Tekoälyn testaus

Matti Vuori www.mattivuori.net matti.vuori@mattivuori.net @Matti_Vuori

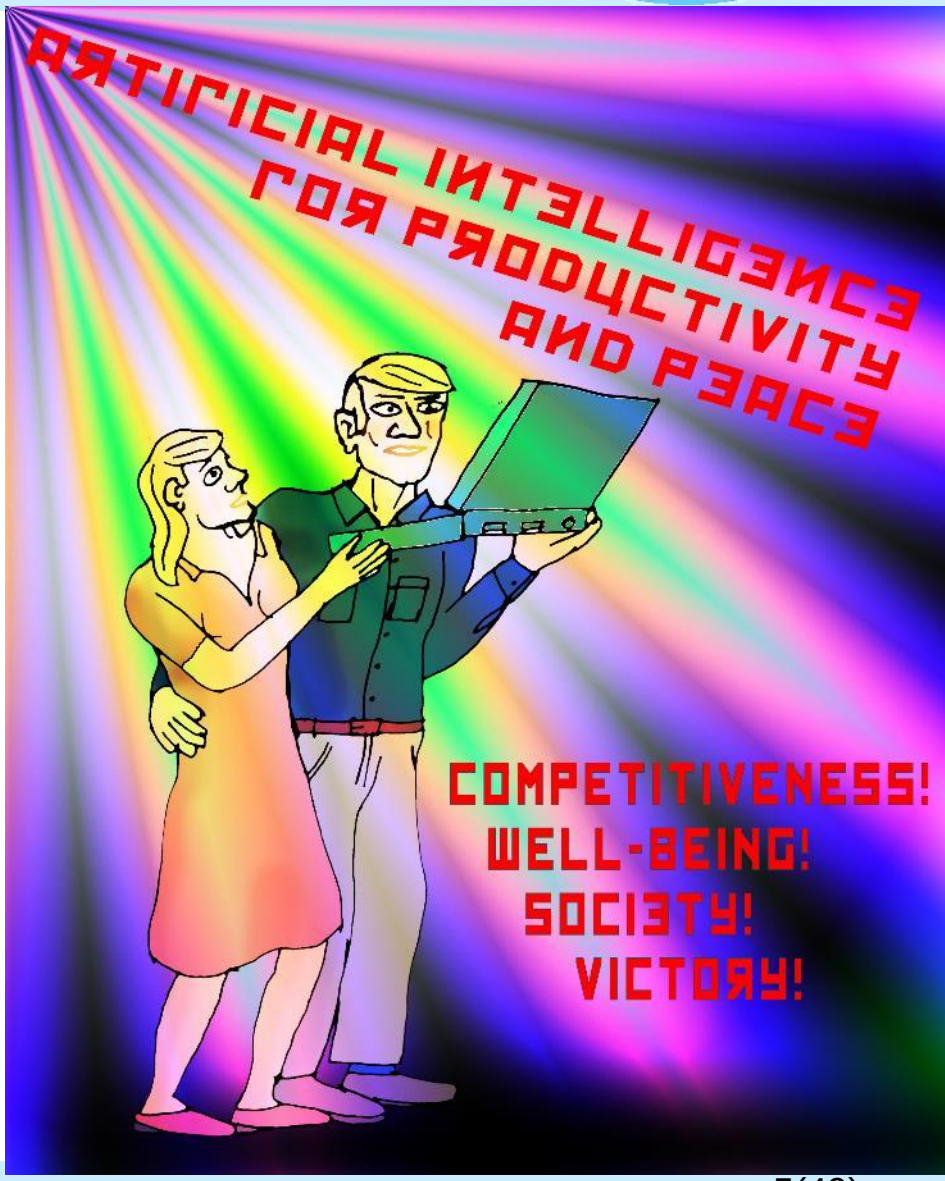
Sisällysluettelo 1/3

Sisällysluettelo 2/3

Sisällysluettelo 3/3

Suhteemme tekoälyyn 1/2

- Tekoälystä puhutaan paljon, se on lähes menestyksemme hopealuoti...
- Testauksen idea on luoda tolkkua (sensemaking) uusiin asioihin, nähdä selväjärkisesti hypen läpi ja auttaa uuteen teknologiaan liittyvässä päätöksenteossa.
- Testauksen (ml. epäempiiriset tuotteen arvioinnit) pitää unohtaa lyhenteet ja lupaukset ja lähestymistavoillaan paljastaa todellisuus.



Suhteemme tekoälyyn 2/2

- Tekoälypuhe on usein teknistä, detaljista, toimii teknologiaretoriikan ehdoilla.
- Jotta testaus tuo lisäarvoja ja ottaa paikkansa, sen pitää tuoda toinen näkökulma asioihin eikä antautua toisten kielipeliin.
- Tämä ei merkitse omaa kielipeliä, vaan selkeyden.
- **Ajatelkaamme siis omilla aivoillamme, toistamatta älykauppaiden retoriikkaa.**
- **Tämän esityksen idea onkin nostaa esille asioita, joissa on ajattelemisen aihetta.**

Edes FBI ei osaa: Kasvojentunnistus ei toimi

- https://www.inverse.com/article/29470-facial-recognition-fbi
- “third-party investigators said the entire system was reckless, unproven, and biased during a House Oversight Committee hearing”
- “FBI’s own tests show the system is only somewhat accurate. When the system was asked to pull the 50 closest-matching faces from a set of nearly 1 million, it got the right one only 86% of the time”
- “The Bureau has not tested the accuracy rate when pulling candidate lists of fewer than 50 potential matches”
- “hasn’t done any false positive testing to see how often non-matching faces are flagged as potential hits”
- **…Mitenkähän hyvin pienemmät toimijat ja innokkaat startupit tekevät testauksensa?...**

Ei arvo sinänsä, vaan eritasoista lisäarvoa

- Tekoäly ei ole arvo sinänsä. Sen käyttö on iso arvolupaus:
  - Voi poistaa tylsiä tai vaarallisia töitä.
  - Auttaa vaikeissa asioissa.
  - Korostaa sitä, mikä tekemisessä / asiassa on hienoa.
  - Vähentää työtä, kustannuksia, vähentää työvoimaa.
- Tekoälyä ei kannata ajatella binäärisesti – on tai ei, vaan miettiä, miten kuhunkin asiaan saisi etua lisäämällä ohjelmallista älykkyyttä, joka kenties hyödyntää kokemuksia ja kerättyä dataa.

Uusi konsepti, uusi teknologiapaketti

- Teollisessa kulttuurissa ihmiset ekstrapoloivat testauksensa lähestymistavan historiastaan ja edellisistä projekteista ”juuri riittävästi”.
- Kun testin alla oleva järjestelmä ottaa ison hyppäyksen haasteissa koko testausta pitäisi arvioida uudelleen.
- Olisi virhe ajatella tekoälyä vain yhtenä uutena ohjelmoitavana laitteena, ja automaation uutena tyyppinä.
- **Pitää esiymmärtää konseptitasolla millaista on hyvä tekoäly ja tekoälytuote, jotta osataan tuottaa tietoa sen hyvyydestä ja asettaa oikeita odotuksia.**
- Toistaalta testauksen pitää lähteä ymmärtämisestä, sensemaking: miten uusi asia toimii ja käyttäytyy.

Mitä uutta eri tasoilla?

Taso	Uutta ja huomioonottettavaa
Ihmisen suhde älyyn	Inhimillistäminen, ihmettely -> tietoisuus tästä ja vastatoimia
Tuote- ja järjestelmäkonseptit	Disruptiivisia? Uusia? Arvioitava konseptitasoa, testattava käyttäjäkokemusta
Objektiivisesti, metodisesti kohdattava asia	Työkaluja löytyy asian hallintaan (siis pitäisi löytyä ammattilaiselta)
Käyttäytyvä systeemi	Älyn luonne ja logiikka pitää kenties tunnistaa testaamalla ”Toiminnon” sijaan tunnistus, päättely
Tekninen systeemi	Bittejä liikkuu, mutta testaustaktiikoita ja välineitä löytyy

Käyttäjäkokemus kriittinen varmistettava

- Laadun yhdellä ylimmällä tasolla on käyttäjäkokemus.
- Se on kriittistä startupeille ja kaikille uuteen konseptiin perustuville tuotteille.
- Mutta teknologiayritykset usein siinä huonoja.
- **Siis tekoälytuotteissa huippuoleellista varmistaa.**
- Ymmärrystä eri aikoina:
  - 1990: Auton ajo-ominaisuudet ovat parhaimmillaan, kun niitä ei huomaa.
  - 2000: Käyttöliittymä on parhaimmillaan, kun sitä ei huomaa; kun sitä on mahdollisimman vähän.
  - 2017: Tekoäly on parhaimmillaan, kun sitä ei huomaa (ellei kyseessä ole lelu).
- **Tekoäly ei ole itseisarvo, vaan sen avulla parannetaan sitä, mikä tuotteessa on hienoa tai vähennetään sitä, mikä ei ole niin hienoa.**

Käytettävyyden heuristisia periaatteeita

- Ihmisen ja AI:n työnjako on hyvä: kumpikin tekee hänelle paremmin sopivia asioita.
- Ihmisellä on viimeinen päätösvalta asioita.
- Älyyn pitää voida luottaa.
- On selvää, kummalla on kulloinkin kontrolli.
- Kontrollin vaihto on luotettava.
- AI yksinkertaistaa systeemiä käyttäjälle.
- AI sopii käyttäjän mentaalimalliin.
- AI viestii kuhunkin tilanteeseen ja olosuhteisiin sopivalla tavalla.
- AI ei vie ihmisen huomiota tehtävästä.
- AI toimii kaikissa olosuhteissa (vrt. automaattibussi, joka ei toimi, kun tiellä on lunta...)

Äly on aina kontekstissa

- Äly ei ole yksinään, se on aina jossain tuotteen tai järjestelmän kontekstissa:
  - Tarkoitus.
  - Toiminta.
  - Käyttäjät.
  - Edut.
  - Riskit.
  - Toimintaperiaatteet.
- ...**Ja älystä saa tolkkua vain kontekstissa** ja sitä pitää testata kontekstin näkökulmasta.
  - Tietokantojenkin yhteydessä kiinnostaa kyky selvitä transaktioista, eikä toiminta kaikilla mahdollisilla SQL-kyselyillä.

Monenlaisia älysysteemejä

- Eri tarkoituksia:
  - Diagnoosi.
  - Asiantuntijajärjestelmä.
  - Älykäs toiminnallisuus.
  - Ohjelmistorobotiikka.
  - Viestintä. Asiakaspalvelija.
  - Lisätty todellisuus.
  - Turvajärjestelmät ml. tietoturva.
  - Jne...
- Itsenäisiä teknisiä tai ihmisen apulaisia.
- Aika usein **lisäävät älyä systeemiin (Augmented Intelligence) tai tehostavat ihmisen älyä**.

Älyn määrä ja luonne vaihtelee

- Yksinkertainen sääntöpohjainen logiikka:
  - JOS jotain, niin SITTEN jotain.
- Oppivat systeemit.
  - Muotoilevat itse käyttäytymisensä opetutuksen ja oppiminen perusteella.
  - Epäeksaktia logiikkaa: todennäköisyydet, painokertoimet...
  - Data tärkeää – Big Data...
- Tietoiset järjestelmät. Ei vielä näköpiirissä...

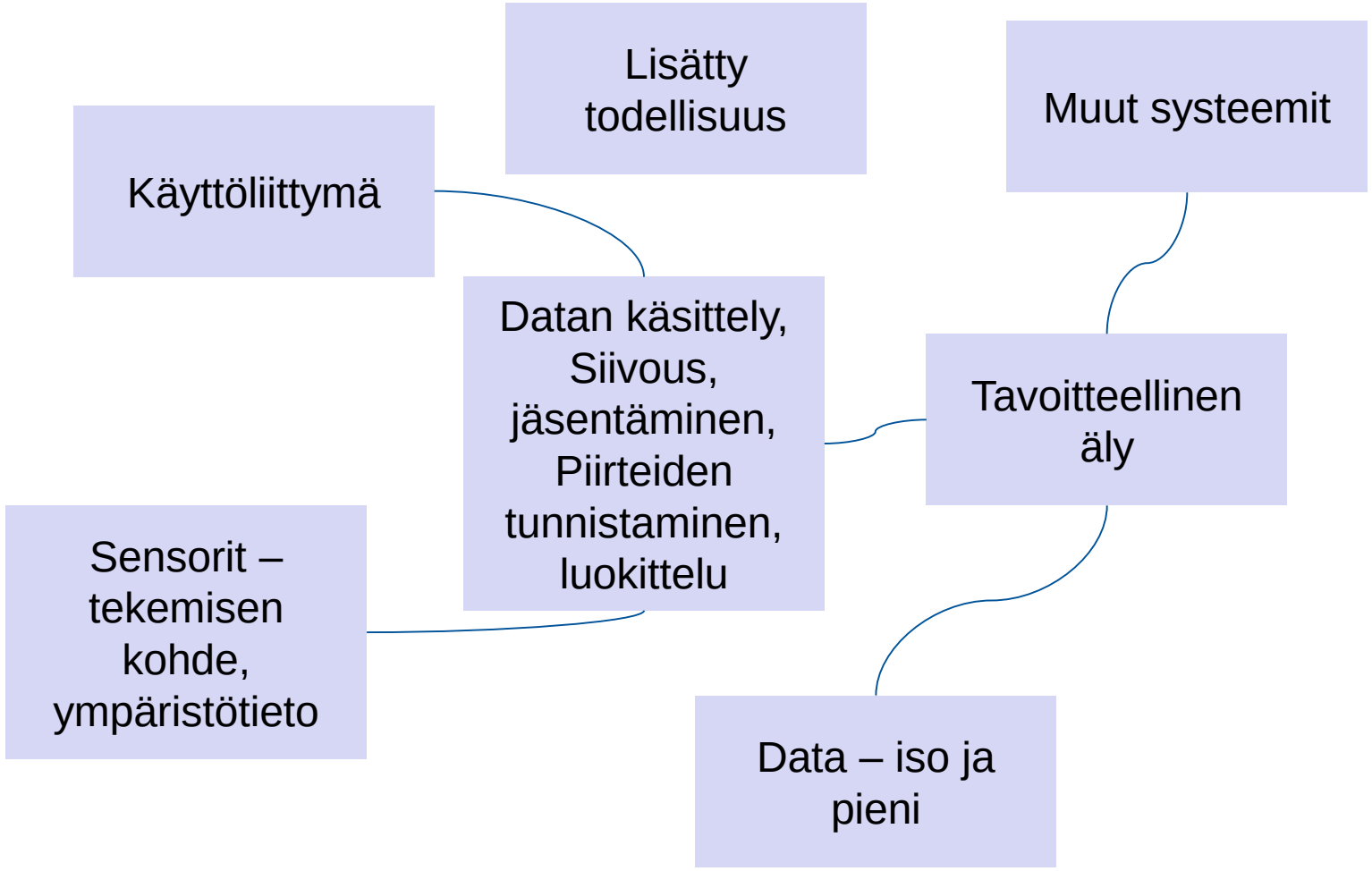
Ja ”älyn” kasvaessa kaikki muuttuu

- Epätietoisuus systeemin luonteesta.
- Epävarmuus sen luotettavuudesta ja turvallisuudesta.
- Oma kontrolli kenties vähenee.
- Epäluuloisuus kasvaa...

Tekoäly voi vaarantaa tietosuojasi, fyysisen turvallisuutesi tai elinkeinosi

Älysysteemin arkkitehtuuri

- Huom: Näitä on paljon erilaisia.



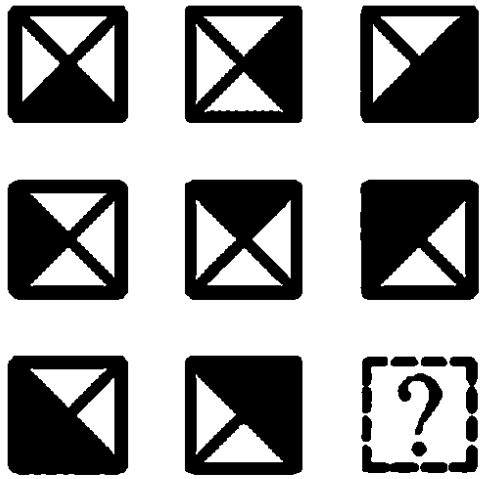
Älyn testaamisen haasteita 1/3

- Mitä älykkäämpi systeemi on, sitä mielenkiintoisia ilmiöitä ja ongelmia sillä on.
- Testauksessa pitää ajatella inhimillisiä virheitä – miten tekoäly voi tehdä niitä.
- Jos ja kun äly on ihmisen tukena, sen pitää löytää sopiva auttajan rooli. Äly on parhaimmillaan silloin, kun sitä ei huomaa!
- Älyn odotetaan osaavan perustella tekemisensä. Jos se ei onnistu, logiikan selvittämisessä on työtä.

```
AssertEquals(AI.Ask(”What is the meaning of life?”), ”42”);
```

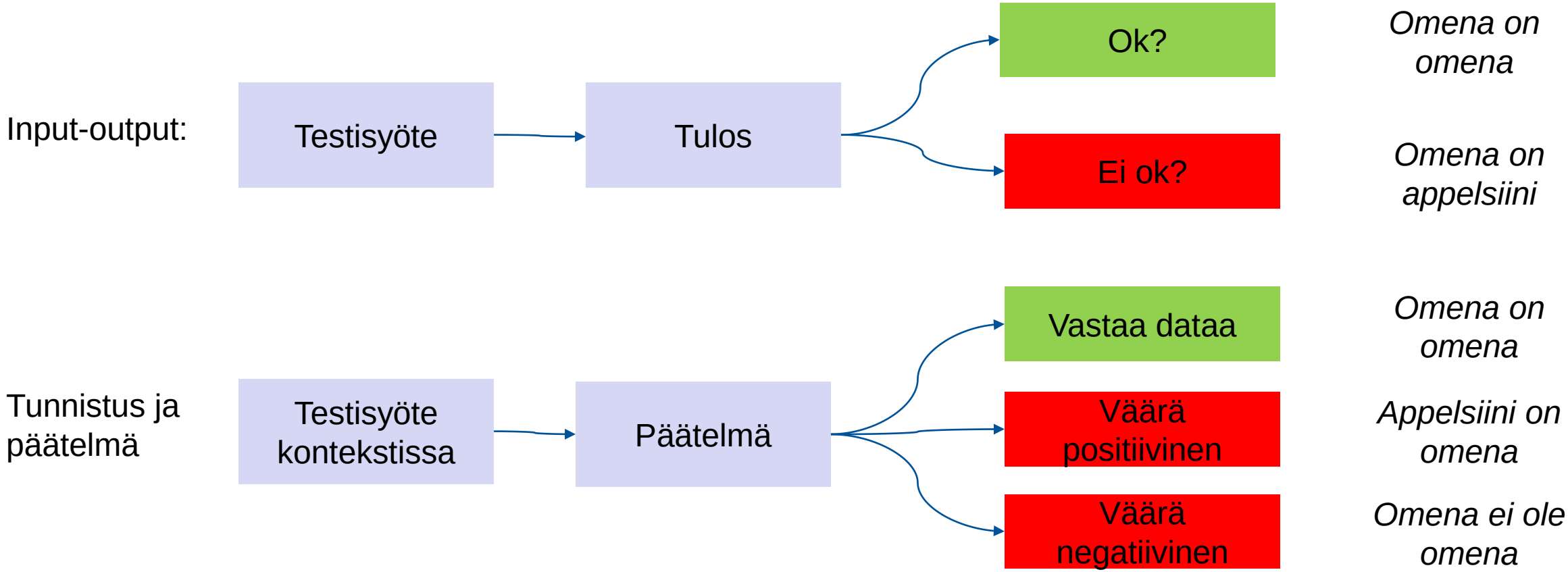
Älyn testaamisen haasteita 2/3

- Perinteisille systeemeille kerrotaan käytössä ja testauksessa asiat eksaktisti.
- Älykkäille asiat näytetään ja asiat eivät ole eksakteja, vaan kaikella on todennäköisyys.
  - Tunnistettiinko potilas oikein, onko nyt kodissa ”pimeä”.
- Normaalien systeemien logiikka voi täydentyä ja sitä voidaan tietoisesti muuttaa, mutta älykkään oppivan systeemin käyttäytyminen muuttuu huomaamatta opettamisen/oppimisen myötä.



Älyn testaamisen haasteita 3/3

- Tieteellisempään havaintojen tulkintaan



Olennaisia testityyppejä lyhyesti 1/4

- Konseptin arviointi.
  - Hyvien konseptien löytäminen on vaikeaa. Konseptin kriittinen arviointi on tärkeää.
  - Mihin uudesta ideasta on? Vertailu vastaaviin ja nykyiseen tilanteeseen. Onko se vaivan ja kustannusten arvoinen?
    - Kenelle, mihin kontekstiin? Vastaako tarvetta (toiminta, käyttäjät, markkinat)? Haluttavuus? Toimivuus käytännössä? Riskit? Teknologia? Eettisyys? Jne...
  - Kokeilut apuna. Analyysi. Tarkistuslistat.

Olennaisia testityyppejä lyhyesti 2/4

- Käytettävyys- ja käyttäjäkokemuksen testaus.
  - Ihminen-tekoäly-kokonaisuuden arviointi ja testaus.
  - Tarvitaan osaavia ammattilaisia. Toiminnallisen testauksen perinne ei riitä.
  - Eri kehitysvaiheissa: alussa ideoita, myöhemmin niiden validointia.
  - Käyttäjäkokemus ja käytettävyys.
  - Työnanalyysi osa laadun arviointia.
  - Analyyttiset arvioinnit. Tarkistuslistat.

Olennaisia testityyppejä lyhyesti 3/4

- Riski- ja luotettavuusanalyysit.
  - Riskianalyysi oleellinen disruptiivisille tuotteille. ”Mikä voisi mennä pieleen...”.
  - Kohteena työjärjestelmä, kokonaistuote, arkkitehtuuri, logiikka, data... Siis kaikilla tasoilla.
  - Luotettavuusanalyysi tarpeen, koska kokonaissysteemin teknologia monimutkaista.
  - => Tietoa kokeelliseen testaukseen.
- Tietoturvatestaus.
  - Lähtökohtana tietoriskianalyysi.
  - Mitä tietoa älysysteemi käyttää, tuottaa; mitä on varjeltava.
  - Kokonaisuus ja sen eri elementit ja näkökulmat.

Olennaisia testityyppejä lyhyesti 4/4

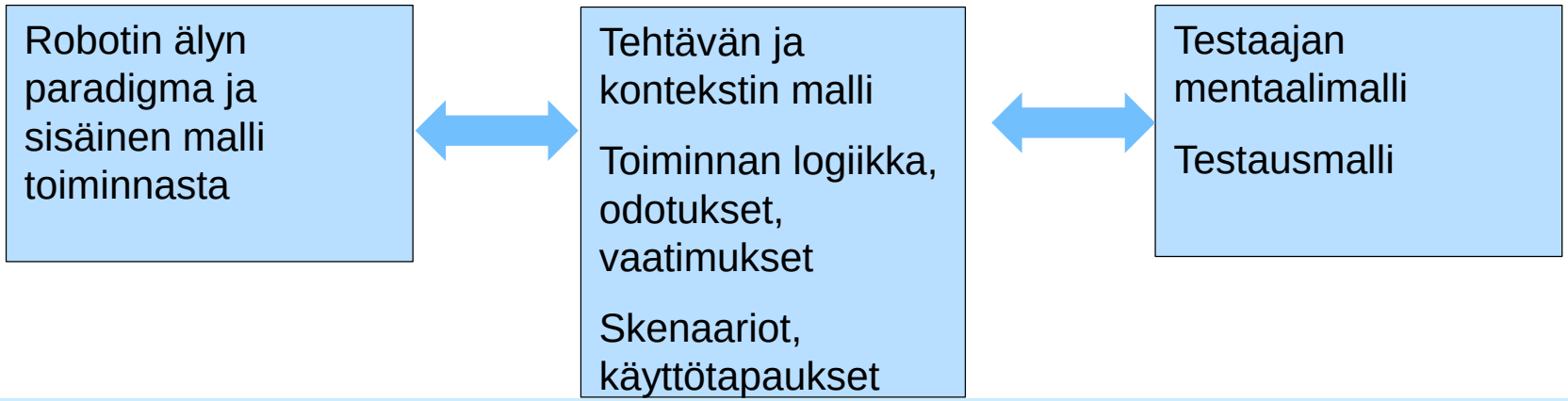
- Tekoälyjärjestelmän toiminnallisuustestaus.
  - Järjestelmän eri elementeille, eri abstraktiotasoille.
  - Tutkiva testaus, datatestaus.
  - **Hyvä uutinen: Järjestelmätasolla / ulkoisen käyttäytymisen tasolla ei edelleenkään tarvitse tuntea tekoälykomponenttien sisäisiä detaljeja (miten se hermoverkko toimii) – sen ymmärtäminen voi jäädä hermoverkkopalikan kehittäjälle.**

Älyn toimintalogiikan selvittäminen

- Ensimmäinen haaste on selvittää tutkivalla testauksella se logiikka, jolla systeemin äly toimii oikeasti.
- Mahdollisimman avoimia testiskenaarioita, jotta älyllä on liikkumavaraa.
- Älykästä järjestelmää ei saa kunnioittaa, vaan se pitää laittaa koville, ongelmiin ja umpikujiin. Tarvitaan lähes psykologin ajattelua.
- Olosuhteita, syötteitä ja muuta vaihtelemalla selviää, miten systeemi toimii.
- Perinteiset testaustekniikat, kuten päätöspuut, ekvivalenttiositus, raja-arvoanalyysi yms. ovat tärkeitä.

Testauksen kohteena käyttäytyminen suhteessa odotuksiin

- Testaajan ei järjestelmätasolla tarvitse tietää älykkyyden mekanismeja. Testauksen kohteena ei olekaan ”robotti” tai ”äly”, vaan käyttäytyminen.
- Ei tarvitse tuntea robotin sisäistä logiikkaa, vaan löytää hyviä testimalleja.
- Järjestelmätestaus ratkaisee. Se on systeemin validoinnin taso.

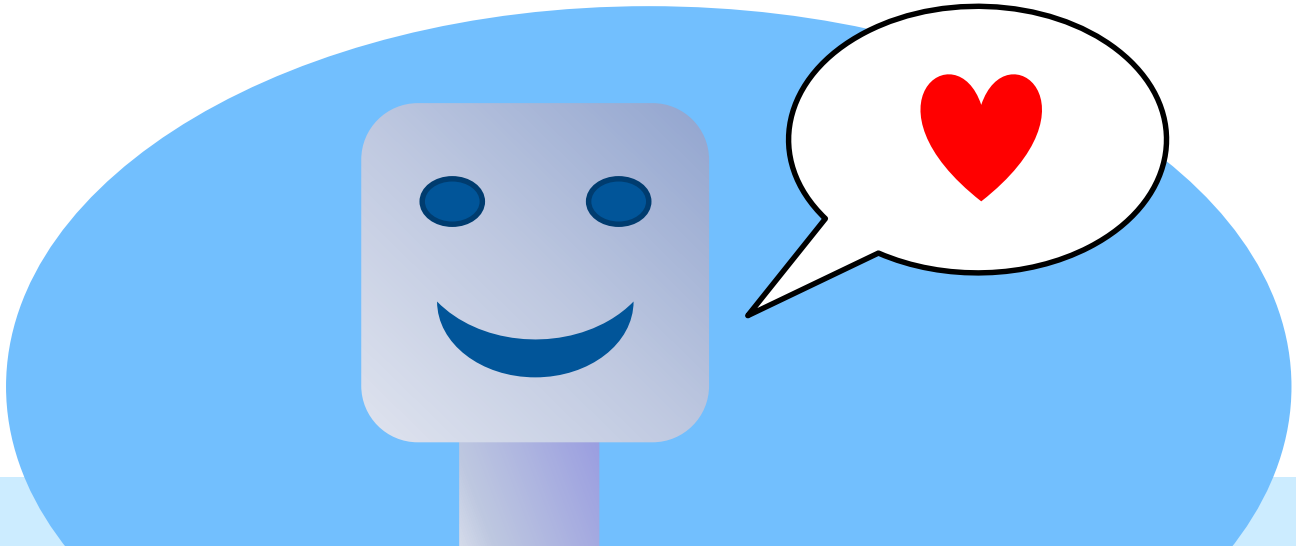


Onko äly sellaista kuin ihmisellä?

- Neuromorphisten tietokoneiden kehittäjät pyrkivät kehittämään ”koneaivoja”, jotka matkivat ihmisiä ja samalla edellytyksiä samankaltaiselle älylle.
- Miksei äly voi olla jotain ihan muunlaista?
- ...Sitä se tulee olemaankin...
- Testauksessa ei ainakaan pidä tehdä oletuksia.

Äly psykologisena haasteena

- Testaajat ovat ihmisiä – psykologia.
- Vaarana ihmetys, kunnioitus, huolenpito – hyvän testauksen vihollisia.
- Hyvän testauksen pitäisi tähdätä ohjelmiston rikkomiseen Ei saa välittää sen hyvinvoinnista.
- Mitä älykkäämpi systeemi on ja mitä enemmän se vaikuttaa inhimilliseltä, sitä enemmän sitä pitää tietoisesti koetella testauksessa.



Hajoita ja hallitse testauskohde teknisesti

- Systeemin eri elementit.
  - Sensorit – tunnistamisen rajat.
  - Datan luokittelija – oikeellisuus, luotettavuus...
  - Logiikka – päätteleekö äly oikein kaikissa tilanteissa.
  - Eri elementeille voi olla erilaisia lähestymistapoja. Päättelyn malleille validointi, sensoreille fuzzaus.
- Kokonaisjärjestelmä teknisesti ja datan kannalta.
  - End to end skenaariot, käyttötapaukset.
- Ihminen-tekniikka -järjestelmän analysointi ja testaus.
  - Työn analyysi.
  - Käytettävyys- ja käyttökokemus.
  - Riskianalyysi.

Hajoita ja hallitse tyyli älyn osa-alueittain 1/2

Osa-alue	Testattavia asioita
Logiikka ja kognitio	Tietojen käsittelyn oikeellisuus
Käyttäytyminen	Tilanteiden hallinta – normaalit tilanteet, poikkeustilanteet, vaaratilanteet Toiminta eettisiä valintoja edellyttävissä tilanteissa
Toiminta työssä	Tavoitteen ymmärtäminen ja säilyttäminen Työnkulun kokonaisuuden hallinta
Kommunikointi	Syötteiden ymmärtäminen Viestintä ulospäin Dialogin hallinta
Aistit	Sensorien tunnistuskyky, tarkkuus, robustius datalle ja olosuhteille

Hajoita ja hallitse tyyli älyn osa-alueittain 2/2

Osa-alue	Testattavia asioita
Luonne	Vuorovaikutustyylin sopivuus ihmiselle, kontekstiin, kulttuuriin
Roolin ottaminen	Sopiva rooli tehtävässä – apulainen, vastuullinen, tietotuki
Oppiminen	Opitun oikeellisuus Opetettavuus
Reflektointikyky	Tekemisen selittäminen, perustelu
Tietopohja (katselmointi)	Mitä tietoja hyödyntää – tietokannat, tiedonkeruu, asiantuntijoilta kerätyt säännöt
Ratkaisujen metataso (katselmointi)	Perusteet älytason ja muun konseptin valinnalle

Tekoälysysteemin käytettävyyden arviointi ja testaus 1/2

- Kun äly on ihmisen auttaja, on testauksen kohteena äly-ihminen -systeemi ja ihmisen ja älyn suhde.
- Perinteinen arviointitapa on kaksiosainen:

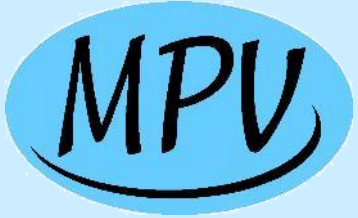
1) Analyysi:
   – Heuristinen arviointi.
   – Työn analyysi – tehtävän / skenaarion analysointi vaiheittain.
   – Tarkistuslistojen käyttö – konseptitasolta detaljeihin.
2) Käytettävyystestaus.

Tekoälysysteemin käytettävyyden arviointi ja testaus 2/2

- Käytettävyystestaus:
  - Ohjattu skenaario, jota koehenkilö toteuttaa.
  - Seurataan koehenkilöä ja tehdään havaintoja.
  - Koehenkilö ääneenajattelee tuntemuksiaan. Muut ovat hiljaa.
- Aluksi ja lopuksi haastattelu.
- Sitten havaintojen analysointi.

- Ks. Käyttöliittymien kehittämisen työkalupakki
- http://www.mattivuori.net/julkaisuluettelo/liitteet/tk-doit.pdf

Inhimilliset virheet 1/2



- Jens Rasmussenin jako:
  - Lipsahdukset: Taitopohjaiset virheet – tekoäly painaa vahingossa väärää nappia. Sensorivirheitä yms...
  - Sääntöpohjaiset virheet: Tuttujen tilanteiden sääntöihin liittyvät. Väärän säännön valinta, virheet säännöissä, vanhentunut sääntö jne... Olennaisia, kun systeemi perustuu staattisiin sääntöihin.
  - Tietopohjaiset virheet. Kun tekoäly joutuu ei-rutiinitilanteessa päättelemään, esim. hakee big datan perusteella potentiaalisimman vaihtoehdon toimenpiteelleen. Olennainen virhetyyppi. Oppiminen muuttaa toimintaa.

Inhimilliset virheet 2/2

- ”Toimintovirheet”, jotka liittyvät älyn tulkintavirheisiin ja ajoitukseen – älykäs voi olla hidas. Olennaisia roboteille:
  - Tehdään väärälle kohteelle (väärä tunnistus).
  - Jätetään jokin asia tekemättä.
  - Tehdään jotain ylimääräistä.
  - Tehdään kaksi kertaa.
  - Tehdään liian aikaisin.
  - Tehdään liian myöhään.
  - Yritetään tehdä, mutta epäonnistutaan.
- Tällaisia pitää testata sopivilla testeillä.

Älysysteemin riskianalyysin kysymyksiä (sampler)

- Toimii väärin?
  - Kokonaisjärjestelmän tasolla mietittävä.
- Muutokset:
  - Älylähde vaihdetaan toiseen?
  - Datalähde vaihdetaan toiseen?
- Väärinkäyttö:
  - Älyn käyttö väärään tarkoitukseen?
  - Älyn väärinopettaminen?
- Tilanteet:
  - Älyn toiminta poikkeustilanteissa?
  - Älyn käyttö eri kontekstissa?
  - Muutokset kontekstissa – toimijat, olosuhteet, välineet...
- Sabotaasi

Testattavuus – lokitus tärkeää

- Testattavuus on ikuinen ongelma...
- Koska on järkevää testata eri elementtejä erikseen, on huolehdittava testattavuudesta.
  - Esimerkiksi sensorien ja datan luokittelijoiden testaus erikseen ja yhdessä: tunnistetaanko kuvassa olevat asiat oikein? Muuten ei älyssä ole mitään tolkkua...
- Lokitusmahdollisuus on osa testattavuutta.
  - Koska tekoäly kaipaa paljon dataa on luonnollista, että dataa käyttäväkin systeemi tuottaa sitä ja tarjoaa sitä ulospäin.
  - Toimenpiteiden perustelu selväkielisesti on huipputärkeää lokitusta – sellaisesta lokista voi selvittää, miten äly tunnisti ja tulkitsi tilanteen ja miten toimenpiteet syntyivät.

Testidata suunniteltava huolella

- Opetusdata vs. testidata.
  - Jos testataan opetusdatalla, mitä se kertoo?
  - Regressiotestauksessa relevanttia.
  - ...Oppiessa suhde aiemmin kohdattuun dataan muuttuu...
- Datan merkitys korostuu:
  - Realismi.
  - Rikkinäinen, puuttuvan datan hallinta.
  - Toiminnon estävä data.
  - Robustius datan siivoukselle.
- => Datapohjainen testaus, fuzz-testaus oleellista.

Datan poikkeamatarkastelun tarkistuslista

- Äly tarvitsee dataa. Mitä tapahtuu jos:
  - Datalähde menee rikki ja dataa ei tule.
  - Data on väärää, faktuaalisesti virheellistä.
  - Datasta puuttuu asioita.
  - Dataformaatti on rikki.
  - Dataa on liian vähän.
  - Dataa on valtavasti.
  - Data on vinoutunuttta.
  - Data on väärästä lähteestä.
  - Datan lokalisointi on väärä.

Olosuhdetestaus

- Toimivuus vaikkapa niiden eri aistien kannalta haastavissa olosuhteissa: heikko valaistus, vastavalo, melu, erilaiset lattiaratkaisut jne…
- Ja tietysti kaikkien niiden vaihtelu.
- Esimerkiksi hahmontunnistus on ongelmallinen, jos se ei toimi luotettavasti hämärässä, vaan tuottaa vääriä tulkintoja tai taustamelu haittaa äänikomentoja.

Tietoturvatestaus

Taso	Älykkään systeemin erityispiirteitä
Yleistä	Ei ole ”AI OWASP”ia → pitää itse soveltaa ja tunnistaa systeemin mahdollisia haavoittuvuuksia.
Data	Tietosuoja ja data omistajuus vaikuttavat testattaviin asioihin.
Käyttäytyminen ja	Implisiittinen käyttäjän tunnistaminen (käyttäjän ”tapa olla ja tehdä”) eksplisiittisen sijaan. Ei saa tulla virheitä -> testattava sen robustius. Testattava pääsy opetustilaan ja opetusdatan lataus.
Arkkitehtuuri	Lisää uhkapinta-alaa uusista komponenteista ja datasta.
Matalan tason design, toteutus	Erilainen koodi: C++-koodia voidaan tarkastaa, katselmoida, mutta opetetulle hermoverkolle se on vaikeaa. Kehittäjät tekevät uudenlaisia ongelmia...

Arkkitehtuurin arviointi ja teknologioiden valinta

- Arkkitehtuurin arviointi on tärkeää, kun on vaihtoehtoja, kun systeemi on uudenlainen, kun on tiedossa muutoksia...
  - Älysysteemit ovat juuri tällaisia.
- Arvioinnin ei tarvitse olla raskas. Nopea skenaarioiden arviointi:
  - Älyn toteutuksen vaihto.
  - Datalähteen vaihtaminen.
  - Uutta dataa.
  - Jne...
  - ... Kaikki perinteiset järjestelmien muutos-skenaariot.
- Arviointi tukee systeemin komponenttien valintaa – ml. oikeanlaisen älymoottorin valinta

Turvallisuuskriittisten systeemien arkkitehtuuri

- Haasteista johtuen oppiva äly on hyvä pitää operatiivisessa järjestelmässä ja sen arkkitehtuurissa.
  - Käsitteellisesti, rakenteellisesti ja toiminnallisesti.
- Varsinaisia virallisia turvatoimia hoitaa tyhmä, muuttumaton järjestelmä.
- Äly voi tietysti tehdä operatiivisessa systeemissä ennakoivaa turvallisuustyötä.
- Tämä helpottaa systeemin validointia ja sertifiointia.
- Selkeä sääntöpohjainen äly, joka voidaan vaikka katselmoida, on helpoimpi tapaus.

Oppimisen testaus

- Osa älyä voi olla oppimiskyky. Testauksella pitää selvittää, että senkin mekanismit toimivat. Että äly:
  - Oppii oikeita asioita.
  - Oppii ne oikein.
  - Ei opi vaarallisia asioita.
  - Varmistaa oppimisensa käyttäjältä tarpeen mukaan.
  - Omasta älystään huolimatta tottelee käyttäjää.

Tuotantotestaus

- Älykkyyden kulttuuriin sopii se, että käytössä kaikki tapahtumat logitetaan ja:
  - Logeista tunnistetaan ongelmia ja viestitään tuotekehitykselle korjaustarpeet.
  - Logeista opitaan käyttöprofiileja, joita käytetään suunnittelussa ja testauksessa.
- Ongelmien tunnistus:
  - Poikkeus testimallista.
  - Poikkeus datassa jo olevista profiileista.
  - Asserttien laukeaminen, poikkeusten heitto.
  - Prosessorin kuormitus, toiminnon nopeus.
  - Jne...

Oppivan turvallisuuskriittisen systeemin kaksi ongelmaa

- Laitteet ovat vaarallisimmillaan poikkeus- ja häiriötilanteissa. Ne pitää testatakin kunnolla.
  - Käyttäytyminen sellaisissa voidaan ohjelmoida, mutta entä jo käyttäytyminen syntyy opettamalla? Jaksetaanko niiden opettamiseen panostaa?
  - Turvajärjestelmän merkitys korostuu, mutta tilanne ei ole helppo.
- Tietty konfiguraatio validoidaan testaamalla, mutta mikä merkitys on tietyn oppimistason validointitesteillä, kun robotin oppiminen muuttaa käyttäytymistä?

”Etiikan” testaus

- Viime aikoina on ollut esillä tekoälyn etiikka.
  - Autonomisen robotin suhde ihmiseen: voidaanko se esimerkiksi opettaa vahingoittamaan ihmistä? Ja milloin? Mitä kaikkea robotti saa tehdä pelastaakseen ihmisen?
  - Jos autolla on valittavana törmäys lapsiin tai vanhuksiin, kumman se tekee?
- Tällaisenkin käyttäytymisen testaus tulee jossain vaiheessa vastaan.
  - Pitää pystyä simuloimaan kaikkia tilanteita
- (Tietynlaista sääntöpohjaista tehtävään sidottua käyttäytymistä ei vielä voida pitää etiikkana, siksi lainausmerkit.)

Ylläpidettävyyden testaus

- Menneen ajan jäykkien sääntöpohjaisten järjestelmien iso ongelma oli ylläpidettävyys. Kun sääntöjä, dataa pitää muuttaa, se ei ole helppoa.
- Uusillekin järjestelmille voi olla, että ”Siperia opettaa...”.
- Ylläpidettävyyttä perinteisesti vain arvioidaan ja katselmoidaan, mutta sitä kannattaa testata:
  - Miten helppoa on opettaa systeemille uutta logiikkaa ja opettaa se pois vanhasta? Pitääkö opittu pyyhkiä kokonaan pois?
- Ylläpidettävyyteen liittyy myös siirrettävyys ja vaikka komponenttien vaihdettavuus, kun sopivampia (tai halvempia) tulee tarjolle.

Tarvittavissa kompetensseissa muutoksia

- Uudenlaisia testaajakompentensseja:
  - Koesuunnittelu – monipuotoisia, päteviä koeasetelmia.
  - Datatiedemiehen osaaminen.
  - Sensoriosaaminen.
  - Luotettavuustekninen osaaminen.
  - Jne...