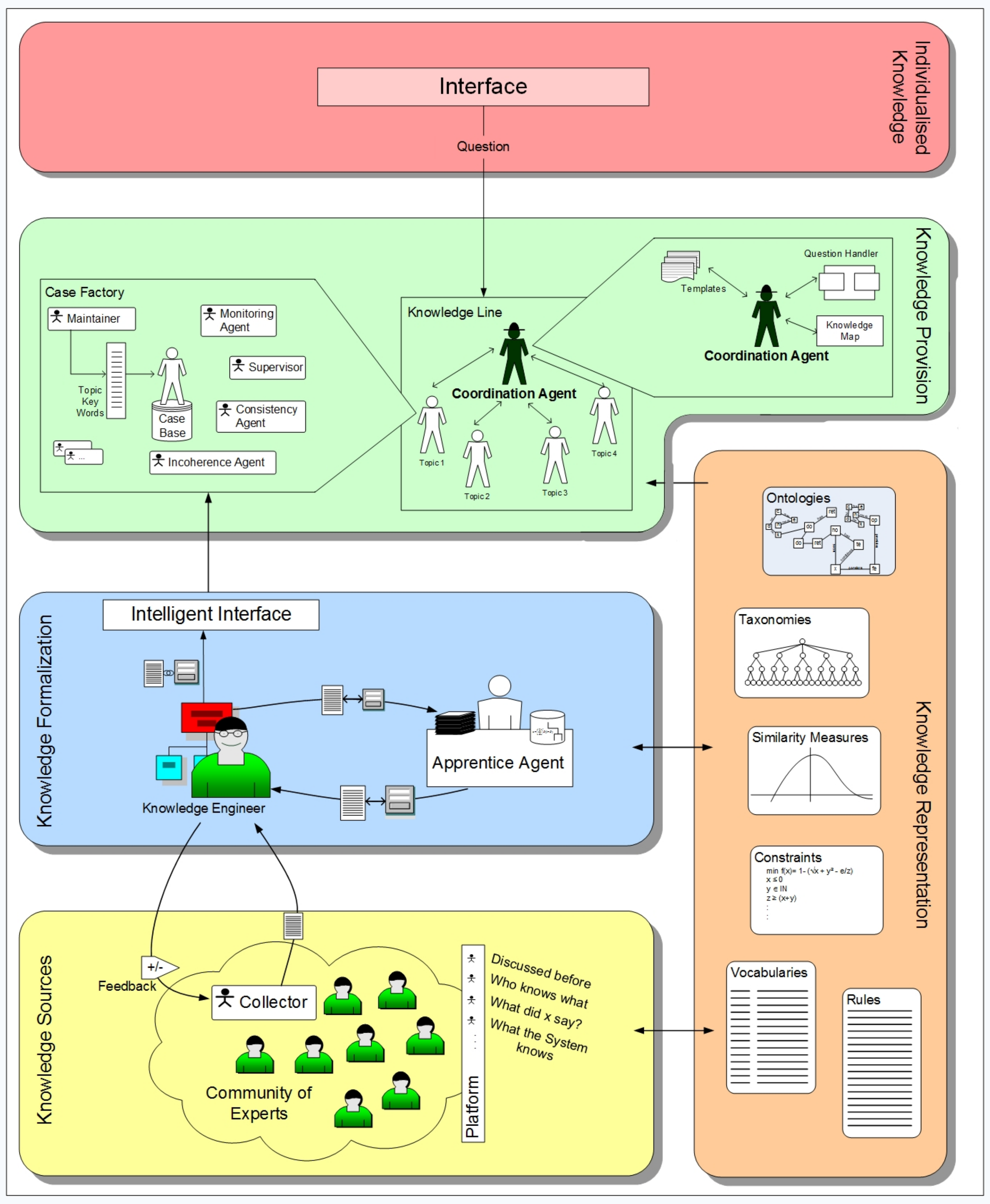
Dependency Modeling for Knowledge Maintenance in Distributed CBR Systems

Pascal Reuss[12], Christian Witzke[2], Klaus-Dieter Althoff[12]

reusspa@uni-hildesheim.de

SEASALT Architecture



Algorithm:

```
D = empty
    foreach (attribute a in A) {
        if (check ( $v_a$ exist in $c_{cb}$) {
            $d_u$ = new d ($v_a$ ,$v_c$ , u)
            if (exist (D, reverse($d_u$ s ))) {
                $d_b$ = new d($v_a$ ,$v_c$ ,b)
                D = D - reverse($d_u$)
            } else {
                D = D + $d_b$
            }}
        if (check ($v_a$ exist in $v_{fct}$ ) {
            $d_u$ = new d ($v_a$ , $v_{fct}$ ,u)
            if (exist (D, reverse($d_u$ ))) {
                $d_b$ = new d ($v_a$ ,$v_{fct}$ ,b)
                D = D - reverse ($d_u$)
            } e l s e {
                D = D + $d_b$
            }}
        if (check ($v_a$ exist in $v_r$) {
            $d_u$ = new d ($v_a$ ,$v_r$ ,u)
            if (exist (reverse($d_u$))) {
                $d_b$ = new d ($v_a$ ,$v_r$ ,b)
                D = D - reverse ($d_u$)
            } else {
                D = D + $d_b$
            }}}
return D
```

Definitions:

V_a	Set of values for attribute a
C_{cb}	Set of cases in a case base cb
v_a	specific value of attribute a
ccb	specific case of casebase cb
vfct	specific value in similarity measure
Vr	specific value in rule

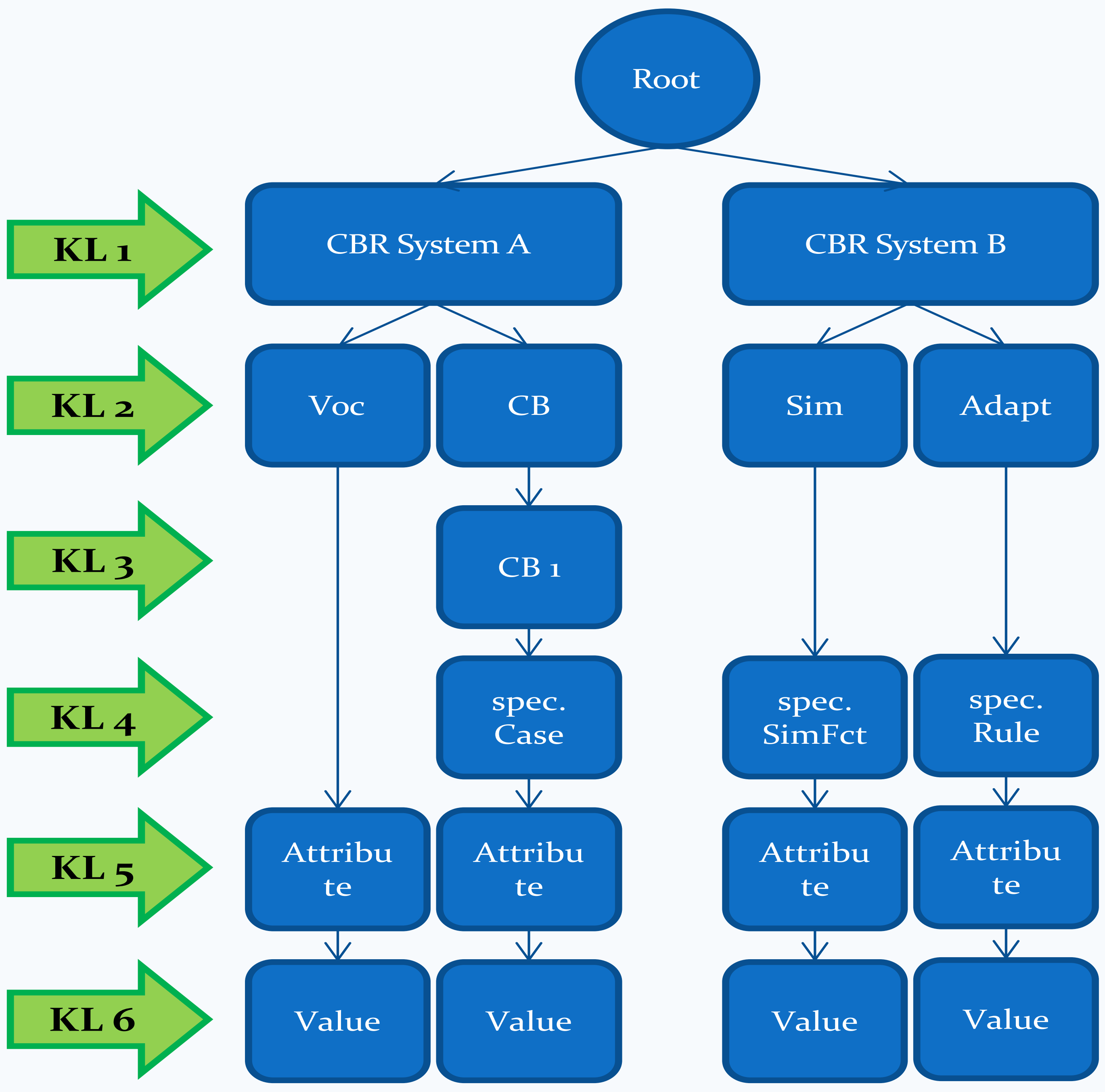
Input:

A	Set of attributes in the casestructure
CB	Set of casebases in a CBR system
R	Set of adaptation rules
S	Set of similarity functions

Output:

D	Set of syntactic dependencies

Algorithm for generation of dependencies



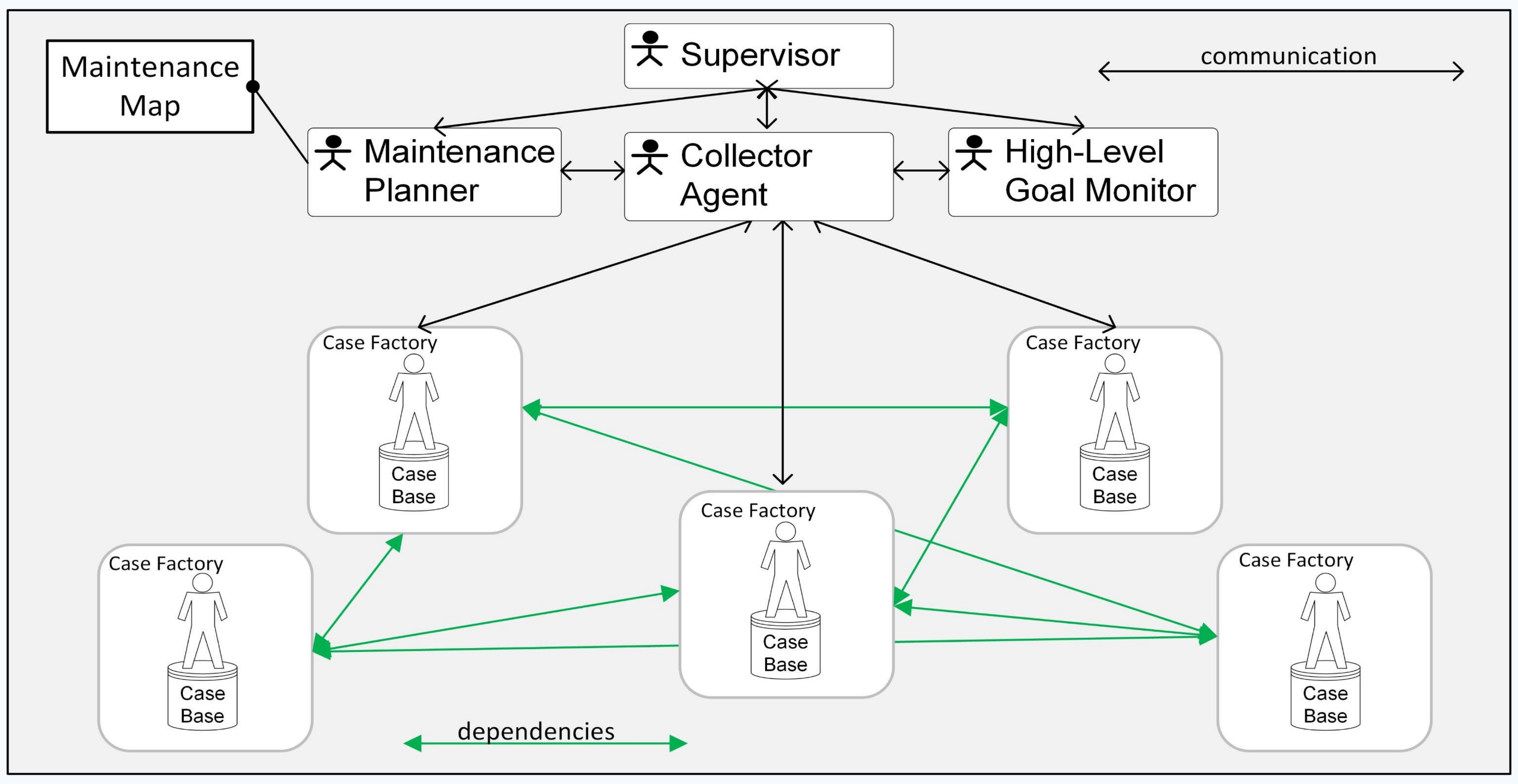
Generic hierarchy for granularity of dependencies

Formal Definition of a Dependency:

$$d = (kle_{source}; kle_{target}; t)$$

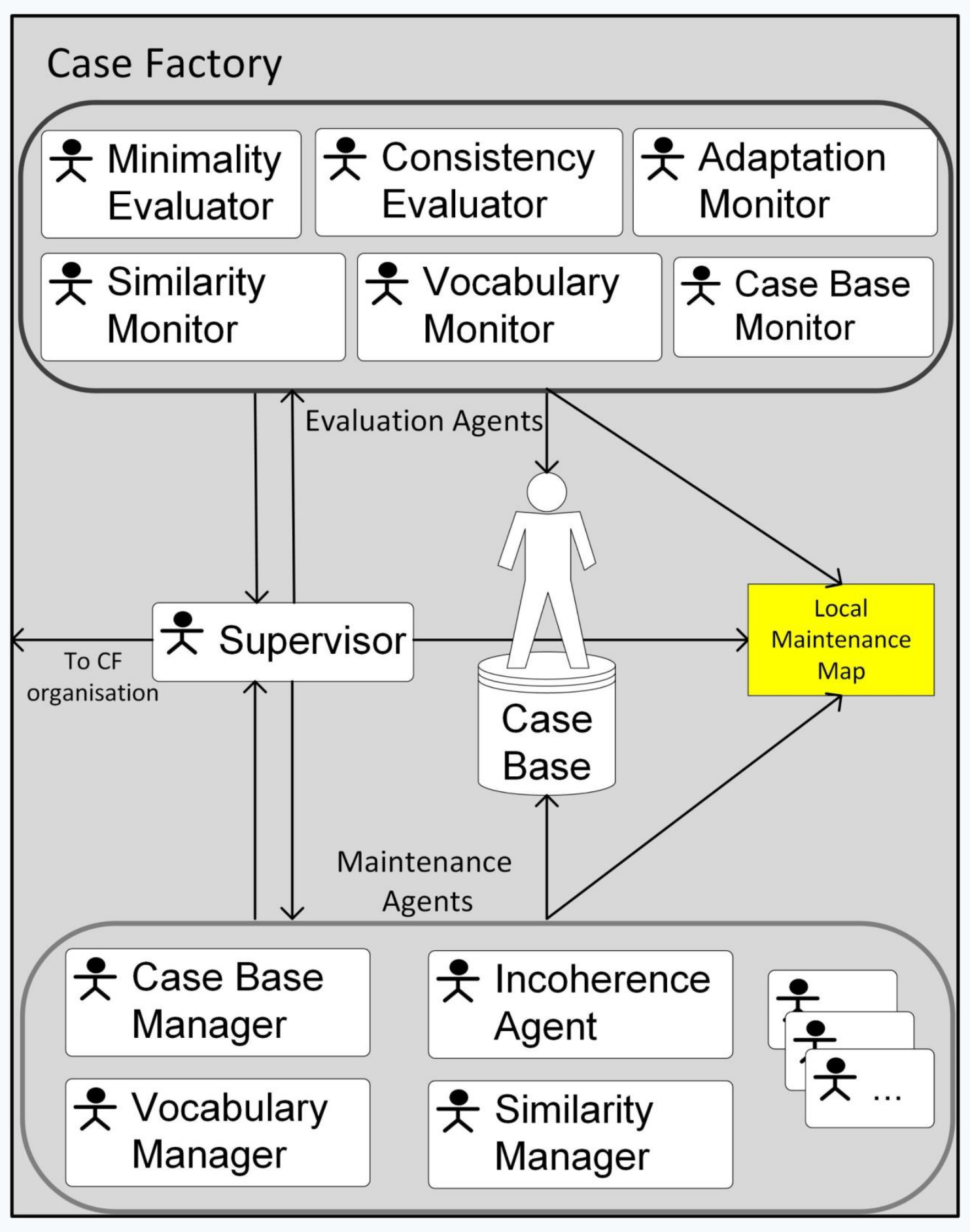
$$\text{where } kle_{source} \text{ and } kle_{target} \in \{hierarchynodes\}$$
$$\text{and } t \in \{u; b\}$$

Case Factory Organization



The extended Case Factory approach extends the SEASALT architecture with a maintenance mechanism for CBR systems. If a topic agent has access to a CBR system, a CF is provided to maintain the CBR system. To coordinate several CFs a so-called Case Factory Organization (CFO) is provided, which consists of several agents to coordinate the overall system maintenance. A Case Factory consists of several agents that are responsible for different tasks: monitoring, evaluation, coordination, and maintenance execution. A monitoring agent will supervise the knowledge containers of a CBR system to notice changes to the knowledge like adding new cases, changing the vocabulary, or deleting cases. Monitoring agents will only notice the fact that changes have occurred and what has been changed. Evaluation agents are responsible for a qualitative evaluation of the consistency, performance, and competence of the CBR system.

Extended Case Factory



- Hierarchy consists of 6 KNOWLEDGE LEVELS
- KNOWLEDGE LEVELS range from CBR systems (KL 0) to specific values for attributes (KL 6)
- Hierarchy defines the GRANULARITY of dependencies
- Hierarchy can be autmoatically generated from an existing knowledge model
- A specific node can be identified by an id code
- The id code consists of the KNOWLEDGE LEVEL, characters, and continuous numbers

Example dependency:

d = (1_V_0_0_1_1, 1_C_1_1_1_1, u)

represents a dependeny between the a **specific value** in an **attribute** of the **vocabulary** in **CBR system A** and a **specific value** in an **attribute** of a **specific case** in a **specific case base** in **CBR system A**.

[1] German Research Center for Artificial Intelligence, Kaiserslautern, Germany
[2] Institute of Computer Science, Intelligent Information Systems Lab, University of Hildesheim, Hildesheim, Germany