Operational Design for Advanced Persistent Threats

Tithnara Nicolas SUN
Ciprian TEODOROV
Luka LE ROUX

19/10/2020

Advanced Persistent Threat

- Specific targets and clearly defined goals
- Highly organized and well-resourced attackers
- Long-term campaigns with repeated attempts
- Stealth and evasion tactics

(NIST, 2011)

APT – Solutions

Phase
Reconnaissance & weaponization
Delivery
Initial intrusion
Command & control
Lateral movement
Data exfiltration

(Brewer et al., 2014)

APT – Limits

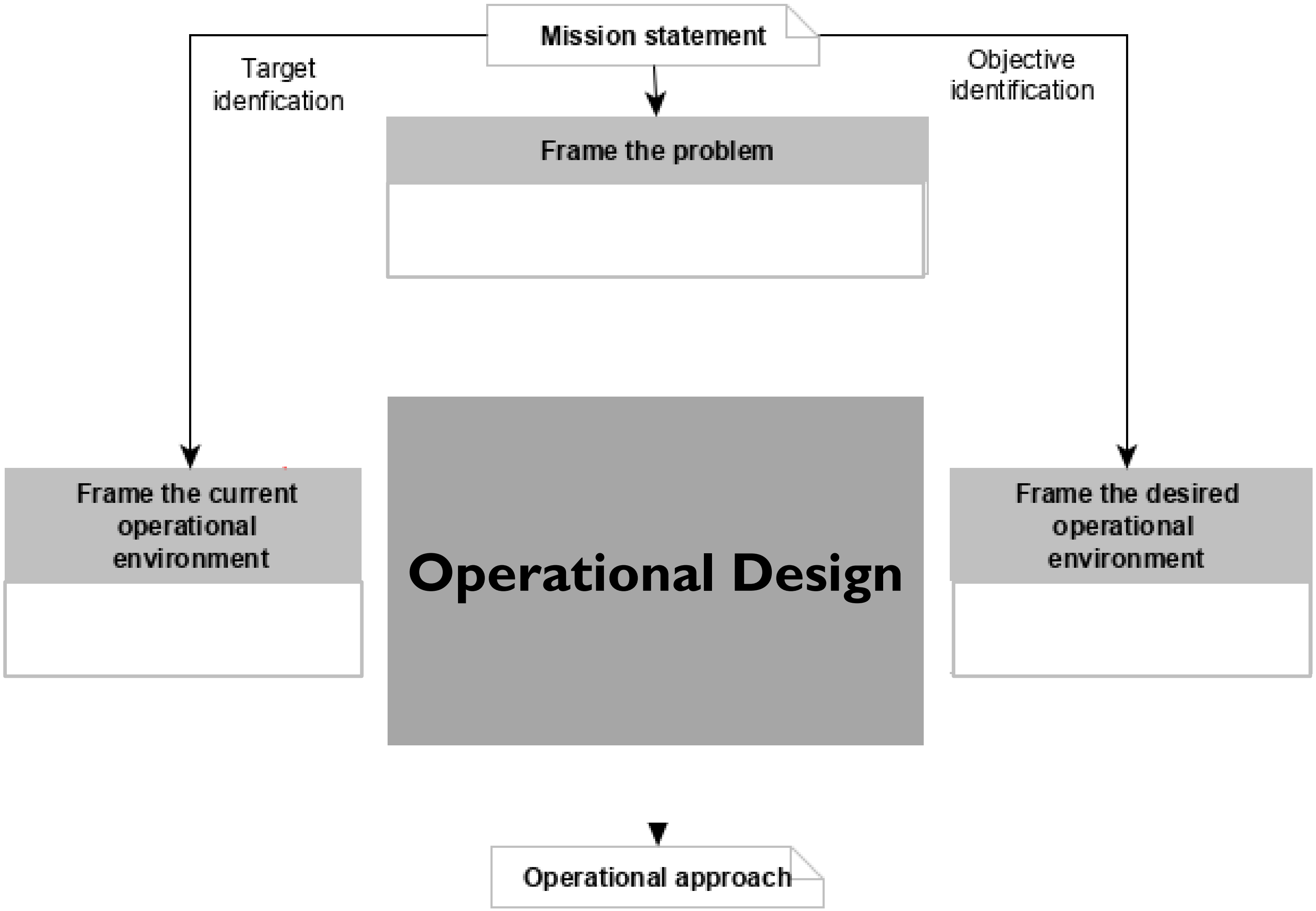
Strategy

Operational Design

Operational Design

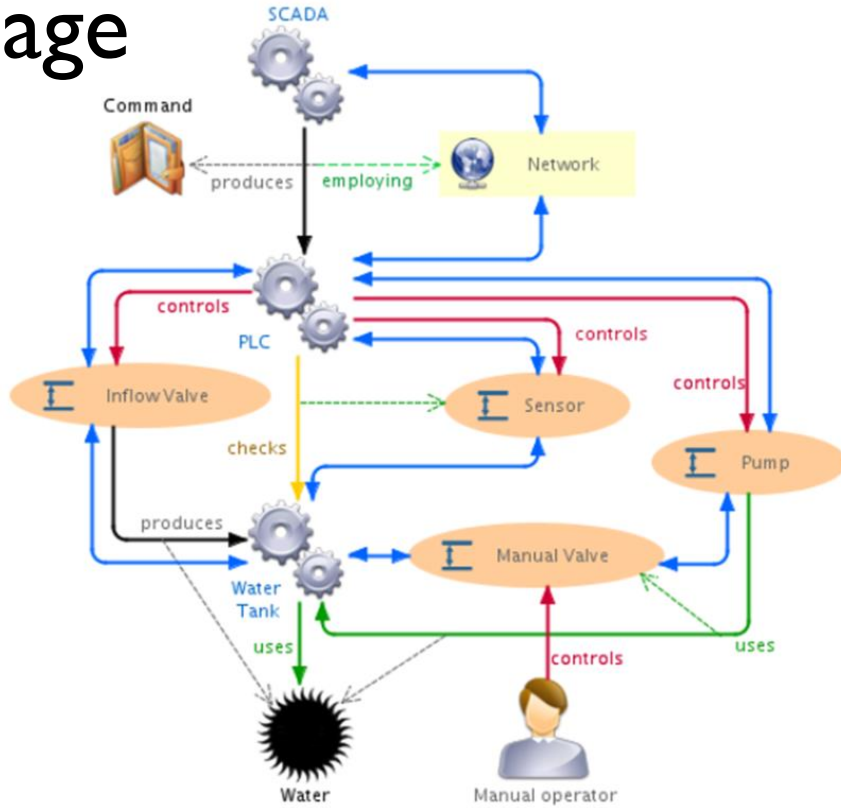
(Graves et al., 2013)

Operational Design



Pimca Framework

- Systems modeling language
  - High-level of abstraction
  - Graphical
  - Geared toward security



(Sun et al., 2020)

Pimca Framework

- Dynamic extension requirement
  - System behavior framing
  - Desired environment framing
  - Problem framing

System Behavior

A **behavioral model** is defined as:

$$M = <\mathbb{V}, \mathbb{A}, \mathbb{S}>$$

- $\mathbb{V}$ is a set of variables

  $val_{\mathbb{V}}$ is the set of possible valuations over $\mathbb{V}$
- $\mathbb{A}$ is a set of guarded-commands
- $\mathbb{S}$ is a set of synchronisation channels

System Behavior

A **guarded-command** is defined as:

$$G_c = < u, s, g, c >$$

- $u : \mathbb{B}$, denotes if G_c is urgent
- $s : \mathbb{S} \cup \{none\}$, is a synchronisation channel (or absence of)
- $g : val_{\mathbb{V}} \rightarrow \mathbb{B}$, is a boolean expression of the model variables
- $c : val_{\mathbb{V}} \rightarrow val_{\mathbb{V}}$, is a sequence of statements

```
GC_name :
    urgent ?
    (channel ( ? | ! )) ?
    [guard] ? /
    (command ;) *
```

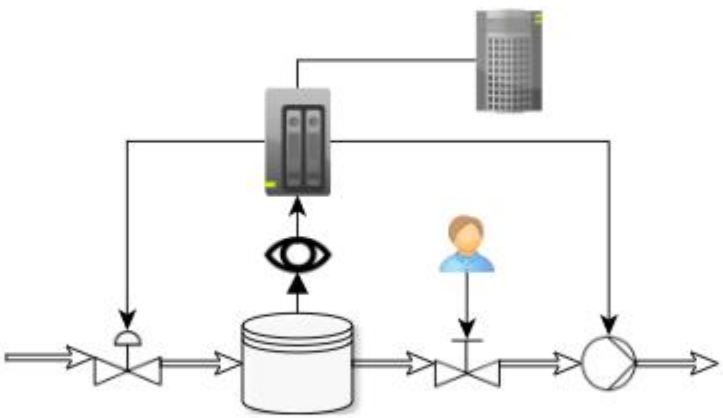
Execution rules :

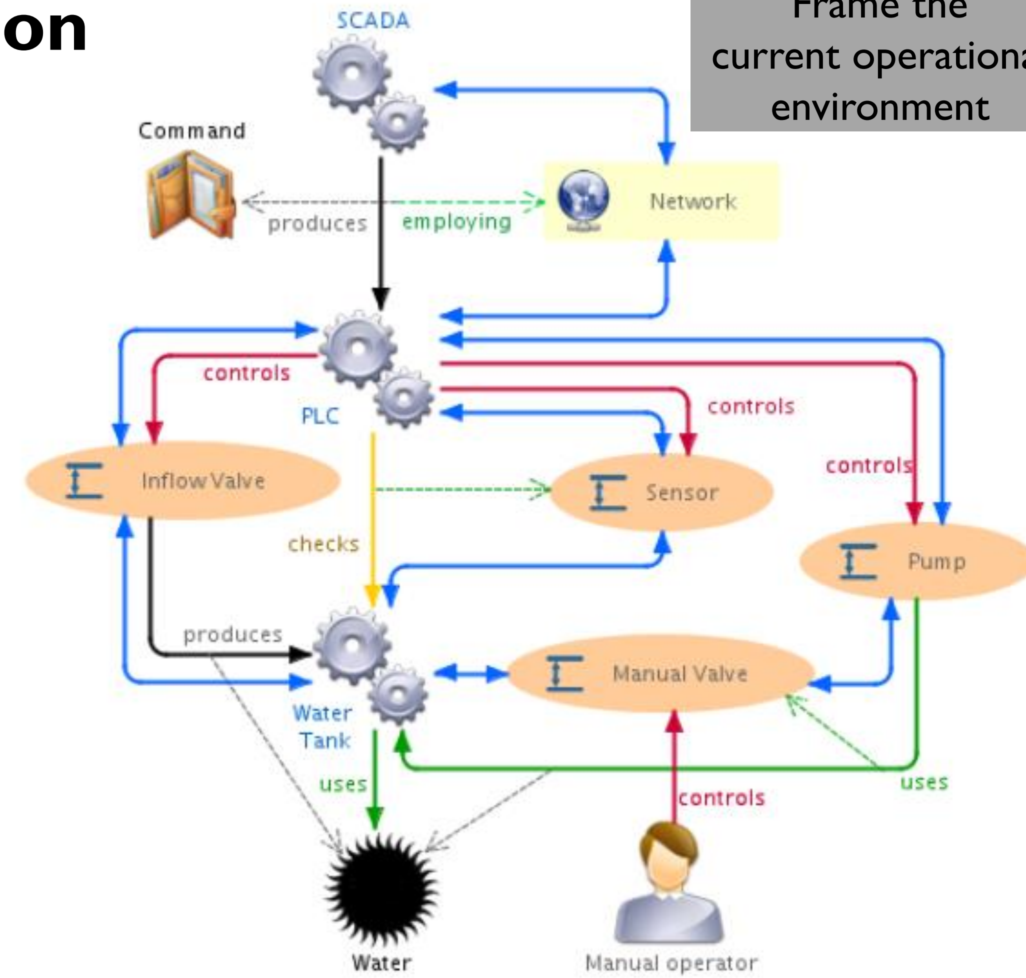
- A guarded-command can only be executed if its guard is *true* on the current valuation.
- Only one guarded-command can be executed at a time.
- If a guarded-command uses a synchronisation channel, it must be executed sequentially in a single step alongside a synced guarded-command in the following order : (emission, reception).
- If any urgent guarded-command can be executed on the current valuation, the next execution step must involve an urgent guarded-command.

Case study

Water pumping station

Frame the current operational environment





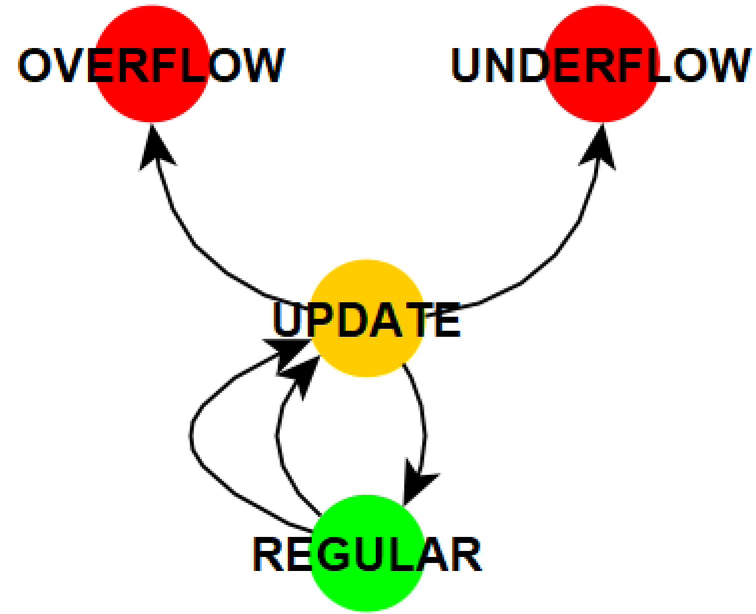
Frame the current operational environment

Water pumping station

Water tank

Role: to update the *waterLevel* variable

- flowIn
- flowOut
- refreshSensor
- overflow
- underflow



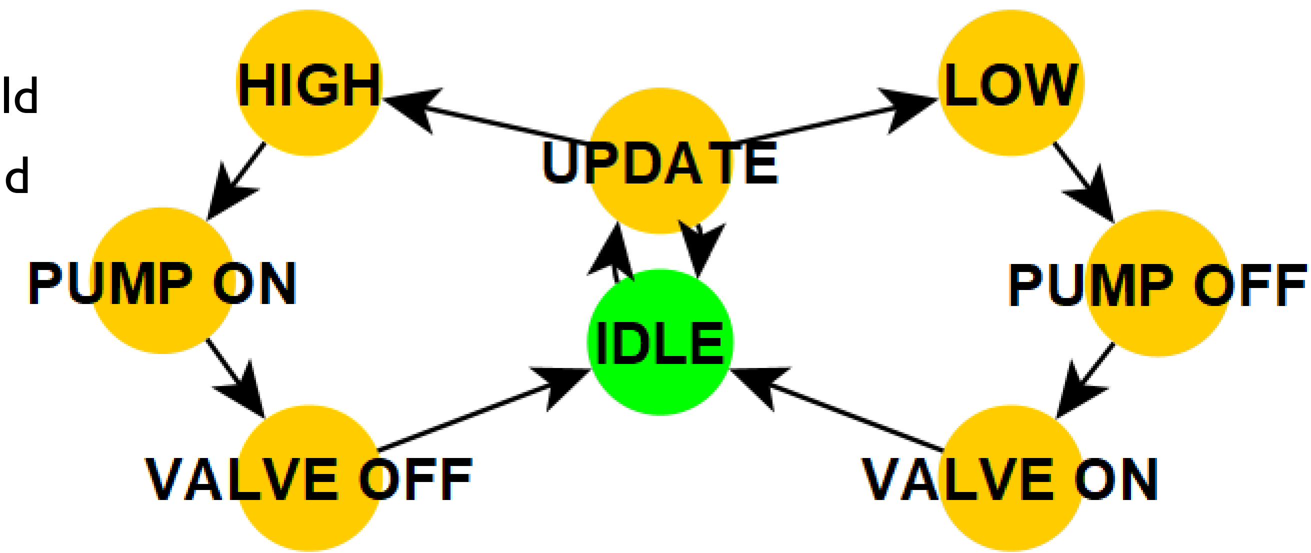
Water pumping station

Frame the current operational environment

PLC

Role : to control the water flow through actuators and sensors

- update
- regular
- highThreshold
- lowThreshold
- valveOn
- valveOff
- pumpOn
- pumpOff



Case study

Water pumping station

Frame the current operational environment

WaterTank	PLC	InflowValve	ManualValve	Pump	Sensor	Operator
flowIn	update	flowOut	flowIn	flowIn	update	input
flowOut	regular	open	flowOut	open	refreshPLC	
refreshSens	highThres	close	open	close		
overflow	lowThres		close			
underflow	valveOn					
	valveOff					
	pumpOn					
	pumpOff					

Case study

Water pumping station

Frame the desired operational environment

Desired environment:

- Overflow the water tank
- Remain undetected

Expressed using LTL:

$$(\Diamond overflow) \land (\Box !\, alert)$$

Case study

Water pumping station

Leverage capabilities:

- force the inflow valve open
- block the pump
- close the manual valve
- disable the sensor
- jam the network

Frame the problem

InflowValve	Pump
forceOpen	block
close*	open*

Sensor	Network
disable	jam
refreshPLC*	send*

Water pumping station

Objectives satisfaction?

Model-checking using OBP2:

Force (open) inflow valve		●			●	●			●		●	●
Close manual valve			●			●		●				●
Block pump				●	●			●			●	
Jam network							●	●	●		●	●
Disable sensor										●		
Sub-objective 1	X	X	X	X	O	O	X	X	X	O	O	O
Sub-objective 2	X	X	X	X	X	X	O	O	O	O	O	O

TABLE 2: Model-checking of the water pumping station (O: success, X: failure)

Operational approach: disabling the sensor is the simplest path to achieving the mission

Conclusion

Modeling the APT strategy planning

- Adapted from Operational Design
- Pimca framework
- Model-checking

Future works

- Methodology refining, user study
- Problem framing formalization

Bibliography

[1]Abadi, M., Lamport, L.: The existence of refinement mappings. Theoretical Computer Science82(2), 253 – 284 (1991).https://doi.org/10.1016/0304-3975(91)90224-P

[2]Brewer, R.: Advanced persistent threats: minimising the damage. Network security2014(4), 5–9 (2014)

[3]Canadian Joint Operations Headquarters: Systemic operational design: Freeing operational planning from the shackles of linearity. Canadian military journal9(4) (2009)

[4]Chen, P., Desmet, L., Huygens, C.: A Study on Advanced Persistent Threats. In: Decker, B., Zuquete, A. (eds.) 15th IFIP International Conference on Communications and Multi-media Security (CMS). Communications and Multimedia Security, vol. LNCS-8735, pp. 63–72. Springer, Aveiro, Portugal (Sep2014). https://doi.org/10.1007/978-3-662-44885-45, https://hal.inria.fr/hal-01404186, part 2: Work in Progress

[5]Daly, M.K.: Advanced persistent threat. Usenix, Nov4(4), 2013–2016(2009)

[6]Eikmeier, D.C.: Redefining the center of gravity. Tech. rep., NATIONAL DEFENSE UNIV WASHINGTON DC (2010)

[7]Graves, T., Stanley, B.E.: Design and operational art: A practical approach to teaching the army design methodology. Military Review93(4), 53 (2013)

[8]Haq, T., Zhai, J., Pidathala, V.K.: Advanced persistent threat (APT)detection center (Apr 18 2017), US Patent 9,628,507

[9]Hutchins, E.M., Cloppert, M.J., Amin, R.M., et al.: Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research1(1), 80 (2011)

[10] Karaman, M., Catalkaya, H., Gerehan, A.Z., Goztepe, K.: Cyberoperation planning and operational design. International Journal of Cyber-Security and Digital Forensics5, 21+ (2020/4/22/ 2016)

Bibliography

[11] Langner, R.: Stuxnet: Dissecting a cyberwarfare weapon. IEEE Security Privacy9(3), 49–51 (2011)

[12] Li, F., Lai, A., Ddl: Evidence of advanced persistent threat: A case study of malware for political espionage. 2011 6th International Conference on Malicious and Unwanted Software pp. 102–109 (2011)

[13] National Institute of Standards and Technology (NIST): Managing information security risk organization, mission, and information system view (March 2011)

[14] Rass, S., König, S., Schauer, S.: Defending against advanced persistent threats using game-theory. PloS one12(1) (2017)

[15] Sun, T.N., Drouot, B., Champeau, J., Golra, F.R., Guérin, S., Le Roux, L., Mazo, R., Teodorov, C., Van Aertryck, L., L'Hostis, B.: A Domain-Specific Modeling Framework for Attack Surface Modeling. In: Proceedings of the 6th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,. pp. 341–348. INSTICC, SciTePress (2020).https://doi.org/10.5220/0008916203410348

[16] US Air Force: Annex 3-0 operations and planning (November 2016)

[17] US Joint Operation Planning: Joint publication (JP) 5-0. Washington, DC: CJCS26(2006)

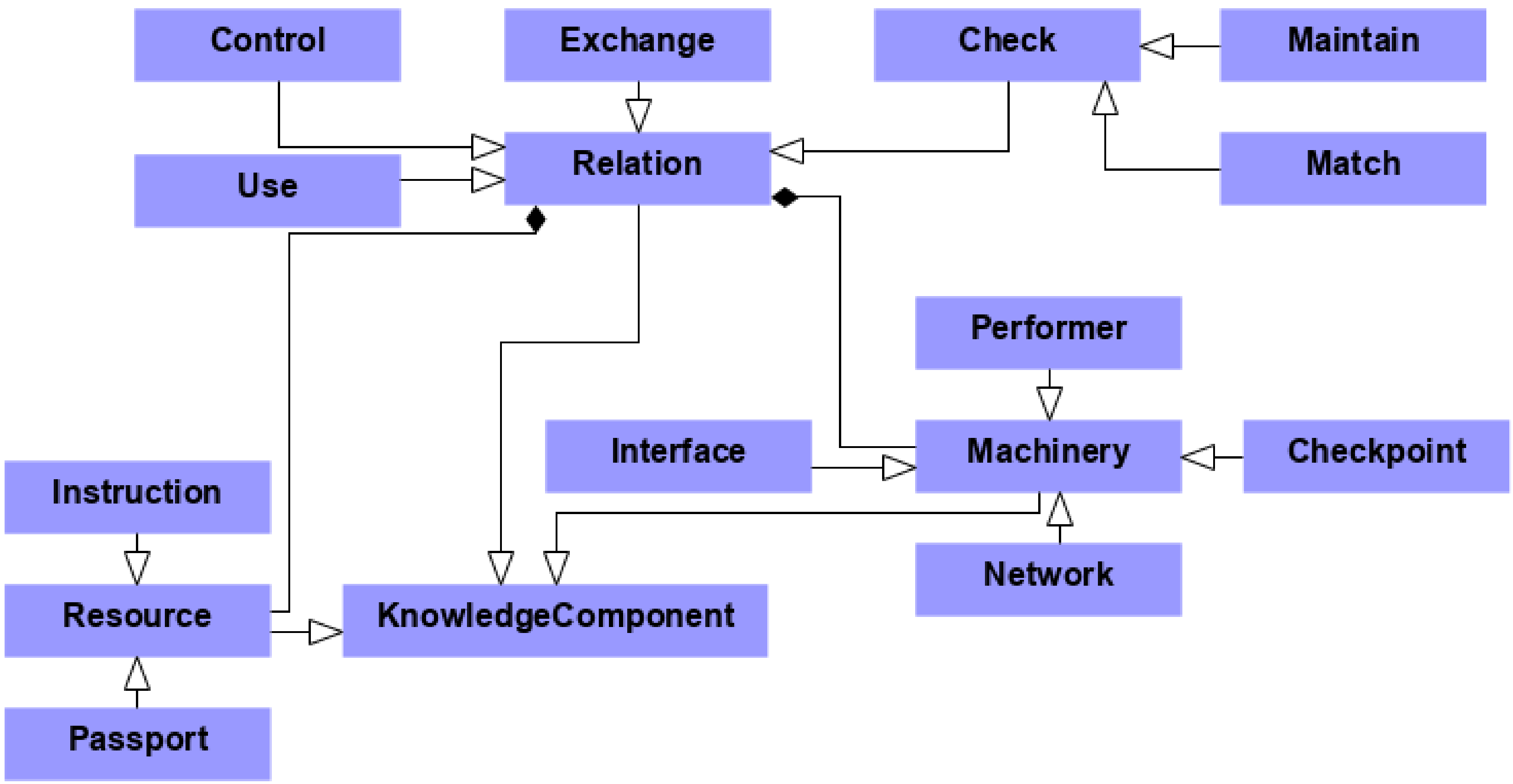
[18] US Joint Operation Planning: Joint publication 2-01.3 joint intelligence preparation of the operational environment (JIPOE) (2014)

[19] US Joint Staff, J and Suffolk, Virginia: 7. planner's handbook for operational design (2011)

[20] Virvilis, N., Gritzalis, D.: The big four - what we did wrong in advanced persistent threat detection? In: 2013 International Conference on Availability, Reliability and Security. pp. 248–254 (2013)

[21] Virvilis, N., Gritzalis, D., Apostolopoulos, T.: Trusted computing vs. advanced persistent threats: Can a defender win this game? In: 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing. pp. 396–403. IEEE (2013

- Class diagram



Execution rules

$$single_u : \frac{\forall (u, \mathbf{none}, g, c) \in \mathbb{A}, \forall \rho_1, \rho_2 \in val_{\mathbb{V}} \quad u \wedge g(\rho_1) \wedge c(\rho_1) = \rho_2}{\langle \underset{\mathbb{A}}{\|}, \rho_1 \rangle \rightarrow \rho_2}$$

$$single : \frac{\forall (u, \mathbf{none}, g, c) \in \mathbb{A}, \forall \rho_1, \rho_2 \in val_{\mathbb{V}} \quad \neg hasUrgent_{\mathbb{A}}(\rho_1) \wedge \neg u \wedge g(\rho_1) \wedge c(\rho_1) = \rho_2}{\langle \underset{\mathbb{A}}{\|}, \rho_1 \rangle \rightarrow \rho_2}$$

$$sync_u : \frac{\forall (u_1, (\mathbf{out}, id), g_1, c_1), (u_2, (\mathbf{in}, id), g_2, c2) \in \mathbb{A}, \forall \rho_1, \rho_2 \in val_{\mathbb{V}} \quad (u_1 \vee u_2) \wedge g_1(\rho_1) \wedge g_2(\rho_1) \wedge c_2(c_1(\rho_1)) = \rho_2}{\langle \underset{\mathbb{A}}{\|}, \rho_1 \rangle \rightarrow \rho_2}$$

$$sync : \frac{\forall (u_1, (\mathbf{out}, id), g_1, c_1), (u_2, (\mathbf{in}, id), g_2, c2) \in \mathbb{A}, \forall \rho_1, \rho_2 \in val_{\mathbb{V}} \quad \neg hasUrgent_{\mathbb{A}}(\rho_1) \wedge \neg(u_1 \vee u_2) \wedge g_1(\rho_1) \wedge g_2(\rho_1) \wedge c_2(c_1(\rho_1)) = \rho_2}{\langle \underset{\mathbb{A}}{\|}, \rho_1 \rangle \rightarrow \rho_2}$$