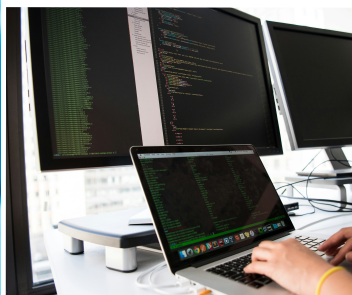


```
st] = parscala.Control.Catsem  
None  
tree.Defn.asObject(defn)  
  
Node], List[DotEdge]) = DotG  
ls)(  
er[tr.Decl, tr.Defn]) =>  
  
ecl) => tr.Decl.toDotGen(dec  
efn) => tr.Defn.toDotGen(defn
```



Tematikus kutatási együttműködések

**Innovatív Informatika és
Infokommunikáció
területén**

SZÉCHENYI 



MAGYARORSZÁG
KORMÁNYA

**Európai Unió
Európai Szociális
Alap**



EFOP-3.6.2-16-2017-00013

BEFEKTÉSAJÖVŐBE

PROJEKT CÉLJA

- az IKT és számítástudomány területén európai szinten is eredményesen működő kutatói hálózat létrehozása
- kutatások nemzetközi beágyazottságának erősítése, európai uniós pályázatokban való részvétel növelése
- kutatói utánpótlás nevelése

PARTNEREK

- Eötvös Loránd Tudományegyetem, ELTE (*konzorciumvezető*)
- Budapesti Műszaki és Gazdaságtudományi Egyetem, BME
- Pázmány Péter Katolikus Egyetem, PPKE

IDŐTARTAM: 2017 - 2020 (39 hónap)
KÖLTSÉGVETÉS: 957,50 millió Ft



TEVÉKENYSÉGEK

- Intézményeken átívelő kutatócsoportok formálása
- Fiatal kutatók nevelése, a projektbe való bevonása, segítése témavezetéssel, mentorálással
- Eszközbeszerzés
- Alapkutatás végzése 3 pillérhez kapcsolódóan
- Humánerőforrás bővítése, kutatók alkalmazása vidéki telephelyen
- Tudományos konferenciák szervezése
- Eredmények láthatóságának növelése: részvétel nemzetközi konferenciákon, publikálás folyóiratokban
- Partnerkereső, konzorciumépítő utazások, pályázati lehetőségek és témák kutatása, projekttervvé formálása
- Vendégkutatók fogadása, nagy hatású kutatók keynote speakernek invitálása
- K+F+I-menedzsment workshopok és tréningek szervezése



KUTATÁSI TERÜLETEK

A. Szoftverfejlesztési módszerek és informatikai biztonság pillér

A szoftverek komplexitásának növekedését a szoftverfejlesztési módszereknek és szoftverfejlesztést támogató eszközöknek is követniük kell. A kutatások ezen pillére a minden informatikai megoldásban központi szerepet betöltő szoftverek előállítására és minőségi jellemzőire (különös tekintettel a kiberbiztonságra) összpontosított.

B. Infokommunikációs infrastruktúrák és kiberfizikai rendszerek pillér

A mindennapjainkban használt szolgáltatások mögött Internet of Things eszközök állnak, míg az adatok tárolását, bárholonnan és bármikor történő elérését és feldolgozását felhő alapú alkalmazások biztosítják. Kutatásaikban nagy figyelmet fordítottak az IoT kommunikációs protokollokra, a valós idejű kiberfizikai rendszerekre, és a rádiós és vezetékes hálózati megoldások hatékonyságát növelő fejlesztésekre.

C. Intelligens adatelemzés pillér

A mindennapjainkban használt szolgáltatások mögött Internet of Things eszközök állnak, míg az adatok tárolását, bárholonnan és bármikor történő elérését és feldolgozását felhő alapú alkalmazások biztosítják. Kutatásaikban nagy figyelmet fordítottak az IoT kommunikációs protokollokra, a valós idejű kiberfizikai rendszerekre, és a rádiós és vezetékes hálózati megoldások hatékonyságát növelő fejlesztésekre.

EREDMÉNYEK



226 Publikációk száma



180 Kutatói utánpótlást támogató programokba bevont résztvevők száma



384 Hazai és nemzetközi tudományos konferenciaelőadás



91 A projektben közreműködő fiatal kutatók száma



15 Formális „szintlépés” vagy nemzetközi díj



149 Kutatás menedzsment és kutatási készségfejlesztési képzésekben részt vettek száma



54 Vendégkutatók száma



17 Új kutatási együttműködések száma



48 Potenciális Horizont 2020 együttműködő partnerek száma

```

def classes : List[tree.Defn.Object] = {
  topLevels map (_.fold(
    (decl : tree.Decl) => None
    , (defn : tree.Defn) => tree.Defn.asClass(defn)
  ))
}

def objects : List[tree.Defn.Object] = parscala.Control.catSomes(
  topLevels map (_.fold(
    (decl : tree.Decl) => None
    , (defn : tree.Defn) => tree.Defn.asObject(defn)
  ))
)

def toDot : dot.DotGraph = {
  val (nodes, edges) : (List[DotNode], List[DotEdge]) = DotGen.e
  for (nodes <- forM(topLevels)(
    (declOrDefn : Either[tr.Decl, tr.Defn]) =>
      declOrDefn.fold(
        (decl : tr.Decl) => tr.Decl.toDotGen(decl)
        , (defn : tr.Defn) => tr.Defn.toDotGen(defn)
      )
  );
  root <- DotGen.node(DotNode.record(
  yield <- mapM_(nodes, (n : DotNode) => {
    edges <- forM(n.edges)(
      (edge : DotEdge) => {
        (src, tgt) : (DotNode, DotNode) = edge.src <-> edge.tgt
        (src, tgt) : (DotNode, DotNode) = edge.src <-> edge.tgt
      }
    )
  })
  )
}

```