Modelling Dynamic Trust Contracts for Industry 4.0 Systems

Adrián Juan-Verdejo
PhD candidate and software engineer

Rima Al-Ali, Robert Heinrich, Petr Hnetynka, Adrián Juan-Verdejo, Stephan Seifermann, Maximilian Walter
Industry 4.0

1st: Mechanization, water power, steam power
2nd: Mass production, assembly line, electricity
3rd: Computer and automation
4th: Cyber Physical Systems
**Data sharing across supply chains**

- **Dynamic context**: analytics and predictions, legal framework, business process, roles, organisational structure, time, location and privacy.
valueStreamer: Shopfloor mgmt. system
ValueStreamer across the entire supply chain
State of the art

- **12 papers** around confidentiality in software design and software operation:
  - confidentiality analyses
  - access control
  - security enforcement
  - data sensitivity

- **Access rights mgmt**, **heterogeneous** and **dynamic** environments, **design time**, enforcement platform, **privacy levels** data sharing with operators
Trust 4.0 approach

Models of data flows and data manipulation

Models of privacy requirements

Trust runtime analysis
Models of data flows & data manipulation (1/2)

- **System and processes**

  - Individual data flows
  - Overall system behaviour
  - Context

- Palladio Component Model (PCM)
Models of data flows & data manipulation (2/2)
Trust 4.0 approach

Models of data flows and data manipulation

Models of privacy requirements

- Ensembles
- Membership
- Goal

Trust runtime analysis
A subcontractor’s technician fixing a machine may see its detailed error rates log but only together with a technician of the department responsible for the machine.
The **brake supplier** share its error rates with the head of a subsidiary company, e.g. for quality improvement, but only after proper **anonymization** to not reveal details of the manufacturing process.

```scala
class SharingWithSubsidiary(val company: Company) extends Ensemble {
  val machines = role(entities.select[Machine].filter(_.company == company))
  val persons = role(entities.select[Person].filter(
    _.hasRole{ case HeadOfCompany(subsidiary) => subsidiary.parentCompany == company }
  ))
  allow(persons, machines, "sum(errorRates)", PrivacyLevel.RESTRICTED)
}
```
Trust 4.0 approach

Models of data flows and data manipulation

Models of privacy requirements

Trust runtime analysis
1. PCM runtime analysis determines system characteristics such as privacy levels of sent data.
2. Trait-based Coalition Formation Framework (TCOOF) evaluates ensembles and derives decisions by means of access permissions.
Trust analysis (2/3)

1. PCM

- Characteristic: privacy levels of data
- Location: transmission link
- Analysis: data processors of all possible paths of data to location and effects of processing
- Result: average error rates
2. Decision Making based on TCOOF

1. Which contracts are relevant?
   - Relatively new in dynamic systems
   - Ensemble -> constraint solving problem (Choco)
   - Constraint (membership, context)

2. Allow/deny data sharing
Conclusion

- **Dynamic systems** with architectural modelling
- **Security and privacy not static**
  - not only roles and hierarchy but roles + dynamic context
  - coordination of multiple parties

- **Contributions**
  - **data flows** described in **system architecture**
  - dynamic **data sharing** based on context
  - runtime **analysis platform** for data exchange
Next steps

- Refine requirements. Technical report. Two pilots
- Simplify domain specific language
- Offer security and privacy guarantees at design time in dynamic systems
- Scalability and decentralise security and privacy
- Validation in two industrial pilot demonstrators
Contributions

1. data flows described in system architecture
2. dynamic data sharing based on context
3. runtime analysis platform for data exchange

Thanks!
Reference architecture