pozitív számok szerepelnek. Ekkor célszerű olyan elemet választani, amelynek oszlopában zérus szerepel. Ha a b-t soronként leosztjuk az adott oszlophoz tartozó értékekkel, és a legkisebbet kiválasztjuk (7/2 vagy 3/1), akkor a következő generálóelemet kapjuk:

	x_1	x_2	x_3	b
u_1	1	2	1	3
u_2	0	3	1	5
u_3	2	3	3	7
$-c$	3	1	4	0

Ezután a generálóelem sorát a reciprokával szorozzuk, oszlopát a reciprok -1 -szeresével, majd elemi bázistranszformációkat végrehajtva ezt kapjuk:

	u_1	x_2	x_3	b
x_1	1	2	1	3
u_2	0	3	1	5
u_3	-2	-1	1	1
$-c$	-3	-5	1	-9

Iteratív módon folytatva az eljárást az új generálóelem:

	u_1	x_2	x_3	b
x_1	1	2	1	3
u_2	0	3	1	5
u_3	-2	-1	1	1
$-c$	-3	-5	1	-9

Áttérés más bázisra és bázistranszformáció után:

	u_1	x_2	u_3	b
x_1	3	3	-1	2
u_2	2	4	-1	4
x_3	-2	-1	1	1
$-c$	-1	-4	-1	-10

Vagyis az $x_1 = 2$, $x_3 = 1$, ebből viszont következik, hogy $x_2 = 0$. Így a $c(x)$ függvény maximális értéke 10 lesz.

Ha az olvasó szeretne a témában jobban elmélyedni, lineáris programozás keresőszóra számos irodalmat találhat.

A döntéstámogató rendszer felépítése

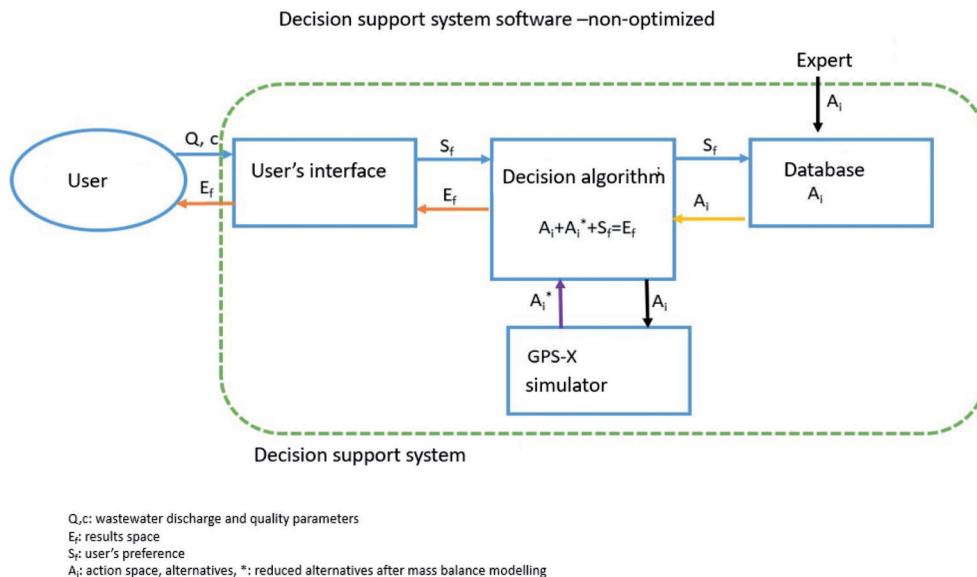
Szennyvíztisztító kisberendezések kiválasztásához a döntéstámogató rendszerek elméleti leírását és a szakmai követelményeket figyelembe véve járunk el. A rendszer létrehozásánál elsődleges szempont az egyszerűség és a könnyen átláthatóság mind a rendszer kezelője, mind a felhasználó részéről.

A felhasználó a döntéstámogató rendszer elemeivel a felhasználói felületre bevitt saját preferenciáján keresztül kommunikál, azaz adott mennyiségű és minőségű szennyvizet szeretne megtisztítani az előírásoknak megfelelően. A felhasználói felület ezt az input adatot a döntésemélet szerinti felhasználói környezeti elemnek felelteti meg, és közvetíti a logikai motorhoz, a döntési algoritmushoz.

A másik oldalról közelítve az adatbázis tartalmazza a lehetséges alternatívák terét (cselekvési tér), vagyis hogy mely technológiai megoldások jöhetnek szóba. A valós biológiai-fizikai folyamatok modellezésével ez a cselekvési tér módosul: szűkebb lesz, és a felhasználó preferenciájához jobban illeszkedő. A szűkített cselekvési tér és a környezeti tér vektorainak segítségével születik meg az eredménytér, illetve a tér elemeiből kiválasztva a felhasználó igényeinek legjobban megfelelő megoldás, amelyet a felhasználói felület segítségével mint eredményt visszajuttatunk a felhasználónak. A fenti leírás szerint kétfajta lehetőség képzelhető el az anyagforgalmi modellezés rendszerbe kapcsolását illetően:

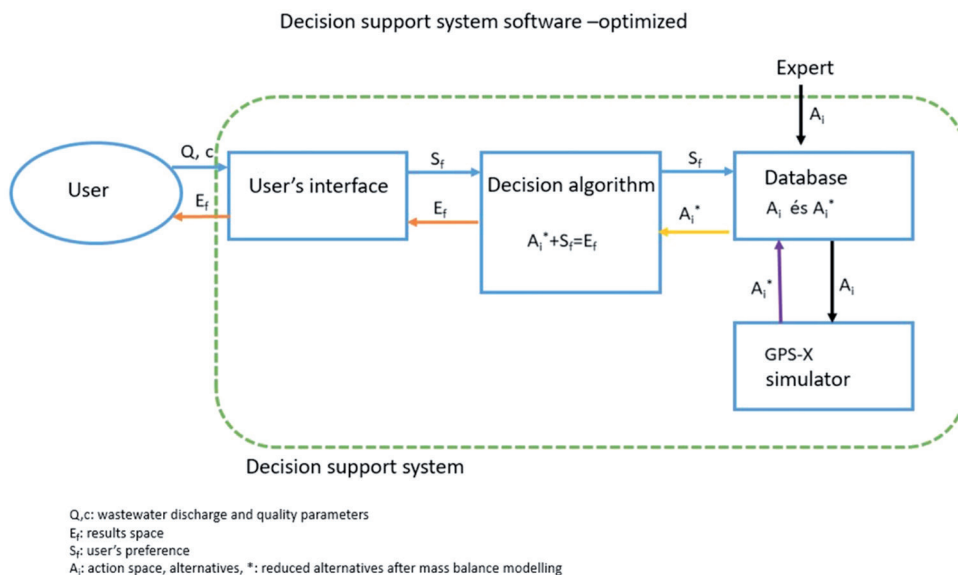
- az adatbázis alternatívateréhez csatlakozik,
- a döntési algoritmushoz csatlakozik.

Az utóbbi esetben a dinamikus, a felhasználó által elindított folyamattal párhuzamosan kell futnia minden lekérdezéskor, ami lassú és költséges folyamat. Ezért megoldásként az optimalizált rendszerben a GPS-X-szel végzett anyagforgalmi modellezés a statikus adatbázist „okosítja” fel, így csökkenthető a szükséges szimulációk száma, és a felhasználó már csak ebből az „okosított” adatbázisból választ.



5.1. ábra

Kisberendezések döntéstámogató algoritmusának vázlata – nem optimalizált (saját szerkesztés)



5.2. ábra

Kisberendezések döntéstámogató algoritmusának vázlata – optimalizált (saját szerkesztés)

Döntéstámogató rendszer alkalmazása

Adatbázis

Amint a döntéstámogató rendszer felépítéséből is látható, a rendszer magját az adatbázis alkotja. Az adatbázis tartalmazza az egyes kisberendezések főbb ismérveit, a névleges kapacitás mellett azonban az adott szennyvízminőséghez rendelt számított kapacitásokat is meg kell határozni. Ennek érdekében az adatbázis az anyagforgalmi modell eredményeit használja fel. Minden egyes kisberendezés-specifikációra a szimulációs környezet az 5. fejezetben leírt módon meghatározta a tényleges kapacitásadatokat. Ehhez a számításhoz az átlagos szennyvízminőséget vettük figyelembe. Ha a szennyvízminőség ettől eltér, akkor az üzemeltetési paramétereken (például levegőmennyiség) keresztül tudjuk a technológiát finomhangolni.

A táblázat azon berendezéseket tartalmazza, amelyeket a mellékletben ismertettünk. Ez azonban nem zárja ki, hogy idővel bővüljön az adatbázis. Sőt a legalapvetőbb kritériuma a döntéstámogató rendszernek, hogy naprakész adatokból induljon ki. Ebből kifolyólag, ha új alternatíva jelenik meg, akkor a következő lépések megtételére és kérdések megválaszolására van szükség:

- megvizsgálni, hogy a lehetséges technológia a döntéstámogató rendszer számára alternatíva-e,
- az alternatíva műszaki specifikációinak összegyűjtése,
- az anyagforgalmon alapuló szimulációs környezetben a layout elkészítése,
- futtatás a névleges teljesítményen az adott szennyvízminőségre nézve,
- iteratív módon megtalálni azt a kapacitást, ahol megfelelő a kezelt szennyvíz minősége,
- az adatbázis bővítése az adott elemmel és szimulációs eredménnyel,
- a döntéstámogató rendszerben az adatbázis frissítése.

5.3. táblázat

A kisberendezések kapacitásadatai (saját szerkesztés)

Kisberendezés típusa		Névleges kapacitás		Számított kapacitás
		LEÉ	m ³ /d	m ³ /d
POLYDOX	Polydox 6	1–8	0,9	0,38
	Polydox 12	8–12	1,8	0,76
	Polydox 30	30	4	1,7
	Polydox 50	50	6	2,5
ÖKO TECH Home	A, B Clear 6	6	0,78	0,42
	A, B Clear 8	8	1	0,53
	A, B Clear 10	10	3	1,6
GRAF	One2Clean 1-3 EW	3	0,45	0,24
	One2Clean 4-5 EW	5	0,75	0,4
AS-VARIO COMP	5K ULTRA	3–5	0,6	0,38
	8K ULTRA	6–10	1,2	0,77
	15K ULTRA	11–17	2,25	1,45
	20K ULTRA	18–24	3	1,9
	30N	25–33	3,75–4,95	2,7
	40N	34–44	5,1–6,6	3,5
	50N	45–55	6,75–8,25	4,6
	60N	56–70	8,4–10,5	5,8
	80N	71–90	10,65–13,5	7,6
	100N	91–110	13,65–16,5	9,2
	125N	111–135	16,65–20,25	11,8
	150N	136–155	20,4–23,25	14,1

Döntési algoritmus

A felhasználó felől érkező információkból a kapacitástábla felhasználásával egy választási algoritmussal kell leképezni az eredményt. A felhasználónak egyszerűen értelmezhető, könnyen megválaszolható és gyorsan kitölthető kérdéseket kell feltenni. Ha szakemberektől kérdeznénk az alapadatokat, akkor bizonyára a vízfogyasztási trendeket, napi ingadozást, a nyers szennyvíz KOI, BOI, TSS, TN, TP és egyéb fizikai paramétereit (például hőmérséklet) és a kívánt tisztítási hatékonyságot kérdezhetnénk meg. Ez esetben a megadott inputokkal a felhasználó felelőssége is nagyobb lenne. Azonban a projekt keretében elkészült döntéstámogató rendszernek letisztultnak kell lennie, és kevés (nem szorosan vett szakmai) kérdésből kell állnia, így a felhasználótól a következő kérdésekre érkező válaszokat dolgozza fel a rendszer:

1. hány lakos szennyvizét kell tisztítani,
2. egy főre eső vízfogyasztás,
3. területhasználat – választási lehetőség több alternatíva közül.

A kérdések között nem szerepel a kívánt tisztítási hatékonyság, mivel azt minden esetben 90%-os szervesanyag-eltávolításra és közel teljes nitrifikációra állítottuk be.

A döntéstámogató rendszer működésének lépéseit a következőkben vázoljuk:

1. *Terhelés meghatározása és a tényleges kapacitással való összevetés – egységek száma:* A biztonság javára tervezünk, ezért feltételezzük, hogy az elhasznált víz 100%-ban szennyvízként jelentkezik. A rendszer hidraulikai terhelése a lakosok száma szorozva az egy főre eső vízfogyasztással. A terhelési adatot kell az egyes egységek tényleges (anyagforgalmi modellen alapuló kapacitásával) összevetni, és a megfelelő rendszert ajánlani. A döntéstámogató rendszer nem egyfajta berendezést javasol, hanem több alternatívát ajánl, és meghatározza a beszerzendő egységek számát.

2. *Levegőtétési igény meghatározása – aerob zóna oldottoxigén-koncentrációja:* Az egy főre eső vízfogyasztásból és a napi fajlagos kibocsátásból indulunk ki (60 g BOI₅/fő.nap, 120 g KOI/fő.nap, 14 g TKN/fő.nap, 1,5 g TP/fő.nap). Ha az egy főre eső vízfogyasztás 100–120 l/fő alatti, akkor a hidraulikai terhelés ugyan kisebb, de a szennyvíz koncentráltabb. Javasolt a DO = 3–3,5 mg/l tartása az aerob zónában. Ha az 1 főre eső szennyvízhozam 120 l/fő-nél nagyobb, akkor a DO = 2–2,5 mg/l is elegendő lehet az adott térfogaton.

3. *Keletkező iszapmennyiség becslése:* A napi szervesanyag-terhelés függvényében az iszaptermelés is meghatározható a kommunális szennyvizek esetében (1 kg KOI-ból 0,85 kg TS lesz) ebből megkapjuk, hogy hány kg TS/d keletkezik. Feltételezve a 3,5–5%-os sűrített iszapot, megtudjuk mondani, ez hány liter naponta (vagy m³ évente). Ezt a számot össze lehet vetni a hidraulikai terhelés 2–4%-ával, közel egy nagyságrendbe kell esnie a két számított értéknek.

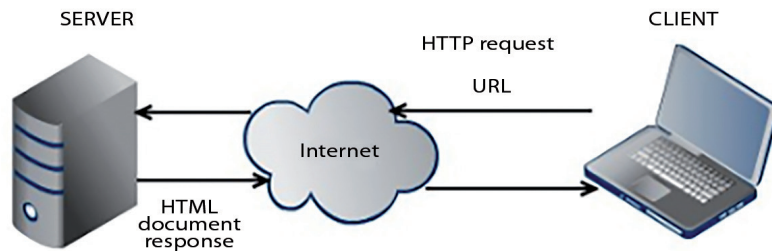
4. *Üzemeltetési javaslatok:* A szakszerű üzemeltetéshez és karbantartáshoz szükséges, hogy minden egyes rendszerhez biztosítsunk a felhasználó számára üzemeltetési kézikönyvet, amely részletezi a teendőket napi, heti, havi, éves bontásban. Ezen információk döntéstámogató rendszerbe való integrálására azért volt szükség, hogy a felelős döntés ne csak a beruházási oldalról, hanem üzemeltetési oldalról is támogatva legyen.

Felhasználói felület

A döntéstámogató rendszerek a felhasználóval egy felhasználói felületen (user interface) keresztül érintkeznek. A felhasználói felület egy egyszerűsített leképezése a rendszernek, a felhasználó számára releváns információkat tartalmazza. Ezenkívül a felhasználó és a döntéstámogató rendszer kommunikációját is biztosítja. A mai kor kihívásainak megfelelően, a könnyű hozzáférés biztosítása érdekében webalapú technológiát érdemes alkalmazni, ezen belül is a legegyszerűbb megoldás egy honlap készítése.

A honlap a webes szerveren található, eléréséhez ún. URL-t alkalmaz. Az URL egy címkézett hely, amellyel a weblap azonosítása történik meg, és lehetővé teszi az egységes hivatkozást. Az egységes szintaxis alkalmazását protokollok segítik. A legismertebb a HTTP és FTP protokoll. Az 5.3. ábrán látható, hogy a kliens (client) az URL segítségével a világhálón keresztül éri el a webszervert. A felkeresés eredménye, hogy a szerver válaszként a kliens rendelkezésére bocsátja a dokumentumot/weblapot.

A HTML hiperszöveg-leíró, jelölő nyelv, amelyet leggyakrabban honlapok készítésére használunk. Gyors, platformfüggetlen, a multimédiás eszközöket támogatja, kiterjesztése.htm vagy a .html. A hiperszöveg jelentése, hogy az egyes fájlok könnyen kapcsolhatók egymáshoz. A szöveges nyelvben nemcsak tartalom, hanem forma is beállítható. Vagyis a szövegleírás határozza meg a megjelenítendő adatok honlapon való elhelyezését és a hivatkozásokat is. A webböngésző feladata ezen utasítások értelmezése és végrehajtása. A HTML az ASCII kódtáblát alkalmazza, vagyis egy szöveges fájl.



5.3. ábra

Kliens-szerver kapcsolat (www.tankonyvtar.hu/hu/tartalom/tamop425/0027_ADW1/ch01s02.html)

A HTML esetében többfajta adattípust különböztethetünk meg:

- szöveg: tetszőleges számú és típusú karaktert enged meg,
- szám: elsősorban méret megadása vagy területjelölés,
- dátum/idő,
- tartalom: a dokumentumba ágyazást teszi lehetővé,
- hivatkozás,
- médiumleíró: például screen: képernyő, projection: kivetítő.

A hiperhivatkozások ma már a weboldal alapvető részei, segítségükkel tudunk tartalmak között váltani, a dokumentumok egyes pontjaira állni. A hivatkozást bárhová elhelyezhetjük az oldalon, hivatkozhatunk a dokumentum más részére, a weboldal más pontjára, de az adott weboldalon kívüli pontra is. Az összekötő kapocs, a link, egy másik fájlra mutat. A linkeknek két kategóriáját különböztethetjük meg; az abszolút és relatív linket.

- Abszolút link: a célfájl teljes útvonalát meg kell adni, nem kell a hivatkozó oldalból kiindulni.
- Relatív link: a hivatkozó oldalból kiindulva adjuk meg az elérési útvonalat.

A döntéstámogató rendszerek alapvető eleme, hogy a felhasználó képes legyen adatot bevinni. Ezt nyomtatványokkal, űrlapokkal (form) tehetjük meg. Az adatok bevitel után lehetőség van arra, hogy a szerveren lévő szoftver azt feldolgozza, vagy még a honlapon más, nem HTML-alapú program használja. A döntéstámogató rendszer kialakításához érdemes megismerni a HTML-űrlapok részeit és kidolgozásának lépéseit. Az űrlapunktól elvárjuk, hogy az adatokat bekérje, tárolja, feldolgozza, majd az eredményt a felhasználó részére e-mailben elküldje. Az adatbevitel esetén, ha hibát vétünk, elvárás, hogy azt a felhasználó számára jelezze.

1. Űrlap kerete. A formok beágyazásához a parancsokat a <form>...<form>közé kell írni. A form attribútuma az action, amely egy hivatkozás a form feldolgozó programjára. Ezt általában egy gomb megnyomásával aktiválhatjuk. Például Enter vagy külön gomb beépítésével. Az eredmény elküldéséhez e-mail-címet kell megadni: <mailto:xy@z.hu>. Meg kell adni milyen módon (method) szeretnénk az adatot elküldeni. Erre alapvetően a GET és a POST áll rendelkezésre. Javasolt a POST használata, hiszen nincs méretkorlátja, és a jelszavakat is kódolva küldi. A kódolásra (enctype) is több változat alkalmazható.

2. Input megadása. Az adat, amit beviszünk, több típusba tartozhat, amelyek a következők:

- text: 20 karakter hosszú mező, de begépelni tetszőleges hosszúságú szöveget lehet,
- password: olyan text, amely a bevitt adatokat a megjelenítéskor *-gal helyettesíti,

- checkbox/radio gomb: kiválasztási lehetőség azzal a különbséggel, hogy míg a checkboxokból több is kijelölhető, a radio gombnál csak egy,
- file: állományok feltöltése a számítógépről,
- submit: elküldi az adatokat,
- reset: a bevitt adatok törlése, alaphelyzetbe visszaállítás.

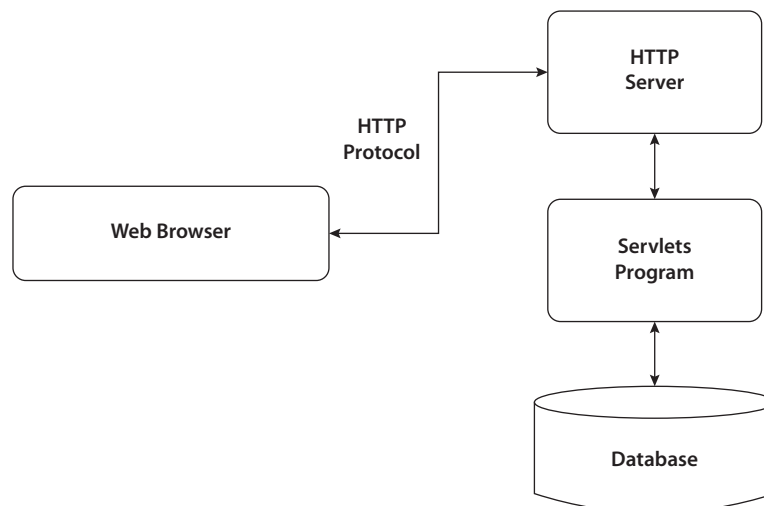
A textnek többfajta elérhető attribútuma van:

- name: vezérlő neve,
- size: mezőszélesség,
- maxlength: beírható maximális szöveghossz,
- value: kezdeti érték,
- readonly: írásvédett mező,
- disabled: nem engedélyezett mező (például a korábbi kérdésekre adott válaszok alapján nem releváns).

Ha sok elem közül kell kiválasztani a megfelelőt, akkor a radio gombok helyett legördülő listát is alkalmazhatunk. Az ilyen listát `<select> ... <select>` közé kell helyezni, majd a listaelemeket az `<option> ... <option>` között soroljuk fel.

Az elküldött adatok lekérdezése

Az elküldött adatok feldolgozásához Java környezetben segíthet egy olyan objektum, amely HTTP-kéréseket dolgoz fel, és HTTP-választ tud létrehozni. Ezáltal dinamikus weblapot készíthetünk. A Java ezt az objektumot servletnek hívja, ebből kifolyólag a servletet Java nyelven kell megírni. A servlet container kezeli a webszerveren található összes servletet, azok életciklusát és URL-hez való hozzárendelését.



5.4. ábra

A servlet működésének alapjai (www.tutorialspoint.com/servlets/servlets_overview.htm)

Megjegyzés: Web Browser: webböngésző, Database: Adatbázis

A servlet életrajza a következőkből tevődik össze:

1. A servlet container létrehozza a servlet objektumot.
2. A container inicializálja a servletet, azaz kezdeti értékkel ruházza fel.
3. A kliens kiszolgálása.
4. A container meghívja a törlés (destroy) metódusát, és a servletet megszünteti.

A fenti lépések a 3. lépés kivételével csak egyszer játszódnak le a servlet életében.

A formokhoz kapcsolódóan háromféle servletet kell megismernünk:

- *HttpServletRequest*: bejövő kérés adatai, információk gyűjtése.
- *HttpServletResponse*: az adatfeldolgozás után a kimenő választ biztosítja a kliens számára.
- *ServletContext*: a servlet környezetét befolyásolja, folyamatosan működik, az inicializációkor kerül „betöltésre”.

A *HttpServletRequest* metódusai:

- *Kérés módjának lekérdezése*:
 - `public String getProtocol()`: visszatérési értéke a kérés által használt protokoll neve és verziószáma;
 - `public String getMethod()`: visszatérési értéke a kérésben adott parancs.
- *Kapcsolat adatainak lekérdezése*:
 - `public String getRemoteHost()`: a számítógép IP-címét adja;
 - `public String getLocalName()`: a kérést kiszolgáló szerver IP-címe;
 - `public int getLocalPort()`: a számítógéphez tartozó kapcsolat helyi portszáma.
- *Form által küldött paraméterek lekérdezése*:
 - `public String getQueryString()`: az URL részeként küldött paraméterstring.
- *Hozzáférés egyetlen paraméterhez*:
 - `public String getParameter(String name)`: visszaadja a megadott nevű paraméter értékét, ha a bevitt adat szöveg típusú;
 - `public String[] getParameterValues(String name)`: visszaadja a megadott nevű paraméterhez tartozó értékeket;
 - `public Enumeration getParameterNames()`: a felsorolás tartalmazza a böngésző által elküldött összes paraméter nevét.
- *Küldött adatokhoz való hozzáférés*:
 - `public String getCharacterEncoding()`: a kliens által a törzsadatok küldésekor használt karakterkódolási módot adja vissza;
 - `public int getContentLength()`: a küldött adat hosszát adja vissza.

A *HttpServletResponse* metódusai:

- *Válasz típusának beállítása*:
 - `public void setContentLength(int len)`: válasz hosszának meghatározása;
 - `public void setCharacterEncoding(String charset)`: a válasz küldése során használt karakterkódolási mód meghatározása.
- *A válasz elküldési lehetőségei*:
 - `public ServletOutputStream getOutputStream()` throws `IOException`: Byte alapú adatok visszaküldésére szolgáló csatorna, például képek esetén;
 - `public PrintWriter getWriter()` throws `IOException`: szöveges adat visszaadása.
- *Átírási kérés küldése*:
 - `public void sendRedirect(String location)` throws `IOException`: adott címre irányítja át a böngészőt.

Meg kell említeni, hogy intelligens weblapok szerkesztésére, a fent vázolt folyamatok és logikák alkalmazásának megkönnyítésére számos felületet hoztak létre. Egészen olyan megoldások is vannak a piacon, ahol a weblapszerkesztő már nem is találkozik a kóddal, csupán egy grafikus felületen keresztül alakíthatja weblapját. Ezek a szerkesztő felületek abból indulnak ki, hogy a weblapoknak vagy azon belüli űrlapoknak vannak állandó elemei (képek, szövegek, fájlok), és ezáltal egy keretet adhat. A szerkeszthetőségnek azonban van korlátja, ha egyedi megoldásokat akarunk választani, arra nem fogunk előre „beprogramozott” megoldást találni. Azonban ha egyszerű, tucatfeladatról van szó (adatbevitel, beolvasás, algoritmusfuttatás, válasz küldése), azokra bátran használhatjuk.

Láthattuk, miként épül fel a döntéstámogató rendszer, hogyan futnak le annak háttér-szimulációi, és a webes alkalmazás hogyan épül fel. Ezen technikák alkalmazása egyszerre igényli a szennyvíztechnológus és informatikus szakértelmét, de ki kell hangsúlyozni, hogy a döntést meghozón van mindig a felelősség. A szakértő-/döntéstámogató rendszer csupán javaslatot tesz, amivel vagy él a felhasználó, vagy nem, de a következményeket mindig a döntést meghozó viseli.

Ajánlott irodalom

- Antal P, Hullám G, Millinghoffer A, Hajós G, Antos A. Valószínűségi döntéstámogató rendszerek. BME–TypoTex; 2014.
- Balogh P, Felföldi J, Herdon M, Kemény G, Nagy L, Nábrádi A, Szöllősi L, Szűcs I. Döntéstámogató módszerek és rendszerek. Elméleti jegyzet. Debrecen: Debreceni Egyetem Gazdálkodástudományok Centruma; 2013.
- Bartels RH, Golub GH. The simplex method of linear programming using LU decomposition. Communications of the ACM. 1969;12(5):266–268. <https://doi.org/10.1145/362946.362974>
- Bharati P, Abhijit Ch. An empirical investigation of decision-making satisfaction in web-based decision support systems. Decision Support Systems. 2004;37(2):187–197. [https://doi.org/10.1016/S0167-9236\(03\)00006-X](https://doi.org/10.1016/S0167-9236(03)00006-X)
- Bhargava H, Power D. Decision support systems and web technologies: a status report. AMCIS 2001 Proceedings; 2001. 46.
- Chaudhuri S, Dayal U. An overview of data warehousing and OLAP technology. ACM Sigmod Record. 1997;26(1),65–74.
- Csordásné Marton M. Matematika példatár 7. Lineáris Algebra II. Nyugat-magyarországi Egyetem; 2010. Letöltve: 2018. 12. 04. www.tankonyvtar.hu/hu/tartalom/tamop425/0027_MAT7/ch01s05.html
- Edelstein H, Barquin RC. Planning and designing the data warehouse. New York, USA: Simon & Schuster; 1996.
- Estrada J, Shore MB. U.S. Patent No. 7,222,291. Washington, DC: U.S. Patent and Trademark Office; 2007.
- Garrido-Baserba M, Molinos-Senante M, Abelleira-Pereira JM, Fdez-Güelfo LA, Poch M, Hernández-Sancho F. Selecting sewage sludge treatment alternatives in modern wastewater treatment plants using environmental decision support systems. Journal of Cleaner Production. 2015;107:410–419. <https://doi.org/10.1016/j.jclepro.2014.11.021>
- Get rights and content
- Giordano R, Passarella G, Uricchio VF, Vurro M. Integrating conflict analysis and consensus reaching in a decision support system for water resource management. Journal of Environmental Management. 2007;84(2):213–228. <https://doi.org/10.1016/j.jenvman.2006.05.006>
- Han J, Pei J, Kamber M. Data mining: concepts and techniques. Elsevier; 2011.
- Herdon M, Szilágyi R. Internetalapú döntéstámogatási rendszerek. Debrecen: Debreceni Egyetem, Agrártudományi Centrum Agrárinformatikai és Alkalmazott Matematikai Tanszék. Letöltve: 2018. 12. 10. <http://real.mtak.hu/10316/1/1212879.pdf>
- Kövári A. Döntéstámogató rendszerek (DSS). 2007. Letöltve: 2018. 12. 09. www.biprojekt.hu/Dontestamogato-rendszer.htm
- Légrádi G, Szénási S. Interaktív weboldalak készítése, XHTML form. Adatok feldolgozása szervezetekkel. Budapest: Óbudai Egyetem; 2012. Letöltve: 2018. 12. 11. http://users.nik.uni-obuda.hu/java/Java_6.pdf
- Matthies M, Giupponi C, Ostendorf B. Environmental decision support systems: Current issues, methods and tools. Environmental Modelling and Software. 2007;22(2):123–127. <https://doi.org/10.1016/j.envsoft.2005.09.005>

- Mordani R, Chan SW. Java™ Servlet Specification. Sun Microsystems Inc., version 3; 2009.
- Mysiak J, Giupponi C, Rosato P. Towards the development of a decision support system for water resource management. *Environmental Modelling and Software*. 2005;20(2):203–214. <https://doi.org/10.1016/j.envsoft.2003.12.019>
- Naár J. Weblap-szerkesztés lépésről lépésre. Debreceni Egyetem, Informatikai Kar; 2008. Letöltve: 2018. 12. 11. <https://dea.lib.unideb.hu/dea/bitstream/handle/2437/70526/Szakdolgozat%20-%20Na%E1j;jsessionid=F38B115866F34D5A3B2D8C502255F0ED?sequence=1>
- Tan PN, Steinbach M, Kumar V. Bevezetés az adatbányászatba. Budapest: Panem; 2011. Letöltve: 2018. 12. 05. www.tankonyvtar.hu/hu/tartalom/tamop425/0046_adatbanyaszat/ch01.html
- Power DJ. Web-based and model-driven decision support systems: concepts and issues. *AMCIS 2000 Proceedings*; 2000. 387.
- Plattner H. A common database approach for OLTP and OLAP using an in-memory column database. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM; 2009, June. 1–2.
- Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R, Carlsson C. Past, present, and future of decision support technology. *Decision support systems*. 2002;33(2):111–126.
- Sidló Cs. Adattárház rendszerek. Diplomamunka. Budapest: ELTE; 2003. 77. Letöltve: 2018. 12. 09. http://scs.web.elte.hu/Work/DW/papers/sidlo_dipl.pdf
- Sprague RH, Watson HJ editors. *Decision support systems: putting theory into practice*. Englewood Cliffs, NJ: Prentice-Hall; 1986.
- Wierzbicki A, Makowski M, Wessels J. editors. *Model-based decision support methodology with environmental applications*. Dordrecht, The Netherlands: Kluwer Academic; 2000. 475.
- Witten IH, Frank E, Hall MA, Pal CJ. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann; 2016.

Fejezetzáró kérdések

1. Milyen döntési szabályokat ismer?
2. Hogyan definiálhatjuk a döntéstámogató rendszereket?
3. Hasonlítsa össze a vezetői irányítási rendszereket a döntéstámogató rendszerekkel!
4. Mely három rétegre osztható egy döntéstámogató rendszer? Mi az egyes rétegek feladata?
5. Milyen tulajdonságai vannak az OLAP-rendszernek?
6. A döntéstámogató rendszerek különböző típusai milyen alkalmazásokon futhatnak?
7. Mutassa be a szimplex módszert!
8. Hogyan lehet az anyagforgalmon alapuló GPS-X szimulációs környezetet beágyazni a döntéstámogató rendszerbe?
9. A döntéstámogató rendszer adatbázisáról mit tud mondani?
10. Min alapszik a döntési algoritmus?
11. Vácsoljon egy kliens-szerver kapcsolatot!
12. A HTML-ben milyen adattípusok lehetnek?
13. Miből áll egy HTML form?
14. Mi a servlet?
15. Milyen HTTP Servlet Request metódusokat ismer?