Cooperative Artefacts: Assessing Real World Situations with Embedded Technology

Martin Strohbach, Hans-Werner Gellersen, Gerd Kortuem, and Christian Kray

Computing Department, Lancaster University, Bailrigg,
Lancaster LA1 4YR, UK
{strohbach, hwg, kortuem, kray}@comp.lancs.ac.uk

Abstract. Ubiquitous computing is giving rise to applications that interact very closely with activity in the real world, usually involving instrumentation of environments. In contrast, we propose *Cooperative Artefacts* that are able to cooperatively assess their situation in the world, without need for supporting infrastructure in the environment. The Cooperative Artefact concept is based on embedded domain knowledge, perceptual intelligence, and rule-based inference in movable artefacts. We demonstrate the concept with design and implementation of augmented chemical containers that are able to detect and alert potentially hazardous situations concerning their storage.

1 Introduction

Many ubiquitous computing systems and applications rely on knowledge about activity and changes in their physical environment, which they use as context for adaptation of their behaviour. How systems acquire, maintain, and react to models of their changing environment has become one of the central research challenges in the field. Approaches to address this challenge are generally based on instrumentation of locations, user devices, and physical artefacts. Specifically, instrumentation of otherwise non-computational artefacts has an important role, as many applications are directly concerned with artefacts in the real world (e.g. tracking of valuable goods [8, 18, 27]), or otherwise concerned with activity in the real world that can be inferred from observation of artefacts (e.g. tracking of personal artefacts to infer people's activity [17]).

Typically, artefacts are instrumented to support their identification, tracking, and sensing of internal state [18, 24, 27]. Complementary system intelligence such as perception, reasoning and decision-making is allocated in backend infrastructure [1, 6] or user devices [26, 28]. This means, only those tasks that could not be provided as easily by external devices are embedded with the artefacts (e.g. unambiguous identification), whereas all other tasks are allocated to the environment which can generally be assumed to be more resourceful (in terms of energy, CPU power, memory, etc). However, this makes artefacts reliant on supporting infrastructure, and ties applications to instrumented environments.

In this paper, we introduce an architecture and system for *Cooperative Artefacts*. The aim is to facilitate applications in which artefacts cooperatively assess their situation in the world, without requirement for supporting infrastructure. Cooperative artefacts model their situation on the basis of domain knowledge, observation of the world, and sharing of knowledge with other artefacts. World knowledge associated with artefacts thus becomes integral with the artefact itself.

We investigate our concept and technological approach in the context of a concrete application domain, chemicals processing, to ensure that it is developed against a real need and under consideration of realistic constraints. We specifically explore how cooperative artefacts can support safety-critical procedures concerning handling and storage of containers with chemical materials. We show that this is an application field in which the ability to detect critical situations irrespective of where these occur is of highest relevance, hence supporting our case for an approach that is not tied to instrumented environments.

Our contribution is twofold. First, preceded by discussion of the application case, we introduce a generic architecture for cooperating artefacts. This architecture defines the structure and behaviour of artefacts in our system model, and serves as model for design of concrete cooperative artefacts. The distinct contribution is that artefacts are enabled to reason about their situation without need for backend services or external databases. Our second contribution, covered in sections 4 to 6, is the development of a prototype system that demonstrates the Cooperative Artefact approach. At the core of this system are chemical containers that are instrumented and configured to cooperatively detect and alert a set of hazardous situations. This addresses a distinct application problem that can not be solved with approaches that rely on instrumented environments.

2 Application Case Study: Handling and Storage of Chemicals

Jointly with the R&D unit of a large petrochemicals company, we have begun to study issues surrounding handling and storage of chemicals in the specific context of a chemicals plant in Hull, UK. Correct handling and storage of chemicals is critical to ensure protection of the environment and safety in the workplace. To guard against potential hazards, manual processes are clearly defined, and staff are trained with the aim to prevent any inappropriate handling or storage of chemicals. However the manual processes are not always foolproof, which can lead to accidents, sometimes of disastrous proportion.

In an initial phase, we have had a number of consultation meetings with domain experts to understand procedures and requirements. Future work will also engage with actual users in the work place, however our initial development work is based on informal problem statements and design proposals that the domain experts formulated for us. We specifically used the following proposal to derive a set of concrete requirements and test scenarios for our technology:

"Alerting against inappropriate materials being stored together or outside of approved storage facilities. *It is not desirable to store materials together with those with which they are particularly reactive. This applies particularly to Peroxides and*

other oxidising agents. Manual processes and training aim to prevent this, but are not always foolproof. It is proposed that materials which are mutually reactive are tagged and that the tags can recognise the close proximity of other "incompatible" materials and hence trigger an alert. The tags should also trigger when the quantity of a material exceeds a limit. A variant of this problem is to alert when dangerous materials e.g. radioactive materials reside outside of approved areas for too long."

From this proposal we have derived a set of potentially hazardous situations that a system must be able to detect and react to, in order to effectively support existing manual processes:

1. Storage of dangerous materials outside an approved area for longer than a pre-defined period of time.
2. Storage of materials in proximity of 'incompatible' materials, in terms of a pre-defined minimum safety distance.
3. Storage of materials with others, together exceeding critical mass in terms of pre-defined maximum quantities.

There are a number of important observations to be made with respect to the identified hazardous situations:

- The identified situations can occur in different environments: at the Chemicals plant, in external storage (e.g. with distributors or customers), or in transit (e.g. when containers are temporarily stored together during transport). Most notably, the environments in which hazardous situations can occur are not under uniform control but involve diverse ownership (e.g. producer, distributors, consumer, logistics). This makes it unrealistic to consider a solution that would depend on instrumentation of the environment with complete and consistent coverage.
- The hazardous situations are defined by a combination of pre-defined domain knowledge (compatibility of materials, safety distances, etc) and real-time observations (detection of other materials, determination of proximity, etc). A generic sensor data collection approach, e.g with wireless sensor networks [2], would not be sufficient to model such situations. It is required that observations are associated with specific domain knowledge.
- The described situations involve a combination of knowledge of the state of individual artefacts, and knowledge about their spatial, temporal, and semantic relationships. As a consequence, detection of situations requires reasoning across all artefacts present in a particular situation. This level reasoning is typically centralized and provided by backend infrastructure. To overcome dependency on backend services, reasoning about artefacts relationships needs to be allocated with the artefacts in a distributed and decentralized fashion.

3 Cooperating Artefacts: Architecture and Components

Figure 1 depicts the architecture we developed for cooperative artefacts. The architecture is comparable to generic agent architectures [13], and independent of any particular implementation platform. However it is anticipated that implementation of

cooperative artefacts will typically be based on low-powered embedded platforms with inherent resource limitations. As shown in figure 1, the architecture comprises the following components:

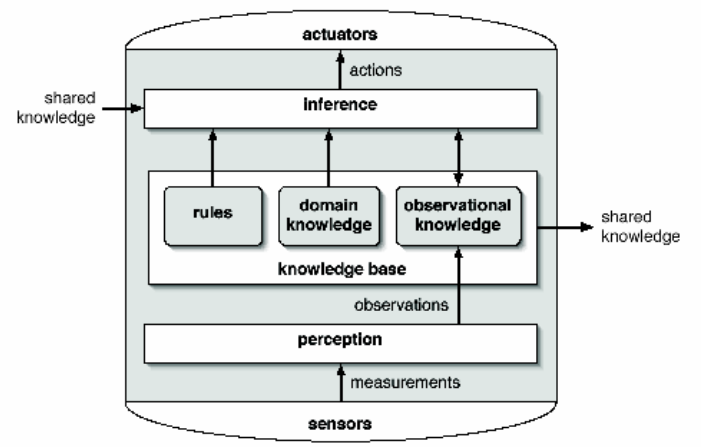

Fig. 1. Architecture of a Cooperative Artefact

- **Sensors.** Cooperative artefacts include sensor devices for observation of phenomena in the physical world. The sensors produce measurements which may be continuous data streams or sensor events.
- **Perception.** The perception component associates sensor data with meaning, producing observations that are meaningful in terms of the application domain.
- **Knowledge base.** The knowledge base contains the domain knowledge of an artefact and dynamic knowledge about its situation in the world. The internal structure of the knowledge base is detailed below.
- **Inference.** The inference component processes the knowledge of an artefact as well as knowledge provided by other artefacts to infer further knowledge, and to infer actions for the artefact to take in the world.
- **Actuators.** Actions that have been inferred are effected by means of actuators attached to the artefact.

3.1 Structure of the Artefact Knowledge Base

It is a defining property of our approach is that world knowledge associated with artefacts is stored and processed within the artefact itself. An artefact's knowledge is structured into facts and into rules. Facts are the foundation for any decision-making and action-taking within the artefact, and rules allow to infer further knowledge based on facts and other rules, ultimately to determine their behaviour in response to their

environment. The type of knowledge and rules managed within an artefact are described in tables 1 and 2.

Table 1. Knowledge stored in a cooperative artefact.

Domain knowledge	Domain knowledge built into the artefact, e.g. facts describing the physical nature of the artefact or general world knowledge.
Observational knowledge	Knowledge describing the situation of an artefact in the world. It is based on facts that result from sensor-based observations.
Inferred knowledge	Knowledge inferred from previously established facts, which may be based on domain knowledge, observation, previous inference, and knowledge made available by cooperating artefacts.

Table 2. Rules of a cooperative artefact.

Inference rules	Rules that describe inference of new facts from previously established facts.
Actuator rules	Rules that describe the facts that must be established in order to trigger an action.

3.2 Cooperation of Artefacts

Artefacts need to cooperate to enable cross-artefact reasoning and collaborative inference of knowledge that artefacts would not be able to acquire individually. Reasoning across artefacts is of particular importance in applications that are concerned with artefact relationships rather than individual artefact state, such as the case study discussed in section 2.

Our model for cooperation is that artefacts share knowledge. More specifically, knowledge stored in an artefact's knowledge base is made available to other artefacts where they feed into the inference process. Effectively, the artefact knowledge bases taken together form a distributed knowledge base on which the inference processes in the individual artefacts can operate. This principle is illustrated in figure 2.

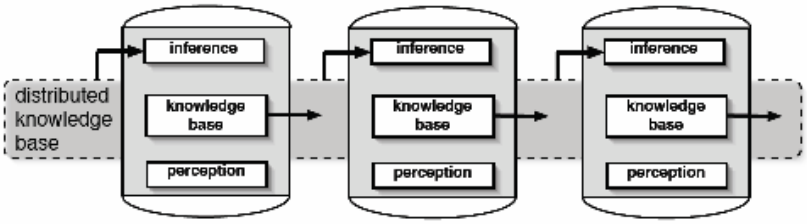

Fig. 2. Cooperation of artefacts is based on sharing of knowledge

For artefact cooperation to be practical and scalable, we require concrete systems to define their scope of cooperation:

- Application scope: artefacts only cooperate with artefacts that operate in the same application or problem domain.
- Spatial scope: artefacts only cooperate with artefacts that are present in the same physical space. The space may be a particular location or defined in relative terms, for example as a range surrounding an artefact.

4 Modelling Chemical Containers as Cooperative Artefacts

In this section, we return to our case study to illustrate how the Cooperative Artefact approach can be applied to a concrete problem domain. In particular, we describe the knowledge embedded in a chemical container that allows them to detect hazardous situations.

The knowledge base of a chemical container contains facts and rules. As representation formalism we use a subset of the logic-programming language Prolog. Thus, all entries of the knowledge base are formulated in Horn logic [12]. Rules and some facts are specified by the developer. Other facts represent observational knowledge derived from observation events in the perception subsystem: `proximity(<container>,<container>)` indicates that two containers are located close to each other; `location(<container>,<in/out>,<time>)` indicates whether a container has been inside or outside of an approved area for a certain amount of time. The sensor systems that enable the derivation of these facts are described in Section 5. Table 3 lists the facts that can be found in the knowledge base, while Table 4 lists rules. In rules, uppercase arguments are variables, while lowercase arguments are constants. The special constant `me` always refers to the artefact that processes the rule.

Table 3. Fact base of a chemical container.

Domain knowledge	`reactive(<chemical>,<chemical>)`
	`content(me,<chemical>)`
	`mass(me,<number>)`
	`critical_mass(<chemical>,<number>)`
	`critical_time(<chemical>,<time>)`

Observational knowledge	`proximity(<container>,<container>)` `location(<container>,<in/out>,<time>)`

Table 4. Rule base of a chemical container.

Inference rules	
(R1) hazard_unapproved:-	content(me, CH), critical_time(CH, T1), location(me, out, T2), T1 < T2.
(R2) hazard_incompatible:-	content(me, CH1), proximity(me, C), content(C, CH2), reactive(CH1, CH2).
(R3) hazard_critical_mass:-	content(me, CH), cond_sum(M1, (proximity(me,C), content(C,CH), mass(C,M1)), S), mass(me, M2), sum(S, M2, SUM) critical_mass(CH, MASS), MASS < SUM.
Actuator rules	
(R4) alert_hazard:- hazard_unapproved	
(R5) alert_hazard:- hazard_incompatible	
(R6) alert_hazard:- hazard_critical_mass	

Rules R1, R2 and R3 define hazards; they are used by the inference engine to evaluate if a hazard can be inferred from the observations.

Rule R1 can be verbalized as follows:

R1: A hazard occurs if a chemical is stored outside an approved area for too long.

This rule is based on three pieces of information: the chemical kept within a container, (modelled by `content(<container>,<chemical>)`), for how long the container has been inside or outside of an approved area (modelled by `location(<container>, <in/out>, <time>)`), and how long the chemical is allowed to be stored outside an approved area (modelled by `critical_time(<chemical>, <time>)`. The `content` and `critical_time` predicates are built-in knowledge that is defined when a container becomes designated for a particular type of chemical. The `location` predicate is an observational knowledge and is added to the knowledge base by the perception mechanism.

Rule 2 can be verbalized as follows:

R2: A hazard occurs if 'incompatible' chemicals are stored too close together.

The second rule, in contrast to the first one, uses distributed knowledge. It takes into account the content of the evaluating artefact (`content(me,CH1)`), the content of a nearby artefact (`content(C,CH2)`), and whether the materials they contain are mutually reactive (`reactive(CH1,CH2)`). The `reactive` predicate captures pre-existing domain knowledge built into the artefacts. The `proximity(<c1>,<c2>)` predicate models the fact that container `c2` is in close proximity to `c1` where spatial proximity is defined in relation to an implicitly defined, built-in safety distance. The proximity fact is an observation that is added to the knowledge base by the perception subsystem.

Rule 3 can be verbalized as follows:

R3: A hazard occurs if the total amount of a chemical substance, stored in a collection of neighbouring containers, exceeds a pre-defined critical mass.

This rule uses a special built-in predicate `cond_sum(OPERAND, CONDITION, SUM)` to build a `SUM` over all instances of `OPERAND` (in this case the mass of chemical content) that satisfy `CONDITION` (in this case being the mass of same material content in nearby containers). Note that `CONDITION` refers to a conjunct of predicates, i.e. all predicates that meet the condition. This means, the variable S in Rule 3 is the sum of the masses of chemicals stored in nearby containers. This sum S is then added to the mass of the evaluating artefacts (using the built-in predicate `sum()`) and compared against the critical limit.

Rules 4 to 6 connect the knowledge base to actuators. They are used by the inference engine to determine whether any hazard exists. These rules have procedural side effects and turn LEDs attached to the containers on and off. More details about the inference process can be found further below in Section 5.

5 Implementation

The facts and rules described in Section 4 define on a logical level how chemical containers perceive their environment, and detect and react to hazardous situations. In this section we discuss a prototype implementation of such a container. In particular, we discuss the sensing, perception, inference and actuation mechanisms.

Our container prototype is a plastic barrel to which an embedded computing device is attached (Figure 3). The device consists of two separate boards that are driven by PIC18F252 micro-controllers. The main functional components of the device are as follows:

- **Sensors.** The device contains two sensors: a range sensor for measuring the distance between containers and an infrared light sensor for detecting if the container is located in an approved area. The range sensor is constructed from an ultrasonic sensor board with 4 transducers, and a sensing protocol that synchronizes measurements between artefacts.
- **Actuators.** The device includes an LED to visually alert users of potential safety hazards.

- **Perception.** The perception component mediates between sensors and knowledge base. It translates ultrasonic distance estimates and IR readings into `proximity` and `location` facts, which are added or modified whenever sensor readings change.
- **Inference Engine.** The inference engine is similar to a simple Prolog interpreter and uses backward-chaining with depth-first search as inference algorithm. Compromises in terms of expressiveness and generality were necessary to facilitate implementation on a micro-controller platform (see below).
- **Communication.** Artefacts are designed to cooperate over a spatial range that is determined by the minimum safety distance specified for storage of chemicals. For communication within this range, artefacts are networked over wireless link. In our concrete implementation we assume that sending range exceeds the safety distance.
- **Knowledge sharing.** A query/reply protocol is implemented over the wireless link to give artefacts access to knowledge of other artefacts.

Figure 3 captures the architecture of the embedded device. It is based on two embedded device modules, both driven by a PIC18F252 microcontroller, and connected over serial line (RS232). On one of the modules is used for sensing and perception of proximity which involves synchronization with other artefacts over a wireless channel, using a BIM2 transceiver, and ultrasonic ranging with 4 transducer arranged for omnidirectional coverage. The other module contains the core of the artefact, i.e. its knowledge base and inference engine. It further contains a BIM3 transceiver to establish a separate wireless link for knowledge queries between artefacts, and a LED as output device.

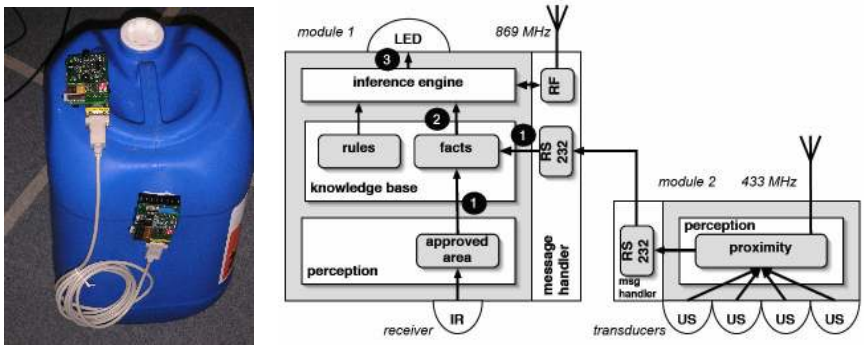

Fig. 3. Physical and architectural view of our augmented chemical container

Inference Process

We have implemented an inference engine with a very small footprint for operation on an embedded device platform with stringent resource limitations. Similar to a Prolog interpreter, the engine operates on rules and facts represented as horn clauses.

The inference engine uses a simplified backward-chaining algorithm to prove a goal, i.e. whether a goal (essentially a query to the knowledge base) can be inferred from the facts and rules in the knowledge base.

The process from perception over inference to actuating is as follows:

Step 1. The perception process transforms sensor readings into an observation which is inserted as a fact into the knowledge base.

Step 2. Whenever there is a change to the knowledge base, the inference engine tries to prove a predefined list of goals. In our chemical container example, the predefined goals are the left sides of rules 1 to 3: `hazard_unapproved`, `hazard_incompatible`, and `hazard_critical_mass`.

Step 3. Depending on the outcome of the inferences in Step 2, actuator rules are triggered. These rules are non-logical rules that have procedural side-effects and control the actuators. In our chemical container example, there is only one actuator which is a LED. It is switched on if at least one of the actuator rules can be triggered.

The inference engine is limited in many respects. For example, backtracking is only possible over local predicates and the number of arguments per predicate is limited to 3. The current implementation fully supports our case study, requiring about 30 % of the 4KB ROM and 80% of the 1.5KB RAM of the PIC18F252 microcontroller for a worst case scenario.

6 Scenario-based Evaluation of Cooperative Chemical Containers

In the following we will demonstrate the capabilities of cooperative chemical containers by describing experiments that we conducted. Our evaluation methodology is scenario-based and involves a testbed and the handling of container prototypes by people. The externally visible behaviour of artefacts is matched against expected outcomes.

Container Testbed

The Cooperative Container Testbed is a scaled-down prototype of a chemical storage facility as it may exist at a chemical processing plant. The testbed is set up in a 16sqm lab space (Figure 4) and consists of

- Cooperative chemical containers as described in Section 5.
- Infrared beacons mounted on cones used for defining approved storage areas
- A set of software tools for remote monitoring of the inference process and communication of augmented containers, and for performance measurement

The purpose of the testbed is to facilitate experimentation with cooperative artefacts in general and chemical containers in particular. Aspects of cooperative artefacts that we are concerned with are correctness, resource consumption, response time,

modifiability and scalability. In the following discussion, however, we limit our attention to correctness.

Fig. 4. Container Testbed

Figure 5 shows the spatial layout of the testbed with various container arrangements. The red area indicates an approved storage area. This means that chemical containers may be stored in this area for an indefinite time. The grey area, in contrast, represents an unapproved storage area. Chemical containers may temporarily be located in this area but must be moved to an approved area after a certain amount of time. Technically, approved storage areas are realized by means of IR beacons that illuminate the approved area. Areas not illuminated by an IR beacon are considered to be non-approved areas (perception and reasoning is still exclusively done within artefacts). IR beacons are mounted on cones and can easily be moved around.

The testbed contains three containers a1, a2, and b. The two containers a1 and a2 are assumed to contain a peroxide, while container b is assumed to be filled with an acid. Acids are incompatible with peroxides. The containers are actually empty, but their knowledge bases contain entries defining their respective content. All containers continuously monitor their environment as described in section 5.

Table 3. Initial fact base

Container a1	**Container a2**	**Container b**
content(me,"peroxide")	content(me,"peroxide")	content(me,"acid")
mass(me, 20)	Mass(me, 20)	mass(me, 40)
Reactive("peroxide", "acid")	reactive("peroxide", "acid")	reactive("peroxide", "acid")
Critical_mass("peroxide", 30)	critical_mass("peroxide", 30)	-
Critical_time("peroxide", 3600)	critical_time("peroxide", 3600)	critical_time("peroxide", 3600)

The fact bases of the containers holds information about the containers themselves, as well as general domain knowledge. The initial fact base of all the containers, as defined by the application developer, is shown in Table 3. It states, among other things, that container a1 contains 20 kg of zinc peroxide, that the critical mass for this peroxide is 30 kg, that zinc peroxide and acids are reactive (and thus may not be

stored at the same location) and that the maximum amount of time container a1 may stored outside an approved storage area is 3600 seconds or 1 hour.

In the following, we will examine a sequence of container arrangements and discuss how the artefacts use the rules in their knowledge to determine whether a safety hazard has occurred.

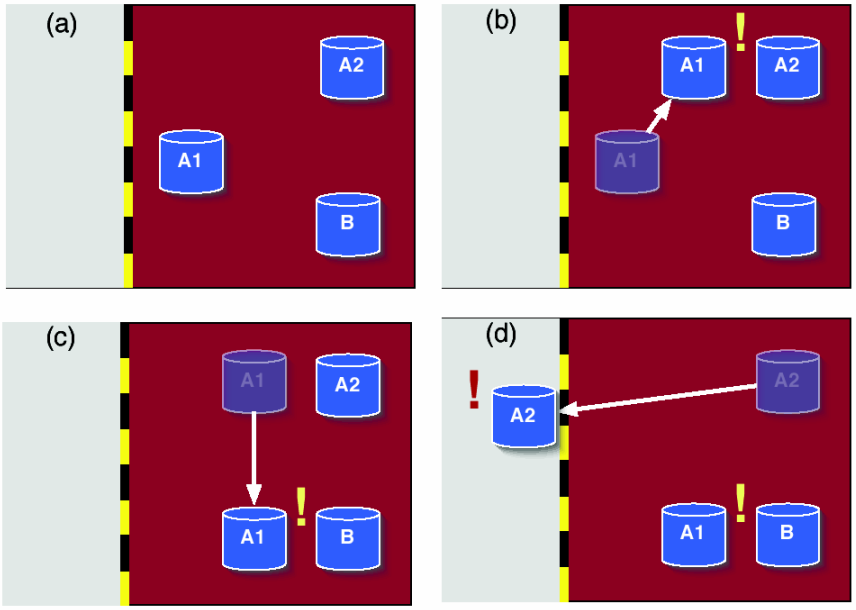

Fig. 5. Example arrangement illustrating different hazards: (a) no hazard, (b) critical mass exceeded, (c) reactive chemicals in proximity, and (d) container stored in a disapproved area too long. The exclamation mark indicates which containers are involved in a hazardous condition.

Scenario 1 (No Hazard)

As soon as the containers are brought into the simulated storage facility, their sensors pick up signals that are translated into facts and added to their knowledge base. Table 4 summarizes the observations of the three containers approximately 1 minute after they are assembled in the arrangement as shown in Figure 4a.

Table 4. Observations in arrangement (a)

Container a1	Container a2	Container b
location(me, in, 35)	location(me, in, 55)	location(me, in, 49)

These observations describe the following situation:

- All containers are currently stored in an approved area. Container a1 has been stored there for at least 35 seconds, container a2 for 55 seconds and container B for 49 seconds..
- The absence of any proximity() fact indicates that containers are not close enough to each other to be detectable by the ultrasound transceivers[1].

In this situation, none of the three hazard conditions can be proven to be true. The goals `hazard_critical_mass` and `hazard_incompatible` fail for all three containers because there is no `proximity` fact in the knowledge base. Goal `hazard_unapproved` fails, because all containers are located in an approved area.

Scenario 2 (Chemical exceeds critical mass)

In Scenario 2 we move container a1 directly next to a2 (Figure 4b.). In this case, both a1 and a2 observe that they are close to one another, and thus `proximity` predicates are added to the knowledge base. Table 5 summarizes the fact bases after the containers have been assembled as shown in arrangement 4b.

Table 5. Observations in arrangement (b)

Container a1	Container a2	Container B
proximity(me, a2)	proximity(me, a1)	-
location(me, in, 70)	location(me, in, 92)	location(me, in, 85)

In this situation, goal `hazard_critical_mass` succeeds. Thus, artefacts a1 and a2 detect – independently of each other – a hazardous situation in which too much of one chemical is stored in one place. In contrast, both `hazard_incompatible` and `hazard_unapproved` fail. During the inference process, a1 and a2 wirelessly send queries to each other to determine each others content and mass.

Scenario 3 (Reactive chemicals stored next to each other)

In Scenario 3, we move container a1 directly next to container b (Figure 4c). As a1 is moved close to b, the proximity facts relating to a1 and a2 are removed and new proximity facts relating to a1 and b are added to the knowledge bases. Table 6 summarizes the fact base of the three containers after they have been assembled in arrangement 4c.

Table 6. Observations in arrangement (c)

Container a1	Container a2	Container B
proximity(me, b)	-	proximity(me, a1)
location(me, in 142)	location(me, in, 154)	location(me, in, 147)

In this situation, goal `hazard_critical_mass` no longer succeeds, thus removing the hazard that previously existed. However, goal `hazard_incompatible` now succeeds, representing a new but different hazard which is detected by simultaneously but independently by containers a1 and b.

[1] Intelligent artefacts make use of the closed world assumption: information contained in a knowledge base is assumed to be complete; facts not stored in the knowledge base are thus false.

Scenario 4 (Container stored in unapproved area for too long)

In Scenario 4, we move container a2 out of the approved area and into the unapproved area (Figure 4d). The `location` fact of container a2 is updated accordingly and now indicates that a2 is located outside an approved area. Table 7 summarizes the fact base of the three containers approximately 30 seconds after they have been assembled in arrangement 4d. The proximity facts of containers a1 and b have not changed.

Table 7. Observations in arrangement (d)

Container a1	Container a2	Container B
proximity(me, b)	-	proximity(me, a1)
location(me, in, 210)	location(me, out, 29)	location(me, in, 215)

In this situation, not much has changed as far as hazards are concerned. As in Situation 3, goal `hazard_incompatible` succeeds, but `hazard_critical_mass and hazard_unapproved` fail. `hazard_incompatible succeeds` because the proximity facts of containers a1 and b have not changed. `hazard_unapproved` fails because the time a2 has spent in an unapproved area (29 seconds) is still too small to trigger a hazard. However, eventually the time indicator of the location fact of container a2 will exceed the maximum permissible time (which is defined in Table 3 as 3600 seconds). At that point in time, `hazard_unapproved` succeeds and a new hazard is detected by container a2. The observations at this time are summarized in Table 8.

Table 8. Observations in arrangement (d) after 1 hour

Container a1	Container a2	Container B
proximity(me, B)	-	proximity(me, a1)
location(me, in , 3810)	location(me, out, 3629)	location(me, in, 3815)

Scenario 5 (Return to safe situation)

In our final scenario, we move the containers back to the original arrangement (Figure 4a). Immediately, proximity facts are removed from the fact base of containers a1 and b. Similarly, the `location` fact of container a2 is updated to indicate that it is again located within an approved area (Table 9).

Table 9. Observations in arrangement (c)

Container a1	Container a2	Container B
location (me, in, 3920)	location(me, in, 20)	location(me, in 3925)

In this situation, just as in Scenario 1, the goals `hazard_incompatible, hazard_critical_mass and hazard_unapproved` fail, indicating that this is again a safe situation.

In sum, we have shown how cooperative chemical containers are able to correctly detect hazardous and non-hazardous situations, even if multiple hazards occur at the same time. This highlights an important aspect of the Cooperative Artefact approach: information gathering and reasoning occur in a decentralized way that enables each artefact to determine the state of the world (i.e. safety) by itself. Consequently, there is no need for an external database or infrastructure.

7 Discussion

The Cooperative Artefacts concept is based on embedding of domain knowledge, perceptual intelligence, and rule-based inference in otherwise non-computational artefacts. The key features of this approach can be summarized as follows: Cooperative artefacts are autonomous entities that actively perceive the world and reason about it; they do no rely on external infrastructure, but are self-sufficient. This enables cooperative artefacts to function across a wide range of (augmented and non-augmented) environments. Collections of co-located artefacts interact to cooperatively assess their situation in the world. Cooperative reasoning enables a system of cooperative artefacts to gain an understanding of the world far beyond the capabilities if each individual artefact. Reasoning occurs in (soft) real-time and is highly context-dependent. This allows cooperative artefacts to be used for time-critical applications. Cooperative artefacts are situated: their ultimate goal is to support human activities in the world. Integration with existing work processes is a key aspect of the design of cooperative artefacts.

Our current implementation of cooperative containers has a number of important shortcomings. Chief among them is the fact that spatial scoping is realized implicitly and that it depends on the capabilities and limitations of the ranging sensors. There is currently no mechanism for explicitly defining the scope of inference rules in a declarative and implementation-independent manner as part of the knowledge base. Furthermore, the complete independence of cooperating artefacts can lead to inconsistent behaviour. For example, it is possible that identical containers interpret the same situation in different ways (for example because of timing issues or slight variations of the sensors readings). Detecting and possibly resolving inconsistencies across a collection of artefacts will become an important issue. Finally, cooperative artefacts have no sense of a global time. This currently prevents to reason about time correlations between observations made by independent artefacts.

A number of questions related to the implementation of cooperative artefacts remain open for future explorations. Among them are: What is the right trade-off between the expressiveness of the representation language and the feasibility of the implementation on an embedded systems platform? Is it necessary to give up completeness of the reasoning algorithms in order to guarantee real-time behaviour (preliminary results indicate that communication is the main limiting factor and not processing)? How can we design the inference engine to minimize energy usage? Although our current implementation provides partial answers, we need to gain a better understanding of requirements and design trade-offs. We thus plan to explore additional application domains and have started further experimentation with the current prototype.

8 Related Work

Our work is generally related to other ubiquitous computing research concerned with instrumentation of the world and with systems that adapt and react to their dynamically changing environment. This includes application-oriented context-aware

systems, that make opportunistic use of information on activity in the world as context for system adaptation and user interaction [9, 25], as well as generic sentient computing infrastructures that collect and provide information on dynamic environments [1]. Most of previously reported systems and infrastructures are based on instrumentation of locations (e.g. office [1, 7, 23], home [6, 15, 22]), or of users and their mobile devices (e.g. [19, 26, 28]).

Previous research has also considered the role of artefacts in addition to locations and users. For instance the Cooltown architecture suggests a digital presence for 'things' as well as people and places, to provide information on artefacts and their relations to users and locations as context [16]. A variety of concrete systems have explored artefacts from different perspectives, for example observation of artefacts to infer information on activity. Examples are tracking of lab equipment to create a record of experiments, as investigated in the Labscape project [4], and tagging of personal artefacts with the goal to create rich activity records of an individual for open-ended uses [17]. More closely related to our work are systems directly concerned with artefacts and their situation, for example for tracking of movable assets and innovative business services [10, 18, 27]. Particularly close in spirit is the eSeal system in which artefacts are instrumented with embedded sensing and perception to autonomously monitor their physical integrity [8].

The actual integration of artefacts in ubiquitous computing systems can involve different degrees of instrumentation. For example, artefacts may be augmented at very low cost with visual tags [24] or RFID tags [18, 30] to support their unique identification and tracking in an appropriately instrumented environment. In contrast, our approach foresees instrumentation of artefacts with sensing, computing, and networking, thus facilitating applications that are fully embedded within artefacts and independent of any infrastructure in the environment. A similar approach underlies the SPEC system that enables artefacts to detect each other and to record mutual sightings independent of the environment [17]. Likewise, Smart-Its Friends are collections of artefacts able to autonomously detect when they are manipulated in the same way [11]. Artefact-based collective assessment of situations has also been illustrated in a system that guides furniture assembly, however with cross-artefact reasoning realized in backend infrastructure [2]. In contrast, Mediacup [5] and eSeal [8] are examples in which artefacts autonomously abstract sensor observations to domain-specific context, using specific heuristics. A more generic framework is provided by the Ubiquitous Chip platform, comprised of embedded sensor/actuator devices whose behaviour is described in terms of ECA (Event, Condition, Action) rules for simple I/O control [29].

In terms of our application case study we are not aware of any similar approaches to detection of potentially hazardous situations in handling of chemical materials. However there is related ubiquitous computing research concerned with assessment of critical situations, such as fire fighting [14], avalanche rescue [21], and guidance through dangerous terrain [20].

9 Conclusion

In this paper we have contributed an architecture for cooperative artefacts, as foundation for applications in which artefacts cooperatively assess their situation in the world. We have demonstrated this approach with implementation of a prototype system in which chemical containers are augmented to detect hazardous situations. There are a number of innovative aspects to be noted:

- It is a novel approach to acquire and maintain knowledge on activity and changes in the world, distinct in being entirely embedded in movable artefacts.
- Embedding of generic reasoning capabilities constitutes a new quality of embedded intelligence not previously demonstrated for otherwise non-computational artefacts.
- The proposed instrumentation of chemicals containers is a novel approach to address to a very significant problem space in handling and storage of chemicals.

The main conclusions that we can draw from our investigation are:

- There is an application need for such approaches to assessment of the state the world, that do not assume infrastructure deployed in the application environment
- The Cooperative Artefact approach meets this need, is technically feasible, and can be implemented efficiently on embedded platforms with limited computational resources.
- The Cooperative Artefact approach has been demonstrated to correctly determine the state of the world on the basis of decentralized information gathering and reasoning, without access to external databases or infrastructure.

References

1. Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A., Hopper, A. Implementing a Sentient Computing System. IEEE Computer 34(5), Aug. 2001, pp. 50-56.
2. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. In Computer Networks, 38(4), March 2002, pp. 393–422.
3. Antifakos, S., Michahelles F., Schiele, B,: Proactive Instructions for Furniture Assembly. Proc. Ubicomp 2002, Gothenburg, Sweden, Sept. 2002.
4. Arnstein, L. F., Grimm, R., Hung, C, Hee, J., LaMarca, A., Sigurdsson, S. B., Su, J., Borriello, G. Systems Support for Ubiquitous Computing: A Case Study of Two Implementations of Labscape, Proc. Pervasive 2002, Zurich, Aug. 2002.
5. Beigl, M., Gellersen H., Schmidt, A. Mediacups: Experience with Design and Use of Computer-Augmented Everyday Artefacts. Computer Networks 35(4), March 2001.
6. Brumitt, B., Meyers, B., Krumm, J., Kern, A. and Shafer, S. EasyLiving: Technologies for Intelligent Environments. Proc. of HUC 2000, Bristol, UK, Sept. 2000.
7. Cooperstock, J.R. Fels, S. S., Buxton, W. and Smith, K.C. Reactive Environments: Throwing Away Your Keyboard and Mouse. Comm of the ACM 40(9), Sept. 1997.

8. Decker, C., Beigl, M., Krohn, A., Robinson, P. and Kubach, U.: eSeal - A System for Enhanced Electronic Assertion of Authenticity and Integrity. In Proc. Of Pervasive 2004, Vienna, Austria, April 2004.
9. Dey, A.K., Salber, D. Abowd, G.D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications In Human-Computer Interaction (HCI) Journal, Vol. 16 (2-4), 2001, pp. 97-166.
10. Fano A., and Gershman A.: The Future of Business Services in the Age of Ubiquitous Computing. In Communications of the ACM, Vol. 45 (12), 2002, pp. 83-87
11. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H-W.: Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts. In Proc. Ubicomp 2001, Atlanta, USA, Sept. 2001.
12. Horn, A.: On sentences which are true of direct unions of algebras. Journal of Symbolic Logic, 16, 14-21, 1951.
13. Jennings, N., Sycara, K. and Wooldridge, M. Autonomous Agents and Multi-Agent Systems, Vol. 1, No. 1, July, 1998, pp. 7 - 38.
14. Jiang, X., Chen, N. Y., Wang, K., Takayama, L., Landay, J. A.: Siren: Context-aware Computing for Firefighting. In Proc. of Pervasive 2004, Vienna, Austria 2004.
15. Kidd, C., Orr, R., Abowd, G., Atkeson, C., Essa, I., MacIntyre, B. Mynatt, E., Starner, T and Newstetter, W.: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In Proc. Cooperative Buildings, CoBuild'99, Pittsburgh, Oct 1999.
16. Kindberg, T., et al.: People, Places, Things: Web Presence for the Real World. In MONET Vol. 7, No. 5, Oct. 2002, Kluwer Publ.
17. Lamming, M., Bohm, D.: SPECs: Another Approach to Human Context and Activity Sensing Research. In Proceedings of Ubicomp 2003. Seattle, WA, USA, October 2003.
18. Lampe M. and Strassner M.: The Potential of RFID for Movable Asset Management. Workshop on Ubiquitous Commerce at Ubicomp 2003, Seattle, October 2003
19. Lukowicz, P. et al.: Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers. Proc. Pervasive 2004, Vienna, Austria 2004.
20. Li, Q., DeRosa, M., Rus, M.: Distributed Algorithms for Guiding Navigation across a Sensor Network. Proc. ACM MobiCom 2003, Sept. 2003, San Diego, CA, USA
21. Michahelles, F. et al.: Applying Wearable Sensors to Avalanche Rescue: First Experiences with a Novel Avalanche Beacon. In Computers & Graphics, Vol. 27, No. 6, 2003.
22. Tapia, E. M., Intille, S. and Larson, K.: Activity Recognition in the Home using Simple and Ubiquitous Sensors. Proc. Pervasive 2004, Vienna, April 2004.
23. Pentland, A.: Smart rooms, *Scientific American*, vol. 274, pp. 54-62, 1996.
24. Rekimoto J. and Ayatsuka, Y.: CyberCode: Designing Augmented Reality Environments with Visual Tags. Proc. Designing Augmented Reality Environments (DARE 2000), 2000.
25. Schilit, B. Adams, N. and Want, R.: Context-aware computing applications. Proc. WMCSA'94.
26. Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., Van de Velde, W.: Advanced Interaction in Context. In Proc. of HUC99, Karlsruhe, Germany, 1999.
27. Siegemund, F. and Flörkemeier, C.: Interaction in Pervasive Computing Settings using Bluetooth-enabled Active Tags and Passive RFID Technology together with Mobile Phones. Proc. IEEE PerCom 2003, March 2003, Fort Worth, USA.
28. Starner, T., Schiele, B. and Pentland, A.: Visual Context awareness in Wearable Computing. Proc. Intl. Symp. on Wearable Computing (ISWC'98), Pittsburgh, Oct. 1998, pp. 50-57.
29. Terada, T., Tsukamoto, M., Hayakawa, K., Yoshihisa, T., Kishino, Y., Kashitani, A. and Nishio, S.: Ubiquitous Chip: a Rule-based I/O Control Device for Ubiquitous Computing. In Proc. of Pervasive 2004, Vienna, April 2004.
30. Want, R., Fishkin, K.O., Gujar, A. and Harrison, B.L.: Bridging Physical and Virtual Worlds with Electronic Tags. Proc. CHI'99.