A rendszer nem azonos a felülettel

Slug: a-rendszer-nem-azonos-a-felulettel | Publikalva: 2026. 06. 30. | Tags: Rendszertervezés, AI-governance, Vállalati AI-kockázat, Vizuális modellezés

Egy alkalmazásban megnyomjuk a "jóváhagyás" gombot. A képernyö egy pillanatra elsötétül, megjelenik egy pipa, és úgy tünik, mintha a döntés ott, abban a négyszögben történt volna. Közben lehet, hogy adatbázisok frissültek, jogosultságok léptek életbe, értesítések indultak el, pén...

Egy gomb mögött ritkán csak egy müvelet történik

Egy alkalmazásban megnyomjuk a "jóváhagyás" gombot. A képernyö egy pillanatra elsötétül, megjelenik egy pipa, és úgy tünik, mintha a döntés ott, abban a négyszögben történt volna. Közben lehet, hogy adatbázisok frissültek, jogosultságok léptek életbe, értesítések indultak el, pénzügyi kötelezettség keletkezett, egy másik ember munkafolyamata megváltozott, és valahol naplóbejegyzés készült arról, ki mit tett.

A felület ezt nem hazudja el feltétlenül. A feladata éppen az, hogy a használathoz szükséges bonyolultságot kezelhetövé tegye. A baj akkor kezdödik, amikor a használati egyszerüséget összekeverjük a rendszer egyszerüségével.

Ez a tévedés nem csak technikai. Ugyanígy nézünk egy állami ügyintézési oldalt, egy vállalati AI-asszisztenst, egy piacteret vagy egy kórházi idöpontfoglalót. Amit látunk, az néhány mezö, státusz és gomb. Amit müködtetünk, az emberekböl, szabályokból, adatokból, szerzödésekböl, jogosultságokból, kivételekböl és felelösségi pontokból álló hálózat.

A felület a rendszer sürített ígérete. Azt mondja: ezt a müveletet itt elvégezheted, ennek ez lesz a látható eredménye. Ebböl azonban még nem következik, hogy a háttérben minden szerep tisztázott, minden hibautat kezelnek, és minden döntés visszakövethetö.



A rendszer nem azonos a felülettel - axonometrikus szolgáltatásmetszet

A prototípus egy útvonalat bizonyít

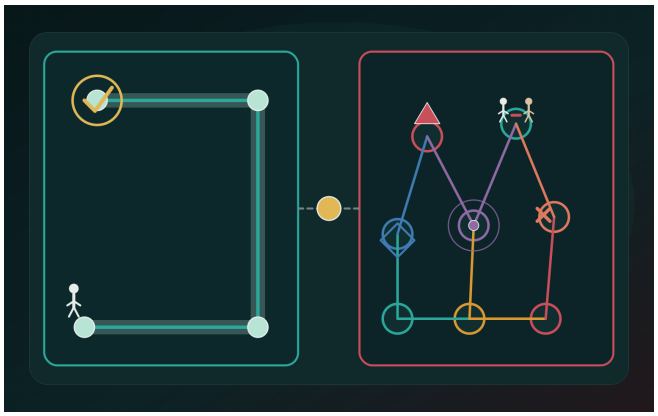
Egy jól müködö bemutató meggyözö. A felhasználó beír valamit, a rendszer válaszol, a grafikon megmozdul, az ajánlás megjelenik. A jelenet kerek. A prototípus valóban bizonyított valamit: legalább egy összeállításban, meghatározott bemenettel, ismert környezetben létrejött a kívánt kimenet.

Ez értékes eredmény. Csak más kérdésre válaszol, mint a tartós müködés.

A mindennapi rendszernek el kell viselnie a hiányos adatot, a párhuzamos használatot, a félbehagyott

Fractal Solutions

folyamatot, az ismételt kattintást, a rossz jogosultságot, a külsö szolgáltatás kimaradását és azt az esetet is, amelyre a bemutató készítöi nem gondoltak. A prototípus többnyire a sikeres útvonalat mutatja. Az intézményesített szolgáltatásnak a sikertelen, vitatott és határeseteket is rendeznie kell.



A rendszer nem azonos a felülettel - kettös folyamatlabor és hibafa

A gépi tanulást használó rendszereknél ehhez új rétegek társulnak. A modell minösége függhet az adatoktól, az elöfeldolgozástól, a környezö kódtól, a szolgáltatási lánctól és a visszacsatolásoktól. Sculley és szerzötársai ezt a láthatatlanul halmozódó függöségrendszert a gépi tanulási rendszerek technikai adósságaként írták le [4]. A látványos modell csak egy kis doboz egy jóval nagyobb rajzon.

A demonstráció tehát fejlesztési bizonyíték. Üzemeltetési, biztonsági vagy intézményi bizonyítékká csak további vizsgálatokkal válhat.

A jogosultság nem menüpont

A kezelöfelületen a jogosultság gyakran egy kapcsoló. Adminisztrátor, szerkesztö, partner, ügyfél. A szerep neve tisztának látszik, pedig négy külön kérdést sürít össze: ki cselekedhet, milyen tárgyban, milyen feltételekkel, és ki viseli a következményt.



A rendszer nem azonos a felülettel - felelösségi térkép és helyzetmátrix

Egy piactér adminisztrátora például törölhet profilt. De törölhet-e folyamatban lévö szerzödés mellett? Láthatja-e a privát üzenetet? Módosíthatja-e a pénzügyi státuszt? Felülírhat-e egy automatikus döntést? Kell-e indokolnia? Marad-e nyoma? Ki vizsgálja a vitát, ha az adminisztrátor maga is érintett?

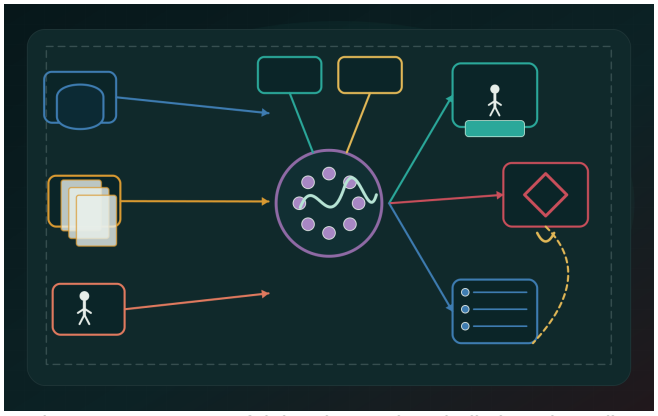
Ezek nem képernyötervezési részletek. Az intézményi felelösség szerkezete jelenik meg bennük. Norman klasszikus tervezési munkái arra mutattak rá, hogy a jó felület érthetövé teszi a lehetséges müveleteket és következményeket [1]. Ez azonban csak a felhasználói oldal. A rendszer másik felén azt is rögzíteni kell, hogy a müveletnek van-e jogalapja, ellenörzési pontja és visszafordítási útja.

Fractal Solutions

A jogosultsági mátrix ezért önmagában kevés. Ugyanaz a szerep másként viselkedik normál müködésben, incidensnél, jogvitában vagy adatvédelmi kérésnél. A rendszer akkor kezd valóságossá válni, amikor a szerepnevek mögé bekerülnek a feltételek, kivételek és felelösségi láncok.

Az AI-válasz még nem döntési rendszer

Egy nyelvi modell képes összefoglalni panaszt, kockázatot keresni egy szerzödésben, vagy javaslatot adni egy ügyfél következö lépésére. Ebböl könnyü úgy beszélni, mintha az "AI" volna a rendszer. Valójában a modell egy komponens, amely bemenetet kap, kimenetet ad, és közben bizonyos hibamintákat hordoz.



A rendszer nem azonos a felülettel - szociotechnikai rendszerdiagram

A döntési rendszerhez ennél több kell. Ki választotta ki a bemenetet? Milyen adat maradt ki? Melyik modellverzió futott? Milyen utasítással? Volt-e emberi ellenörzés? Mit lehet tenni, ha az eredmény téves? A javaslat státusza világos volt a felhasználónak? Megörizhetö-e úgy a napló, hogy az ellenörzés lehetséges, az érzékeny adat pedig ne terjedjen feleslegesen?

A NIST AI-kockázatkezelési kerete ezért a technikai mérés mellett irányítási, feltérképezési és kezelési feladatokat is elkülönít [6]. A hangsúly az egyetlen modell "jóságáról" áttevödik arra, hogyan illeszkedik az adott szervezet, felhasználási cél és kockázat világába.

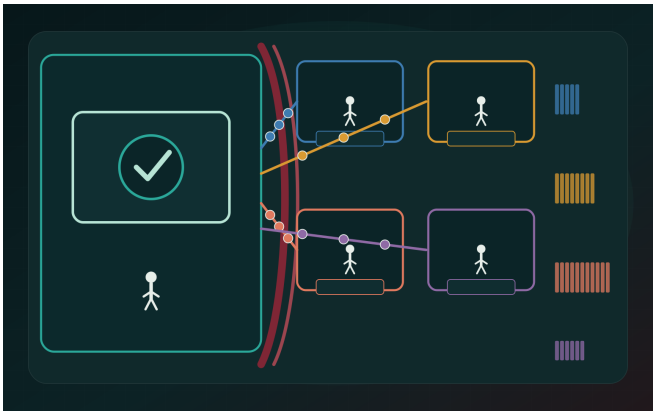
A szociotechnikai rendszerekröl szóló kutatás hasonló problémát jelez. Ha a társadalmi helyzetböl kivágunk egy tiszta, számítható részletet, könnyen elveszíthetjük azt, ami a döntés jelentését adja. Selbst és szerzötársai ezt az absztrakciós csapdák között tárgyalják: egy formálisan rendezett modell úgy is lehet elégtelen, hogy a saját határain belül következetes [5].

Az AI-kimenet ezért lehet hasznos bizonyíték, javaslat vagy ellenörzési nyom. Döntési joggá attól válik, hogy az intézmény ténylegesen ráépíti a hatáskört, a felelösséget és a jogorvoslatot.

A háttérmunka eltünése is rendszertervezési döntés

Sok "automatikus" szolgáltatásban emberek dolgoznak a háttérben. Kivételt kezelnek, adatot javítanak, tartalmat ellenöriznek, manuálisan összekötnek két folyamatot, vagy telefonon oldják fel azt, amit a felület nem tudott.

Fractal Solutions



A rendszer nem azonos a felülettel - dokumentarista backstage-jelenet és kapacitásdiagram

Ez átmenetileg teljesen ésszerű lehet. A veszély abból származik, ha a szervezet maga sem tudja, mennyi müködés maradt rejtett kézi munkában. Ilyenkor a rendszer terhelhetösége, költsége és hibaaránya nem ott van, ahol a vezetöi dashboard mutatja. Egy új ügyfél felvétele látszólag egy automatikus folyamatot bövít, valójában három munkatárs napi félóráját adja hozzá.

Lucy Suchman munkái óta erös érv szól amellett, hogy a technikai cselekvést ne elöre megírt terv végrehajtásaként lássuk. A valós helyzetben emberek értelmezik a körülményt, korrigálnak és rögtönöznek [2]. Ezt a rugalmasságot nem kell eltüntetni. Láthatóvá kell tenni, különben az emberi korrekció költségként, kockázatként és tudásként is kiesik a rendszer önképéböl.

A jó automatizálás minöségét az mutatja, hogy világos, mely ponton miért van szükség emberre, mit láthat, mit módosíthat, és mikor kell visszaadnia a döntést a szabályozott folyamatnak.

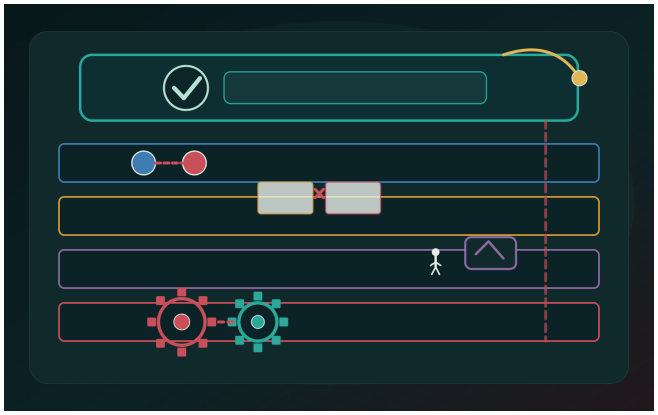
A hiba nem feltétlenül ott keletkezik, ahol megjelenik

A felület piros hibajelzést mutat. A felhasználó azt látja, hogy nem sikerült a müvelet. A háttérben azonban lehet hibás adat, elavult üzleti szabály, rossz rendszerkapcsolat, túl szük jogosultság vagy olyan konfliktus, amelyet a tervezés eleve nem tudott reprezentálni.

Komplex rendszerekben a káros kimenet gyakran több, külön-külön elfogadhatónak tünö döntés találkozásából jön létre. Leveson rendszerbiztonsági megközelítése ezért a komponenshibák mellett a vezérlési és visszacsatolási struktúrát is vizsgálja [3]. Ez a gondolat digitális szolgáltatásnál is használható: a modulhiba azonosítása után azt is fel kell tárni, milyen korlát, jelzés vagy ellenörzés hiányzott a teljes folyamatból.

Egy hibajegy önmagában gyakran túl kicsi. "A gomb nem müködik." A javítás után lehet, hogy a gomb müködik, a felhasználó mégsem jut elöre, mert a jogosultsági állapot és a szerzödéses állapot továbbra is ellentmond egymásnak. A felületi hiba eltünt, a rendszerfeszültség megmaradt.

Fractal Solutions



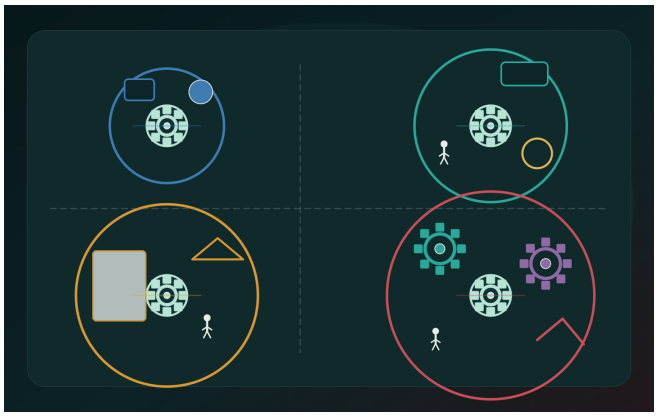
A rendszer nem azonos a felülettel - hibametszet és oksági rétegábra

Ezért kell külön látni a tünetet, a lokális müszaki okot és a müködési feltételeket, amelyek között a hiba következménnyé vált.

Hol kezdödik és hol ér véget a rendszer?

Erre nincs egyetlen, minden célra jó határ. A fejlesztö számára a rendszer lehet a saját kódbázis és az általa hívott szolgáltatások. Az ügyfél számára a márka, az ügyfélszolgálat és a fizetés együtt alkot egy rendszert. A jogi elemzésbe bekerülhet az adatkezelö, az alvállalkozó, a szerzödés és a panaszút. A biztonsági vizsgálat még a beszállítói láncot és a felhasználói kerülöutakat is figyelembe veheti.

A határ tehát vizsgálati döntés. Akkor hibás, ha a kérdés szempontjából fontos következmény rendszeresen kívülre esik rajta.



A rendszer nem azonos a felülettel - összehasonlító határtérkép

Egy AI-alapú értékelö eszközt például lehet pusztán modellként mérni: pontosság, hibák, késleltetés. Ha a rendszer munkavállalók besorolását támogatja, a vizsgálatba már bele kell férnie annak is, ki adja a célt, hogyan használják a pontszámot, ki kérhet felülvizsgálatot, és mi történik a vitatott esettel. A modellhatár technikailag tiszta lehet, miközben a döntési határ félrevezetö.

A rendszerhatár akkor jó, ha láthatóvá teszi a vizsgált felelösséget. Más kérdéshez más térkép kell, de a térképváltást nem szabad úgy elöadni, mintha ugyanazt a tárgyat néznénk változatlanul.

A dokumentáció nem díszlet a rendszer körül

A dokumentációt sokszor úgy kezelik, mint valamit, ami a kész termék után következik. Pedig a rendszer leírása maga is ellenörzési eszköz. Ha nem lehet világosan megmutatni, milyen állapotok, szerepek, adatmozgások és hibautak léteznek, akkor jó eséllyel a müködésben sincsenek teljesen rendezve.

Fractal Solutions

Egy használható rendszerleírásnak nem kell mindent egyetlen óriásábrába zsúfolnia. Külön nézet kellhet a felhasználói útra, a felelősségre, az adatra, az integrációkra, a jogosultságokra és az incidensekre. A nézetek közötti kapcsolat viszont nem maradhat puszta feltételezés.

A dokumentáció legerősebb mondatai gyakran feltételesek. Ha ez a külső szolgáltatás kiesik, akkor ez a folyamat áll meg. Ha ezt a szerepet visszavonják, akkor ezek az aktív ügyek maradnak gazdátlanul. Ha az AI-javaslatot felülírják, akkor az eredeti kimenet és az indoklás eddig örzödik meg. Ezek a mondatok nem a rendszert gyengítik. Megmutatják, hol van valódi döntési pont.

Az egyszerü felület és az öszinte rendszer

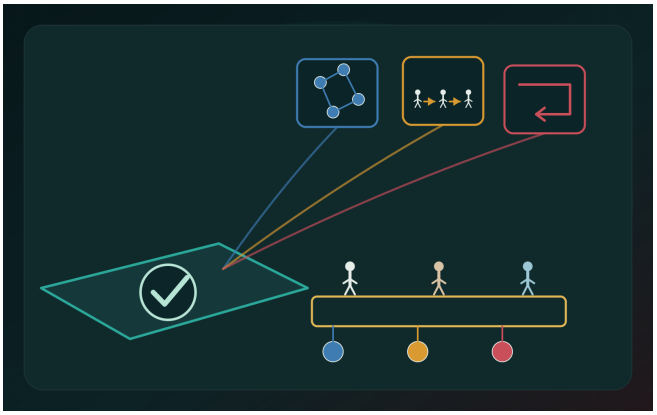
A felhasználónak nem kell minden adatbázist, szolgáltatást és felelösségi vitát látnia. Egy jó felület éppen attól használható, hogy a pillanatnyi feladathoz szükséges információt emeli ki.

A szervezet azonban nem engedheti meg magának ugyanezt a vakságot.

Belül láthatónak kell maradnia, miböl áll az egyszerüség. Melyik rész automatizált, hol dolgozik ember, mi történik kivételnél, kinek van joga beavatkozni, milyen adat támasztja alá a döntést, és hol lehet megállítani vagy visszafordítani a folyamatot.

A felület akkor tisztességes, ha nem ígér többet, mint amit a rendszer elbír. A rendszer pedig akkor érett, ha a háttér összetettsége nem ürügy a felelösség elmosására.

A pipa továbbra is lehet egyszerü. Csak tudnunk kell, mi történt mögötte.



A rendszer nem azonos a felülettel - rétegzett rendszeratelier

Hivatkozások

[1] Norman, Don A. (2013): The Design of Everyday Things. Revised and Expanded Edition. Basic Books.

[2] Suchman, Lucy A. (2007): Human-Machine Reconfigurations: Plans and Situated Actions. 2nd edition. Cambridge University Press.

[3] Leveson, Nancy G. (2011): Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press.

[4] Sculley, D. et al. (2015): Hidden Technical Debt in Machine Learning Systems. Advances in Neural Information Processing Systems 28, 2503-2511.

[5] Selbst, Andrew D. - Boyd, Danah - Friedler, Sorelle A. - Venkatasubramanian, Suresh - Vertesi, Janet (2019): Fairness and Abstraction in Sociotechnical Systems. Proceedings of FAT '19, 59-68. DOI: 10.1145/3287560.3287598.

Fractal Solutions

[6] National Institute of Standards and Technology (2023): Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. DOI: 10.6028/NIST.AI.100-1.

Fractal Solutions