Towards a Constraint Programming approach for Cognitive IoT

Zakarya Erraji[1], Imade Benelallam[1,2], and El Houssine Bouyakhf[1]

[1]LIMIARF, Faculty of sciences, Mohammed V University in Rabat, Morocco
[2]SI2M Laboratory, INSEA, Rabat, Morocco
{zakarya.erraji.e,imade.benelallam}@ieee.org,
bouyakhf@mtds.com

Abstract. The Internet of Things (IoT) has the potential to impact on how we live and also how we work. Nowadays, the number of devices grows continuously, and the number of their constraints and data evolves exponentially. As a result, the computational complexity of these applications may become NP-Complete. In the last decade, as many approaches of real-time constraint handling have been proposed, Constraint Programming (CP) has been considered to be a stand-alone technology that can be used in several research areas. In this paper, we present a new extension of Cognitive IoT (CIoT) based Constraint Programming frameworks. We introduce a dynamic and distributed Constraint programming platform that covers explicitly several CIOT considerations (e.g. constraint acquisition, constraint reasoning, distributed communication protocols, etc). A real world use case application based on IoT techniques has been modeled, implemented and illustrated using our platform. Therefore, Constraint Programming techniques have been proved to be a very elegant paradigm to handle CIoT applications. The experiment results are promising and meet our expectations to pursue the Holy Grail of computer science.

Keywords: Constraint Programming, Cognitive Internet of Things, Artificial Intelligence.

1 Introduction

Constraint programming has become an important technology for solving hard combinatorial problems. As indicated in figure 1, Constraint Programming has a diverse range of application domains. It has its roots in Artificial Intelligence, operations research, mathematical programming, etc. The solving process in Constraint Programming distinguishes between the phase of modeling the problem and the phase of searching for the solution. Constraint programming is a general framework providing simple, general and efficient models and algorithms for solving real-world and academic problems. The search phase is made by specific algorithms with/without heuristics depending on the type of the problem (i.e. centralized, distributed, satisfaction, optimization, etc).

A constraint network is a formulation of an instance of the constraint satisfaction problem (CSP) which is at the core of constraint programming. This problem or framework has many derivatives [19], and extensions, as indicated in Figure 2: temporal CSP (TCSP), weighted CSP (WCSP), valued CSP (VCSP), quantified CSP (QCSP), constraint optimization problem (COP), Max-CSP, distributed CSP (DisCSP), etc. Usually, a concept or technique introduced for basic CSP has turned out to be relevant to its extensions. For example, the concept of arc consistency has been applied to the most of these extensions.

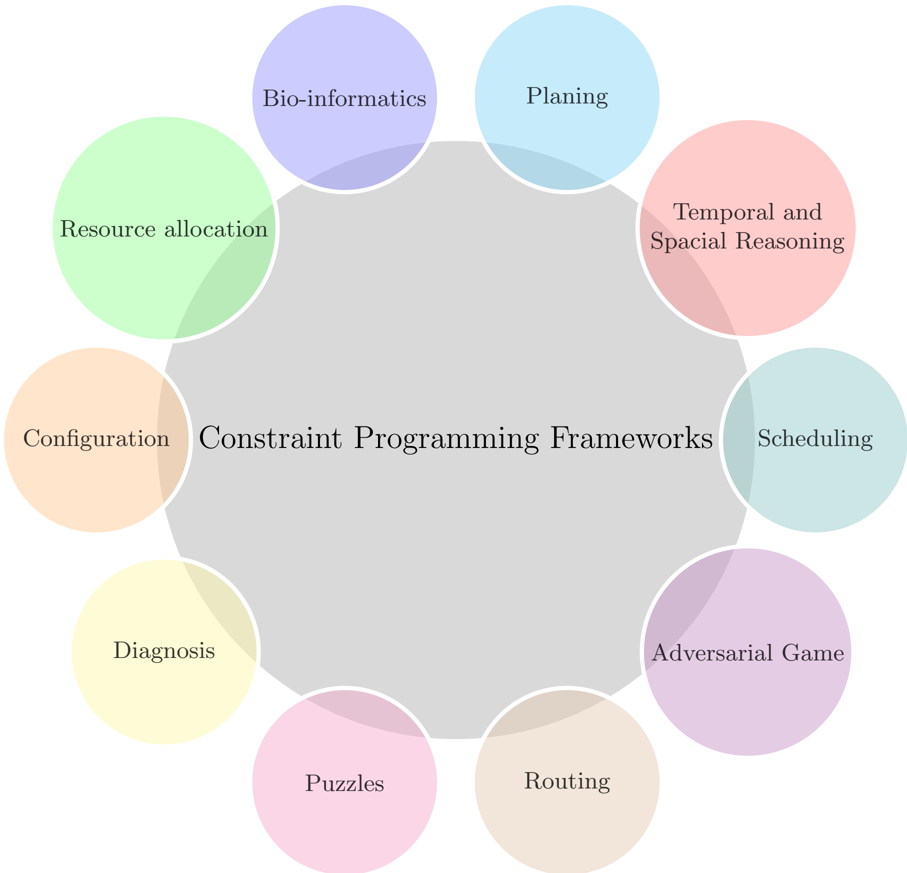

Fig. 1. Some application domains of Constraint Programming frameworks

Many of these frameworks belongs to the Distributed Constraint Reasoning (DCR) formalization. DCS is a framework for modeling and solving various distributed constraint satisfaction/optimization problems arising in Distributed Artificial Intelligence. In DCR, a problem is expressed as a Distributed Constraint

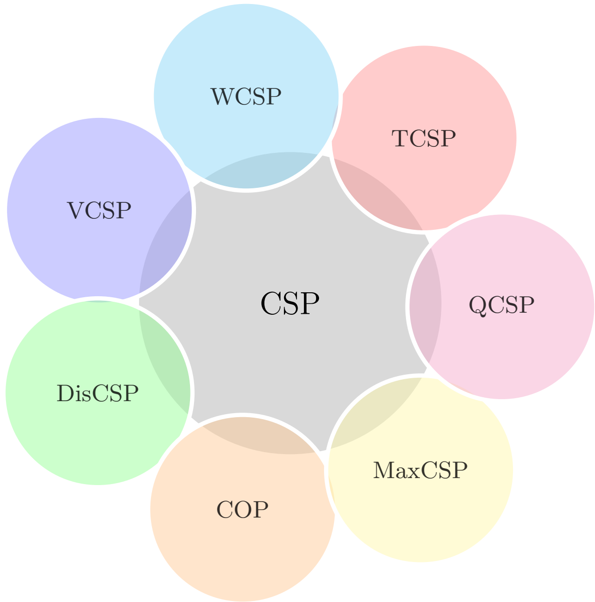

Fig. 2. Constraint Programming frameworks

Network (DCN). A DCN is composed of a group of autonomous agents where each agent has control of some elements of information about the whole problem, namely, variables and constraints (i.e. Each agent own its local constraint network). therefore agents are connected by inter-agent constraints. Agents try to find a local solution (locally consistent assignment) and communicate it with other agents using a DCR protocol to check its consistency against constraints with variables owned by other agents [12, 13]. A DCN offers an elegant way for modeling many real-world and academic problems that are distributed by nature (e.g., distributed resource allocation [14], distributed meeting scheduling [15], sensor networks [16]).

As Connected devices, sensors, and algorithms all operate in ways that involve massive amounts of data, IoT can be considered as a real world Constraint Programming application. So far, The IoT is a recent paradigm where objects are interconnected and equipped with operating systems, sensors and other necessary electronics. The IoT concept aims the enable easy interaction with a wide variety of devices. Several objects can provide different functionality to be combined in a single application. This results in the generation of enormous amounts of data which have to be stored, processed and presented in a seamless, efficient and easily interpretable form. In the literature many surveys have been addressed the Internet of Things paradigm and its promising future. Debasis Bandyopadhyay et al [17] studies the state-of-the-art of IoT and presents the key technological drivers, potential applications, challenges and future research areas in the domain of IoT, where the network discovery mechanisms, the networking, the communication, architecture, and the software and algorithms are a key technologies involved in Internet of Things. Daniele Miorandi et al [9] present a survey of technologies, applications and research challenges for Internet-of-Things. That 'Internet of Things' Thing [10], was the first article that talks about the Internet of Things. The Internet of Things (IoT), firstly coined by Kevin Ashton as the title of a presentation in 1999 [1], is a technological revolution that is bringing us into a new ubiquitous connectivity, computing, and communication era.

Current research on Internet of Things (IoT) mainly focuses on how to enable general objects to see [11], hear, and smell the physical world for themselves, and make them connected to share the observations, but, only connected is not enough. Beyond that, general objects should have the capability to learn, think, and understand both physical and social worlds by themselves. New paradigm, named Cognitive Internet of Things (CIoT) to empower the current IoT with a brain for high level intelligence. Inspired by the effectiveness of human cognition.

Two big issues rise with the IoT: Security and the massive amount of data that all of these devices are going to produce. In the constraint programming research field we found in one hand some algorithms (ABT 1ph and ABT 2ph [5], DisFC [6], etc) proposed to keep confidentiality of constraints and/or values during the solving process and when exchanging messages, and in the other hand some research works (e.g. Opportunities and Challenges for Constraint Programming. [8]) proposed to represent the big data as an opportunity of Constraint Programming.

In this paper we present both Constraint Programming and The Internet of Thing paradigms and some related works. After that, we provide the architecture of our new Constraint Programming based Platform for Cognitive Internet of Things. Then, we illustrate the use of the platform on a real world problem. Finally, we draw conclusions from the results obtained and we give guidelines form future work.

2 Contribution

2.1 The Architecture of Constraint based CIoT platform

Constraint based CIoT platform is an extension of the "Dynamic JChoc" Platform[2]. "Dynamic JChoc" Platform deals with agents with local complex problems and allows a realistic use of agents on a real distributed and dynamic framework. In fact, many approaches can be implemented and tested on this platform [18]. So that, this platform is a distributed constraint multi-Thing system, proposed for IoT applications.

Constraint based CIoT platform is implemented in JAVA and provides classes that implement and inherit from JADE [27] and Choco [4] 4.0 platforms to define the behavior of each connected thing. Figure 3 represents the main Constraint based CIoT platform architectural elements. This platform has these main modules described as bellow:

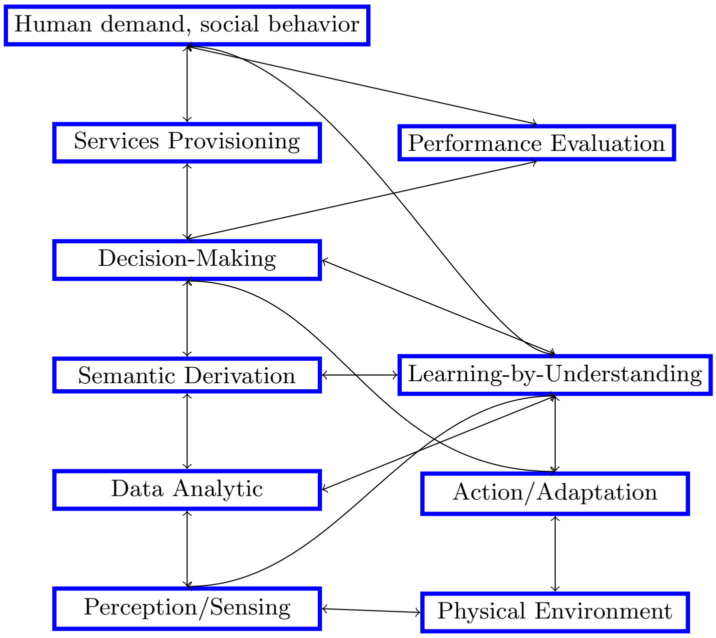

Fig. 3. Constraint based CIoT platform architecture

- **Human demand, social behavior :** The configuration could be defined as a constraint formalisation problem using a xml file;
- **Service Provisioning :** Agent that manages services (e.g. resources as a service) in the platform;
- **Performance Evaluation :** Using Constraint Programming metrics;
- **Decision-Making:** Using Constraint Programming algorithms (e.g. MAC [20], LiveABT [21], AFC-ng [22], etc);
- **Semantic Derivation and Knowledge Discovery:** Pattern discovery and machine learning via CP;
- **Data Analytic:** Constraint-based concept mining;
- **Perception/Sensing:** the values of Agent sensors are represented as unary constraint of equality;
- **Learning-by-Understanding :** Constraint acquisition [24], Nogood messages, etc;

- **Action/Adaptation :** Dynamic CP frameworks; messages (e.g. OK? and CPA, etc);
- **Physical Environment:** Sensors, ROS [23], etc;

The figure 3 show the Constraint based CIoT platform platform architecture, inspired from the work of Qihui Wu et al [11]. In this architecture, there are many consistent elements working on satisfying the users, generally, without any human intervention, but in some cases, when the platform cannot found any solution and the user did not define its preferences, it can ask him to get the needed information. The users define the whole problem and some information about the devices (connected thing) then they save that in their xml files (i.e. input data). The platform uses implemented algorithms to solve problems and to learn from its environment and to take decision about the devices (i.e. how to work, send data, receive data, ask for sources, etc.).
In brief, all these architecture elements are necessary to give the best working performance. The platform uses Constraint programming algorithms those we can improve using the best ones published by the community.

2.2 Simulation

There are many examples we can provide to explain how to use the Constraint based CIoT platform platform. As a matter of fact, IoT encompasses any connected device. Nowadays, we are surrounded with a world submerged of connected devices, for instance, connected watches, smart phones, connected and smart cars, connected fridges and many other useful connected things. In this section, we will present an application from the agriculture domain.

Leslie Lipper et al. [25] say that Climate-smart agriculture (CSA) is an approach for transforming and reorienting agricultural systems to support food security under the new realities of climate change Climate change disrupts food markets, posing population-wide risks to food supply. Threats can be reduced by increasing the adaptive capacity of farmers as well as increasing resilience and resource use efficiency in agricultural production systems.

In this example, we will use some IoT devices to control the soil and plant status. This is for enhancing the whole yield. Those devices communicate and act according to the data given by the sensors. This example is not for giving a complete solution to manage a farm. However, we explain through this example how to use the platform and we show also the benefit of Constraint Programming paradigm.

We consider a field in the form showed in the figure 4. The field contains a water tank (WT) three sensors (i.e. S1, S2, S3) and each sensor measure the soil moisture. Whenever the value given by the sensor is low, the water demand is high. In addition, all these sensors are connected to a water tank that has a max value and a min value (i.e. quantity of water in the tank). The sensors must have the same value in the solution. When the quantity of water in the tank is under the min value, the tank send itself an order to the farm master to fill water.

Listing 1.1. IoT xml file problem sample

```xml
<?xml version="1.0" encoding="UTF-8"?>
<instance>
  <presentation name="Custom" type="DisCSP" benchmark="RandomDisCSP" model="Simple" format="XDisCSP 1.0" />
  <agents nbAgents="4">
    <agent name="S1" id="1" description="sensor measure the soil moisture" />
    <agent name="S2" id="2" description="sensor measure the soil moisture" />
    <agent name="S3" id="3" description="sensor measure the soil moisture" />
    <agent name="WT" id="4" description="The water tank" />
  </agents>
  <domains nbDomains="1">
    <domain name="D1" type="Integer" nbValues="11">40..50</domain>
   <domain name="D2" type="Integer" nbValues="100">1..100</domain>
   <domain name="D3" type="Integer" nbValues="1">10</domain>
   <domain name="D4" type="Integer" nbValues="1">23</domain>
   <domain name="D5" type="Integer" nbValues="1">41</domain>
   <domain name="D6" type="Integer" nbValues="1">50</domain>
  </domains>
  <variables nbVariables="4">
    <variable agent="S1" name="X_S1d" id="1" domain="D2" description="Current level" />
   <variable agent="S1" name="X_S1c" id="2" domain="D3" description="Final level" />
   <variable agent="S1" name="X_S1f" id="3" domain="D1" description="Demand" />

   <variable agent="S2" name="X_S2d" id="1" domain="D2" description="Current level" />
   <variable agent="S2" name="X_S2c" id="2" domain="D4" description="Final level" />
   <variable agent="S2" name="X_S2f" id="3" domain="D1" description="Demand" />

   <variable agent="S3" name="X_S3d" id="1" domain="D2" description="Current level" />
   <variable agent="S3" name="X_S3c" id="2" domain="D5" description="Final level" />
   <variable agent="S3" name="X_S3f" id="3" domain="D1" description="Demand" />

   <variable agent="WT" name="X_Wc" id="1" domain="D2" description="Current level" />
   <variable agent="WT" name="X_WsumDemand" id="2" domain="D6" description="Water Demand" />
  </variables>
  <predicates nbPredicates="2">
    <predicate name="P1">
      <parameters>int X int Y int</parameters>
      <expression>
          <functional>eq(X, Y)</functional>
      </expression>
    </predicate>
    <predicate name="P2">
      <parameters>int X int Y int cte</parameters>
      <expression>
          <functional>geq(sub(X,Y),cte)</functional>
      </expression>
    </predicate>
  </predicates>
  <constraints nbConstraints="2">
    <constraint name="C1" arity="2" scope="X_S1f X_S2f" reference="P1"/>
   <constraint name="C2" arity="2" scope="X_S1f X_S3f" reference="P1"/>
    <constraint name="C3" arity="2" scope="X_S2f X_S3f" reference="P1"/>


    <constraint name="C4" arity="3" scope="X_Wc X_WsumDemand 1" reference="P2"/>
  </constraints>
  <relations nbRelations="0">
  </relations>
  <GuiPresentation type="DisCSP" benchmark="Custom" name="instance2" nbAgents="4" />
</instance>
```

This problem can be expresses as a Distributed Constraints Satisfaction Problem (DisCSP). The DisCSP is described by the five elements (X, D, C, A, ψ) where:

- X = $\{X_{S1d}, X_{S1c}, X_{S1f}, X_{S2d}, X_{S2c}, X_{S2f}, X_{S3d}, X_{S3c}, X_{S3f}, X_{WsumDemand}, X_{Wc}\}$ is the set of variables
- D = {

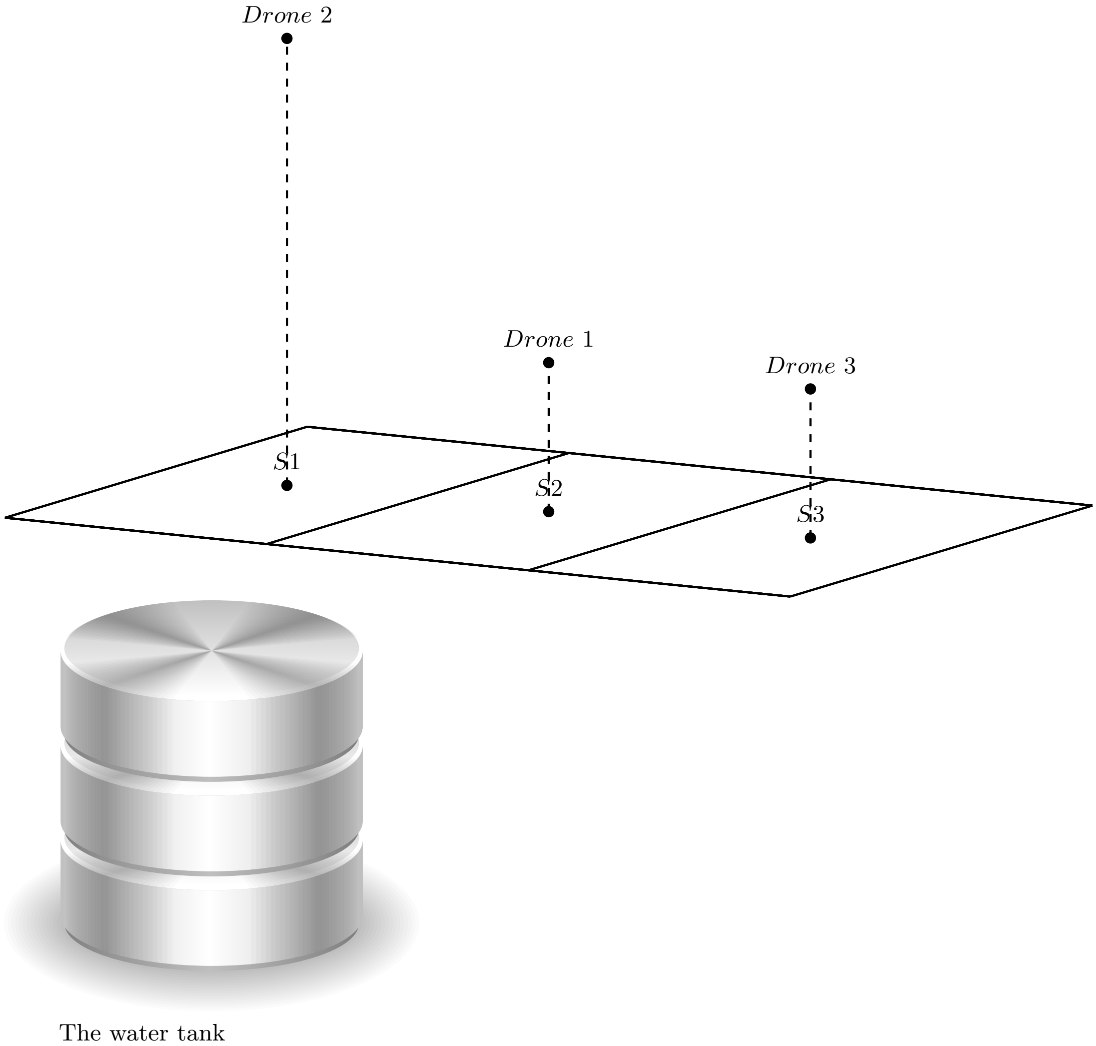

Fig. 4. Field with a water tank and three soil moisture sensors

- $D(X_{Sif}) = \{40, 41, \ldots, 50\}$;
- $D(X_{WsumDemand}) = \{1, 2, 3, \ldots, 100\}$;
- $D(X_{Sid}) = \{1, 2, 3, \ldots, 100\}$;
- $D(X_{S1c}) = \{10\}$;
- $D(X_{S2c}) = \{23\}$;
- $D(X_{S3c}) = \{41\}$;
- $D(X_{Wc}) = \{50\}$;

} is the set of domains
- C = {
  - C1 : All Equal(X_{Sif})
  - C1 : $X_{Wc} - X_{WsumDemand} \geq 1$ where $X_{WsumDemand} = sum(X_{Sid})$

  } is the set of constraints
- A={S1, S2, S3, W} is the set of agent
- ψ: $X_{Si*} \rightarrow$ Si And $X_{Wf} \rightarrow$ W / (Si and W) $\in$ A and $X_{Si*} \in$ X / is a function that maps each variable to its agent

All initial data and values are saved in advance in a XML file presented in the listing 1.1 where we can find the different elements detailed. The given example is a DisCSP problem and we have chosen the ABT algorithm [26] for solving. ABT is a solving protocol that will lead each device/agent to take the correct decision. ABT uses a global priority order to manage agents. The id of each agent (XML file 1.1) can be considered as a global priority order.

The exchanged messages and the final solution as showed in the figure 5. ABT protocol uses 4 types of messages. Ok? message is to inform another agent of taken value, nogood massage is to ask another agent to change his value, adl message is to ask an unlinked agent to make a new link, and stp message is for stopping the resolution.

A second mission can be defined to survey the field with 3 drones. Each drone is considered as an autonomous agent that have to control a part of the field. As shown in the figure 4 the field contains three areas 1, 2 and 3. Let consider a scenario where each drone controls two variables: The altitude position X_{hi} and number of the area to survey represented by X_{ai}. This scenario can be represented by the DisCSP described by the five elements (X, D, C, A, ψ) where:

- X = $\{X_{h1}, X_{h2}, X_{h3}, X_{h3}, X_{a1}, X_{a2}, X_{a3}\}$ is the set of variables
- D = {
  - $D(X_{h1}) = \{1, \ldots, 10\}$;
  - $D(X_{h2}) = \{1, \ldots, 5\}$;
  - $D(X_{h3}) = \{1, \ldots, 10\}$;
  - $D(X_{f1}) = \{1, \ldots, 3\}$;
  - $D(X_{f2}) = \{1, 2\}$;
  - $D(X_{f3}) = \{1, \ldots, 3\}$;

  } is the set of domains
- C = {
  - C1 : AllDiff(X_{ai}) / i = 1,2,3
  - C2 : if $X_{f1} = 2$ then $X_{h1} = 7$

  - C3 : if $X_{f2} = 2$ then $X_{h2} = 7$
  - C4 : if $X_{f3} = 2$ then $X_{h3} = 7$

  } is the set of constraints
- A={D1, D2, D3} is the set of agent (i.e. drones)
- ψ: $X_{fi} \rightarrow$ Di and $X_{di} \rightarrow$ Di / Di $\in$ A and $(X_{Si*}, X_{Si*}) \in$ X / is a function that maps each variable to its agent

In this second example we assume that the drone D2 cannot reach the area number 3 (Domain definition) in the field because of the battery autonomy. And also we have added two constraints, The first is for allowing each drone to control one area and the second is related to the areas 2 that must be surveyed on an altitude equals to 7.

The problem can be defined in the XML like in the listing 1.1, and also, we can use the ABT algorithm to solve it. A solution found by the ABT algorithm is bellow :

- Drone D1 : $X_{h1} = 7$ and $X_{a1} = 2$
- Drone D2 : $X_{h2} = 1$ and $X_{a2} = 1$
- Drone D3 : $X_{h3} = 1$ and $X_{a3} = 3$

As interpretation, the drone D1 surveys the area 2 on an altitude equals to 7, and the drone D2 surveys the area 1 on an altitude equals to 1, and finally the drone D3 surveys the area 3 on an altitude equals to 1.

3 Conclusion

In this paper, we have introduced the use of Constraint Programming paradigm with the Cognitive Internet of Things paradigm. The marriage between these paradigms has been done using a new platform namely Constraint based CIoT platform. As a main result, we showed through a simple example, how can the platform be useful to solve many problems without any human intervention. Our results motivate further work by integrating a learning mechanism inside the platform to let objects lean from their environment.

References

1. Bessire, Christian, Arnold Maestre, Ismel Brito, and Pedro Meseguer. "Asynchronous backtracking without adding links: a new member in the ABT family." Artificial Intelligence 161, no. 1-2 (2005): 7-24.
2. Benelallam, Imade, Zakarya Erraji, Ghizlane EL Khattabi, and El Houssine Bouyakhf. "Dynamic jchoc: A distributed constraints reasoning platform for dynamically changing environments." In International Conference on Agents and Artificial Intelligence, pp. 20-36. Springer International Publishing, 2015.
3. Wahbi, Mohamed, Redouane Ezzahir, Christian Bessiere, and El Houssine Bouyakhf. "Nogood-based asynchronous forward checking algorithms." Constraints 18, no. 3 (2013): 404-433.

4. Charles Prud'homme, Jean-Guillaume Fages and Xavier Lorca. "Choco Documentation" http://www.choco-solver.org (2016)
5. Brito, Ismel, Amnon Meisels, Pedro Meseguer, and Roie Zivan. "Distributed constraint satisfaction with partially known constraints." Constraints 14, no. 2 (2009): 199-234.
6. Brito, Ismel, and Pedro Meseguer. "Distributed forward checking." In CP, pp. 801-806. 2003.
7. Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. "ROS: an open-source Robot Operating System." In ICRA workshop on open source software, vol. 3, no. 3.2, p. 5. 2009.
8. O'Sullivan, Barry. "Opportunities and Challenges for Constraint Programming." In AAAI. 2012.
9. Miorandi, Daniele, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. "Internet of things: Vision, applications and research challenges." Ad Hoc Networks 10, no. 7 (2012): 1497-1516.
10. Ashton, Kevin. "That internet of things thing." RFiD Journal 22, no. 7 (2011).
11. Wu, Qihui, Guoru Ding, Yuhua Xu, Shuo Feng, Zhiyong Du, Jinlong Wang, and Keping Long. "Cognitive internet of things: a new paradigm beyond connection." IEEE Internet of Things Journal 1, no. 2 (2014): 129-143.
12. Yokoo, Makoto, Edmund H. Durfee, Toru Ishida, and Kazuhiro Kuwabara. "The distributed constraint satisfaction problem: Formalization and algorithms." IEEE Transactions on knowledge and data engineering 10, no. 5 (1998): 673-685.
13. Yokoo, Makoto, and Katsutoshi Hirayama. "Algorithms for distributed constraint satisfaction: A review." Autonomous Agents and Multi-Agent Systems 3, no. 2 (2000): 185-207.
14. Petcu, Adrian, and Boi Faltings. "A value ordering heuristic for local search in distributed resource allocation." In International Workshop on Constraint Solving and Constraint Logic Programming, pp. 86-97. Springer, Berlin, Heidelberg, 2004.
15. Wallace, Richard J., and Eugene C. Freuder. "Constraint-based multi-agent meeting scheduling: Effects of agent heterogeneity on performance and privacy loss." (2002): 176-182.
16. Bjar, Ramn, Carmel Domshlak, Csar Fernndez, Carla Gomes, Bhaskar Krishnamachari, Bart Selman, and Magda Valls. "Sensor networks and distributed CSP: communication, computation and complexity." Artificial Intelligence 161, no. 1-2 (2005): 117-147.
17. Bandyopadhyay, Debasis, and Jaydip Sen. "Internet of things: Applications and challenges in technology and standardization." Wireless Personal Communications 58, no. 1 (2011): 49-69.
18. Erraji, Zakarya, Amal Hakkou, Amine Benamrane, Imade Benelallam, and El Houssine Bouyakhf. "A Distributed Constraint Reasoning approach towards intelligent marketplace environment." ModRef, CP (2016).
19. Lecoutre, Christophe. Constraint Networks: Targeting Simplicity for Techniques and Algorithms. John Wiley & Sons, 2013.
20. Sabin, Daniel, and Eugene Freuder. "Contradicting conventional wisdom in constraint satisfaction." In Principles and practice of constraint programming, pp. 10-20. Springer Berlin/Heidelberg, 1994.
21. Benamrane, Amine, Yosra Acodad, El Houssine Bouyakhf, and Imade Benelallam. "LiveABT: A Real-Time Repairing Protocol for Incremental and Dynamic DisCSPs." International Journal of Artificial Intelligence 15, no. 1 (2017): 126-146.

22. Ezzahir, Redouane, Christian Bessiere, Mohamed Wahbi, Imade Benelallam, and El Houssine Bouyakhf. "Asynchronous inter-level forward-checking for DisCSPs." In International Conference on Principles and Practice of Constraint Programming, pp. 304-318. Springer, Berlin, Heidelberg, 2009.
23. Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. "ROS: an open-source Robot Operating System." In ICRA workshop on open source software, vol. 3, no. 3.2, p. 5. 2009.
24. Daoudi, Abderrazak, Younes Mechqrane, Christian Bessiere, Nadjib Lazaar, and El Houssine Bouyakhf. "Constraint Acquisition Using Recommendation Queries." In IJCAI: International Joint Conference on Artificial Intelligence, pp. 720-726. 2016.
25. Lipper, Leslie, Philip Thornton, Bruce M. Campbell, Tobias Baedeker, Ademola Braimoh, Martin Bwalya, Patrick Caron et al. "Climate-smart agriculture for food security." Nature Climate Change 4, no. 12 (2014): 1068-1072.
26. Bessiere, Christian, Arnold MAESTRE, and Pedro MESEGUER. "La famille ABT." Journes nationales sur la rsolution pratique de problmes NP-complets. 2002.
27. Bellifemine, Fabio, Giovanni Caire, Giovanni Rimassa, Agostino Poggi, Tiziana Trucco, Elisabetta Cortese, Filippo Quarta, Giosue Vitaglione, Nicolas Lhuillier, and Jereme Picault. "Java agent development framework." TILAB Italia, http://jade. cselt. it, status 10 (2002): 2002.

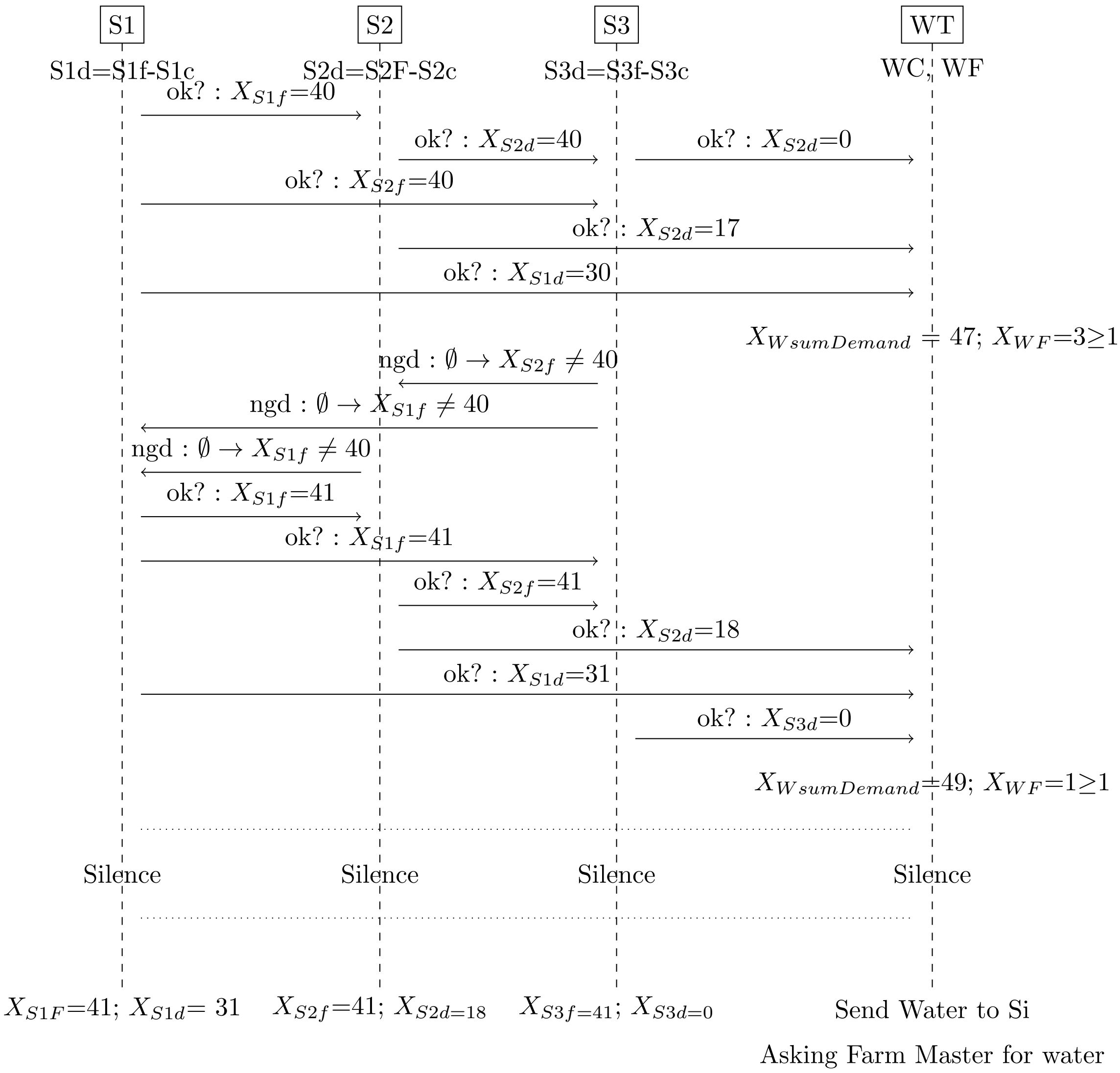

Fig. 5. Solving process with ABT algorithm