UMF IN A NUTSHELL
CAPABILITY LEVEL 1.1

+ TeamContext
  (team size;
  load, trust;
  stability, ...)

Contexts

KPI's

Teams

LISTEN to
Team interface

GOV

Any means (e.g. SNMP)
to manage the T.CONFIG

Decision

Sense

Rules

Act

Parameters

(T+C).Rules:=Team Behaviour Rules
On BOOT send JOIN(TEAM*);
On Ctxt_Change SEND(TEAM*, Ctxt_Change);
IF Time=Period & KPI_i<T_i THEN SEND(TEAM*, Utility_Promise), etc.

(D).(T+C).Rules:=Decision-in-Team rules
IF KPI_i<T_i & Promised Utility Increase is the Highest in the Team & Ctxt=Allowed THEN P:=P+δ
UMF IN A NUTSHELL
CAPABILITY LEVEL 1.2

+ Team Context
  (team size; load, trust; stability, ...)

Contexts

KPI’s

Groups

GOV

Decision

Rules

Act

Team

Parameters

LISTEN to Team interface

Team behaviour Rules, including
T.CONFIG = {teams, timers, msg. patterns, shared context, shared predictions, ...}

JOIN, LEAVE, PAUSE, SEND, VOTE, ...

Any means (e.g. SNMP) to manage the T.CONFIG

(D-\(T+C+K\)).Rules:=Decision-in-Team rules

IF KPI_i<T_i & Predicted Utility Increase is the Highest in the Team & Ctxt=Allowed THEN P:=P+δ

(T-\(T+C+K\)).Rules:=Team Behaviour Rules

On BOOT send JOIN(TEAM*);
On Ctxt_Change SEND(TEAM*, Ctxt_Change);
On PredictedUtility>Threshold SEND(TEAM*, PredictedUtility);

IF Time=Period & KPI_i<T_i THEN SEND(TEAM*, Utility_Promise), etc.
**UMF IN A NUTSHELL**

**CAPABILITY LEVEL 2.0**

---

**Rules**

**Decision**

**Sense**

**Act**

**Parameters**

---

**Contexts**

**KPI’s**

**Teams**

---

(D). T.G.Rules := Decision-in-Group under Governance rules

On ROLE_1: (=Team Leader) On JOIN: SEND(Team*, Status), ...

G.Rules := Governance Rules

GOV-NEM: START(NEM), START(Team); STOP(NEM), STOP(Team), REGISTER, ASSIGN_ROLE(), ...

---

Event, Date, Location

37
(D).(T).(G+C). Rules := Decision-in-Group under Governance rules
On ROLE_1: (=Team Leader) On JOIN: SEND(Team*, Status), ..., On WATCH(Ctxt): SEND(Team*, Ctxt:=Relevant);
(G+C). Rules := Governance Rules
GOV-NEM: START(NEM), START(Team); STOP(NEM), STOP(Team),
REGISTER, ASSIGN_ROLE(), WATCH (Context), STOPWATCH (Context), ...
**UMF IN A NUTSHELL**
**CAPABILITY LEVEL 2.2**

Build specific Know on behalf of Global GOV

- **Contexts**
- **KPI's**
- **Teams**

Share GOV Know building In Team

---

(D)(T).(G+C+K).Rules:=Decision-in-Group under Governance rules

On ROLE_1: (=Team Leader) On JOIN: SEND(Team*, Status), ..., On WATCH(Ctxt): SEND(Team*, Ctxt:=Relevant);
On BUILD(Know): SEND(Team*, Build(Know));

(G+C+K).Rules:=Governance Rules

GOV-NEM: START(NEM), START(Team); STOP(NEM), STOP(Team)
REGISTER, ASSIGN_ROLE(), WATCH (Context), STOPWATCH (Context), BUILD(Knowledge), STOPBUILD(Knowledge), ...

---

Event, Date, Location
0 – Reliable operation of a standalone NEM
  – 0.1: Reliable decision making under noise
  – 0.2: ... with context awareness
  – 0.3: ... with prediction

1 – Trustworthy interworking of NEMs in a Team
  – 1.0: Orchestrated Team work with maximal utility
  – 1.1: ... with sharing of relevant context changes
  – 1.2: ... with sharing of relevant predictions

2 – Seamless Deployment of NEMs, NEM Teams
  – 2.0: NEM/NEM Team Lifecycle Management
  – 2.1: ... with governed context sharing
  – 2.2: ... with governed knowledge building

Problem being solved by a UMF+NEM group (ecosystem)

Problem being solved by a NEM group (ecosystem)

Components of a Message

- Objective: Trust in Autonomics
- Audience: Operators
- Technology: P+SL+A
- Impact: Certification

Technology: Predicates + Subjective Logic + Assessment

Capability=2: ROLE a set of connected behaviours, rights and obligations as conceptualised by actors in a network situation