Autonomous Aerial Vehicle
Based on Non-Monotonic Logic

José Luis Vilchis Medina[1], Pierre Siegel[1] and Andrei Doncescu[2]
[1]*Aix Marseille Université, CNRS, LIF, Marseille, France*
[2]*LAAS, CNRS, Toulouse, France*

Keywords: Autonomous Motor-glider, Non-monotonic Reasoning, Default Logic, Decision-making, Solar Cells, Artificial Intelligence.

Abstract: In this article we study the case of an autonomous motor-glider. The aims of the aircraft is to maintain its flight as long as possible, taking advantage of the rising air from the ground, known as thermals, despite of limited energy resources and possible external influences, such as turbulences. The pilot task being to make decisions with incomplete, uncertain or even contradictory information, as well as driving to the desired path or destination. We propose the formulation of a model from the point of view of logical theory, using non-monotonic logic and more specifically default logic, to tackle these problems. Finally, we present the results of a simulation for further application in a glider(reduced model) which use solar cells for power management in embedded system.

1 INTRODUCTION

The glider is one of the most relevant aircraft, in terms of the ratio of the distance traveled and the loss of altitude. Making it more efficient to fly than other aircrafts. In this paper, we focus on an autonomous glider(reduced model), a convenient choice(cost) in terms of weight and aerodynamics. The principle of a glider is that it uses the "vertical" wind (thermal and dynamic energy). It is to "climb", which that means to increase in altitude and reach an updraft, while "descent", expresses the rate of descent in a downward burst. "Zero" descent means that updrafts are strong enough to maintain flight, but not enough to allow climbing. As in the nature, there are various species of birds using the same principle. For example: albatross and condor. To perform those flight maneuvers, a pilot must take decisions concerning the flight situation from the cockpit. He has access to different instruments showing altitude, wind speed, inclination of the aircraft, etc. Another constraint is that he also needs to find thermals, respecting the aeronautical and security regulations. Taking these considerations into account, the pilot must follow his desired flight path applying control commands. Increasing or decreasing altitude, turning to the right or left, etc. These rules are applicable to many types of aircraft. We introduce a discrete model of flight rules. This method is based on non-monotonic logic. There are different research proposals in non-monotonic logic reasoning: default reasoning, autoepistemic reasoning, reasoning in the presence of contradictory information and negative reasoning (El-Azhary et al., 2002). On another side, we can consider our model as a resilient system. These systems have the ability to resist and adapt from disturbances. They also are highly adaptive, having the capability to merge information, make decisions, interact with multiple agents and have a memory to facilitate learning (Goerger et al., 2014; Chandra, 2010). The main objective is to present a system based on flight rules capable to choose actions with incomplete information. The case study is presented in section 2. In section 3 classical logic representation is described. Section 4 is dedicated to the theoretical definition of default logic and its properties. Section 5 contains an explanation of the logical system using default logic representation and finally, section 6 practical case is described.

2 RELATED WORK

Research shows (Toulgoat et al., 2011; Toulgoat, 2011; Doncescu and Siegel, 2015; Le et al., 2013) that prove decision-making by non-monotonic logic is encouraging in the field of artificial intelligence. A

Medina, J., Siegel, P. and Doncescu, A.
Autonomous Aerial Vehicle - Based on Non-Monotonic Logic.
DOI: 10.5220/0006304002360241
In *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2017)*, pages 236-241
ISBN: 978-989-758-242-4

pilot is a person who has the function of guiding an aircraft in flight. Generally, the main cockpit flight control are: control yoke, rudder pedals and engine speed. Aircraft cockpit provides necessary information to control the trajectory and operation of the aircraft (de l'Aviation Civile., 1992).

2.1 Flight Indicators

On-board aircraft the basic set of instruments, also called "six pack"(de l'Aviation Civile., 1992). These six instruments are standard and many cockpits basic indicators are:

- *Altimeter*: shows the aircraft's altitude (in feet) above sea-level. As the aircraft ascends, the altimeter to indicate a higher altitude and vice versa.
- *Airspeed indicator*: shows the aircraft's speed (in knots) relative to the surrounding air. It works by measuring the ram-air pressure in the aircraft's Pitot tube relative to the ambient static pressure.
- *Vertical speed indicator*: sometimes called a variometer or also rate of climb indicator, senses changing air pressure, and displays that information to the pilot as a rate of climb or descent in feet per minute or meters per second.
- *Attitude indicator*: also known as an *artificial horizon*. Shows the aircraft's relation to the horizon. From this the pilot can tell whether the wings are level (*roll*, Fig. 1) and if the aircraft nose is pointing above or below the horizon (*pitch*, Fig. 1).
- *Turn indicator*: This instrument includes the Turn-and-Slip Indicator and the Turn Coordinator, which indicate rotation about the longitudinal axis.
- *Heading indicator*: displays the aircraft's heading with respect to magnetic north when set with a compass.

2.2 Flight Control

One time the pilot knows the flight situation through on-board instruments, he must take decisions and apply control commands(actions). Thereby accurately correcting the trajectory. For flight control, the pilot has 3 types of controllers. Generally, these controllers are in much type of aircrafts (de l'Aviation Civile., 1992). Description of controllers for our case, are described below:

- *Yoke*: has the function of controlling in both pitch and roll (Fig. 1).

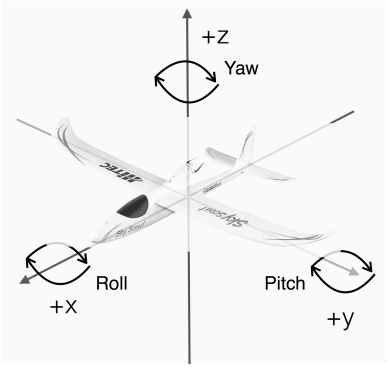

Figure 1: Definition of aircraft body axes.

1. *Pitch controller*: When the yoke is pulled back the nose of aircraft rises and when it is pulled forward the nose descend.
2. *Roll controller*: When the yoke is turned right the aircraft rolls to the right(turning right) and when it is turned left the aircraft rolls to the left(turning left).

- *Rudder*: also *pedal direction*. When pilot push left pedal, rudder deflects to the left and when pilot push right pedal, rudder deflects to the right. With rudder, pilot can not use it to turn left or turn right. Rudder is important for flight stabilization.
- *Engine power*: When pilot turn on the engine, it provides propulsion to the aircraft, if necessary.

Aircraft control systems are based on integro-differential equations or state-space form (Fossen, 2011). We present another method/solution tackled from the perspective of Artificial Intelligence. Next, we introduce how we can describe the issue by using logical representation.

3 CLASSICAL LOGIC REPRESENTATION

To describe actions, we use classical logic language L (Propositional or First-Order Logic). In L, we can represent, for example, $motor(t_i)$ to say that motor is active at the time t_i. Or $altitude(low,t_i)$ to say that the altitude is decreasing at the time t_i. We are in a logical framework, so it is possible to represent almost everything we want in a natural way. Classical logic, such as First-Order Logic is monotonous. What it means, $A \vdash w$ then $A \cup B \vdash w$. This is, by adding new information or set of formulas to a model, the set of consequences of this model is not reduced. The property of monotony is very important in the world of mathematics, because it allows to describe lemmas previously demonstrated. But this property cannot be

applied to uncertain and incomplete information. But in real world, non-monotonic is presents in many situations, this is, each new information we learn can modify or invalidate previous deductions. A classic example is:"Birds typically fly". This expression cannot be translated by the formula $bird(X) \rightarrow fly(X)$, which will signify that all birds fly. But, there are exceptions to this rule (i.e. if $X = Penguin$, by definition a penguin is a bird but he not fly). These exceptions are well known in Artificial Intelligence. In the next table, we represent the actions that pilot can do to correct the trajectory of the glider. We consider a discrete system with three possible actions for each of the controllers, except for the motor(which it has two states), resulting 11 states.

Table 1: Possible actions.

Roll:	*Pitch:*
$yoke(left,t_i)$ $yoke\_roll(neutral,t_i)$ $yoke(right,t_i)$	$yoke(pull,t_i)$ $yoke\_pitch(neutral,t_i)$ $yoke(push,t_i)$
Yaw:	*Propulsion:*
$rudder(left,t_i)$ $rudder(neutral,t_i)$ $rudder(right,t_i)$	$motor(t_i)$ $non\_motor(t_i)$

For the first approach, we will explicitly give basic pilot rules. For example, we might say, "the yoke is to the right position at the time t". We can represent in classical logic: $yoke(left,t_i)$. We use this notation: *var* = variometer, *ther*= thermal, *alt* = altitude, *non_ther* = non thermal, *srh_th* = searching thermal. A set of rules is explicitly given:

$$var(up,t_i) \wedge non\_motor(t_i) \rightarrow ther(t_{i+1}) \quad (1)$$

$$batt(low,t_i) \wedge alt(low,t_i) \rightarrow land(t_{i+1}) \quad (2)$$

$$non\_ther(t_i) \wedge alt(low,t_i) \rightarrow land(t_{i+1}) \quad (3)$$

$$non\_ther(t_i) \wedge alt(low,t_i) \rightarrow srh\_th(t_{i+1}) \quad (4)$$

$$non\_motor(t_i) \wedge var(up,t_i) \wedge turn(left,t_i) \rightarrow yoke(right,t_{i+1}) \quad (5)$$

$$non\_motor(t_i) \wedge var(up,t_i) \wedge turn(right,t_i) \rightarrow yoke(left,t_{i+1}) \quad (6)$$

$$var(up,t_i) \wedge non\_motor(t_i) \rightarrow alt(up,t_{i+1}) \quad (7)$$

$$batt(low,t_i) \wedge non\_motor(t_i) \rightarrow alt(down,t_{i+1}) \quad (8)$$

$$motor(t_i) \wedge yoke(pull,t_{i+1}) \rightarrow alt(up,t_{i+1}) \quad (9)$$

Intuitively, we can say,"the variometer is increasing at the time t", in logical representation is: $var(up,t_i)$. It is of course possible to add others rules(Aeronautics Legislation) to take into account relations between the real scenario and our logical system. These logical formulations before presented, are problematic because there is a conflict. If for example we have a set of formulas: $F = \{var(up,t_i), non\_motor(t_i), batt(low,t_i)\}$, and now, we infer (rule 7 and 8) F: $alt(up,t_{i+1})$ and $alt(down,t_{i+1})$, which is a contradiction. To solve this conflict, we use non-monotonic logic, more specifically, the default logic.

3.1 Definition of Non-monotonic Logic

It is a family of formal logic devised to capture and represent inference, reserving the right to retract deductions when new information is added. The first works in the field of non-monotonic logics began with the realization to precise characterization of defeasible reasoning. Among the pioneers of the field in the late 1970's were J. McCarthy (McCarthy, 1980; McCarthy, 1986), D. McDermott and J. Doyle, and R. Reiter(Reiter, 1980). Deriving in different non-monotonic logics(Sombé, 1990; Suchenek, 2006; Delgrande and Schaub, 2005; Delgrande and Schaub, 2003) reasoning: default reasoning(Reiter, 1980), autoepistemic reasoning, reasoning in the presence of contradictory information, counterfactual reasoning, priority reasoning, and negative reasoning(El-Azhary et al., 2002).

3.2 Definition of Default Logic

A default theory T consists of a set of facts W, which are formulas of First-Order logic and a set of defaults D, which are rules of inference (Reiter, 1980; Grigoris, 1999; Lukaszewicz, 1988; Lifschitz, 1999). The main representational tool is that of a default rule, or simply a default. A default is an inference rule of the form: $\frac{A(X) : B(X)}{C(X)}$, where $A(X), B(X), C(X)$ are well-formed formulas (First-Order Logic). Where $X = (x_1, x_2, x_3, ..., x_n)$ as a vector of free variables(non-quantified). $A(X)$ are the prerequisites, $B(X)$ are the justifications and $C(X)$ are the consequent. Intuitively, a default means:"if $A(X)$ is true, and there is no evidence that $B(X)$ might be false, then $C(X)$ can be true". Default rules are used to create extensions. These can be contemplated as a set of inferences. It is named normal default, if $B(X) = C(X)$.

3.3 Definition of Extension

When defaults are calculated, the number of formulas inferred in the knowledge base W increase. An extension of the default theory $\Delta = (D,W)$ is a set E

of logical formulas(Reiter, 1980). An extension must to verify following property: If d is a default of D, whose the prerequisite is in E, without the negation of its justification is not in E, then the consequent of d is in E. Formally, E is an extension of Δ if and only if:

- $E = \cup_{i=0}^{\infty} E_i$ with:
- $E_0 = W$ and for $i \geq 0$
  $E_{i+1} = Th(E_i) \cup \{C(X) | \frac{A(X) : B(X)}{C(X)} \in D, A(X) \in E_i \text{ and } \neg B(X) \notin E\}$

where $Th(E_i)$ is the set of formulas derived from E_i. The previous definition is difficult to apply in practice. Because $\neg B \notin E$ supposes E is known, but E is not yet calculated. In the case of normal defaults($B(X) = C(X)$), an extension is defined: E is an extension of Δ if and only if:

- $E = \cup_{i=0}^{\infty} E_i$ with:
- $E_0 = W$ and for $i \geq 0$
  $E_{i+1} = Th(E_i) \cup \{C(X) | \frac{A(X) : C(X)}{C(X)} \in D, A(X) \in E_i \text{ and } \neg C(X) \notin E_i\}$

where $Th(E_i)$ is the set of formulas derived from E_i. According to Reiter(Reiter, 1980), if all defaults are normal, it exists at least one extension. Extensions are defined by a fixed point.

4 DEFAULT LOGIC REPRESENTATION

Flight rules are described by a set of rules. For example, $alt(up)$ is a positive literal, means that the glider increases in altitude. The dynamic of the system can be described by $var(up,t)$, which means that the variometer at the time t is increasing. Clauses are the simplest type of formula. Formally, a clause is a disjunction of literals $l_1 \vee l_2 \vee l_3 \vee ... \vee l_n$. A Horn clause is a clause with a maximum of one positive literal. It's a formula defined as $(f_1 \wedge f_2 \wedge f_3 \wedge ... \wedge f_i) \rightarrow g$, where f_i and g are positive literal. Similarly, the formula defined as $\neg(f_1 \wedge f_2 \wedge f_3 \wedge ... \wedge f_i)$ is equivalent to the negative Horn clause(literals can not be true simultaneously). In default logic, we have two sets of rules. The set D of defaults and the set W of facts. For example, set W: $batt(good)$ that is an elementary fact says that battery has sufficient power. And set D: $\frac{batt(good) : motor(on)}{motor(on)}$, this default rule means, "Generally when battery has sufficient power, motor is on". Therefore, we introduce normal defaults representation.

4.1 Set of Default [D]

The set of inference rules D describes possibles actions, including incomplete and contradictory information. It represents a normal default. By using default logic, the rules(7, 8 and 9 in section 3) above could be expressed intuitively as:

7' If $var(up,t_i), non\_motor(t_i)$ are true, and if $alt(up,t_{i+1})$ is not contradictory, then $alt(up,t_{i+1})$ is true.

8' If $batt(low,t_i), non\_motor(t_i)$ are true, and if $alt(down,t_{i+1})$ is not contradictory, then $alt(down,t_{i+1})$ is true.

9' If $motor(t_i), yoke(pull,t_{i+1})$ are true, and if $alt(up,t_{i+1})$ is not contradictory, then $alt(up,t_{i+1})$ is true.

Taking the rules(7',8' and 9'), default logic representation is presented, $d = \frac{A(X) : C(X)}{C(X)}$.

$$d_1 = \frac{var(up,t_i) \wedge non\_motor(t_i) : alt(up,t_{i+1})}{alt(up,t_{i+1})}$$

$$d_2 = \frac{batt(low,t_i) \wedge non\_motor(t_i) : alt(down,t_{i+1})}{alt(down,t_{i+1})}$$

$$d_3 = \frac{motor(t_i) \wedge yoke(pull,t_{i+1}) : alt(up,t_{i+1})}{alt(up,t_{i+1})}$$

4.2 Set of Default [W]

The set of facts W represents accurate and non-revisable information. The facts are formulas always true. Unary clauses are elementary source of information. We can use such a clause to represent a mutual exclusions. For example, taking the rule "The motor-glider can not search a thermal and land at the same time", $\forall t, \neg(glider(search\_ther,t_i) \wedge glider(land,t_i))$.

4.3 Extensions Calculation

To illustrate the use of the default logic, we give in the next paragraph an example of calculation by using the Algorithm 1. $W = motor(off,t), alt(down,t), var(down,t), batt(good,t)$. With: $\forall t, \neg(glider(land,t+1) \wedge glider(srh\_th,t+1))$. Intuitively, W means that glider has not using motor(as propulsion), its altimeter is decreasing, its variometer is decreasing and its battery has sufficient power at time t_i, respectively.

The Algorithm 1 presented below is written in Prolog. Our algorithm calculates the extensions. As the clauses are Horn clauses, and as the defaults are normal. The results obtained using a set D(subsection 4.1) and a set W(subsection 4.2)

are: $E_0 = \{glider(srh\_th,t), motor(off,t), alt(down,t), var(down,t), batt(good,t)\}$

$$d_0 = \frac{alt(down,t) \wedge var(down,t) : glider(srh\_th,t+1)}{glider(srh\_th,t+1)} \quad (10)$$

$E_1 = \{glider(land,t), motor(off,t), alt(down,t), var(down,t), batt(good,t)\}$

$$d_1 = \frac{alt(down,t) \wedge var(down,t) : glider(land,t+1)}{glider(land,t+1)} \quad (11)$$

We can see that W uses a rule that says $glider(srh\_th,t+1)$ and $glider(land,t+1)$ are mutually exclusive. According to this rule, it is not possible to use d_0 and d_1 at the same time. If d_0 is used, $glider(land,t+1)$ is added to the extension, then mutual exclusive will cancel d_1. In the same way, if d_1 is used we cannot use d_0. We obtain 2 contradictory extensions(solutions). In the first one there is $glider(srh\_th,t+1)$(The glider can search a thermal at time $t+1$) and the other one there is $glider(land,t+1)$(The glider can land at time $t+1$). In this way, the conflict shown in section 3 is resolved.

Data: $E = \emptyset$ (Set of extensions E is empty)
Result: $E = \cup_{i=0}^{N} E_i$
Initialization;
CalculExtension(E);
while *there is a default* $(A(X) : C(X))/C(X)$ *that has not yet been inspected* **do**
  Select the default D;
  Verify $A(X)$ are true;
  Verify $C(X)$ is consistent with W;
  Add $C(X)$ to W;
end
Backtracking(deleting $C(X)$ added to W);
CalculExtension(E);
Algorithm 1: Calculation of extensions.

In practice, the choose of extension corresponds a weight to each default considering its importance. An example could be, to search a thermal is more priority (more weight) than land(Toulgoat, 2011; Grigoris, 1999). These decisions are based on weighted product model: $P(A_K/A_L) = \prod_{j=1}^{n}(a_{Kj}/a_{Lj})^{w_j}$ for $K,L = 1,2,3,...,m$. Using this method we ponder extensions then we select the best response.

5 APPLICATION

In this section we present the practical application and current work. In the Fig. 2 shows the different parts of our discrete model. On the left side in the

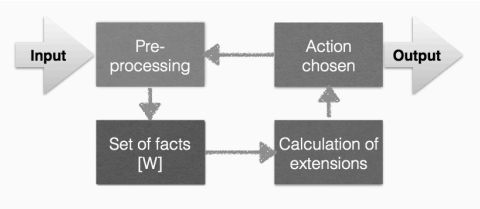

Figure 2: Operating diagram.

Fig. 2, we have the *Input* such as data, which accessible from electronic devices. Different data are airspeed, altitude, angles of pitch and roll, etc. Block *Set of facts[W]* accepts data for knowledge representation. Next, *Calculation of extensions* and *Action chosen* blocks are the representation of our Algorithm 1. Finally, we have *Output* such as an action to apply. Certainly to the servo-motors to control pitch, roll and eventually yaw(Fig. 1). These actions will be applied to a glider(reduced model,Sky Scout version R2GO). Our glider is equipped with a motor, in case he needs desperate increase his altitude or find a rising air. On the other hand, the motor-glider is equipped with an on-board computer. Prolog has been installed and sensors have been successfully implemented. The Algorithm 2 is presented which represents the blocks of *Pre-processing* and *Set of facts[W]*, described in Fig. 2. T_s is sampling time in seconds.

Data: From electronic device $[I]$
Result: Set of facts $[W]$
Initialization of electronics devices;
while T_s*=1* **do**
  CaptureData(I) from electronics devices;
  DescriptionFacts(W) by following our syntax;
end
Algorithm 2: Knowledge representation.

Data are from the Inertial Measurement Unit(IMU). An electronics device which collects angular velocity and linear acceleration data. Usually a combination of accelerometers and gyroscopes, sometimes also magnetometers. An IMU works by detecting changes in rotational attributes like pitch, roll and yaw, using gyroscopes. On the other hand, signal processing techniques(Fig. 2) has tremendous potential in the information fusion theory and in practical applications. Important part because is the process of integration of multiple data for knowledge representation. The acquisition of knowledge is defined by rules(subsection 4.3). In Fig. 3, rotation on the X axis is represented, the combination of accelerometer(circle) and gyroscope(line) data, the result is a signal(triangle) that we use to describe the world.

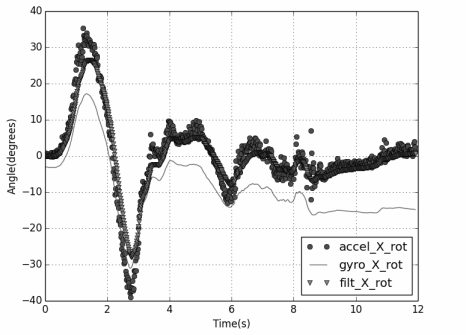

Figure 3: Data fusion.

6 CONCLUSIONS

In our approach, an algorithm based on default logic is proposed to tackle contradictory information. Flight rules are uncertain, because many situations include contradictions and we have to verify them. Furthermore, aeronautics legislation is based on contradictory rules. Non-monotonic logic has a great advantage in these kind of cases. We presented a simulation of a glider searching thermals. The results provide the basis for an autonomous motor-glider. We presented short and simple Prolog programs. For initial tests(around 100 rules), we calculate all extensions in a short time. The calculation takes 1956600 Logical Inferences Per Second (LIPS) and 0.049 seconds of central processing unit (CPU) time on a MacBook. Discrete model is currently being developed for testing. Sensors and on-board computer set-up is complete. We configured an embedded system, installing prolog (SWI-Prolog). We also have installed an embedded sensor with 10 degrees of freedom (DOF). To capture data, such as altitude, acceleration, pitch, roll, yaw, etc. for knowledge representation. More rules are currently incorporated to take into account the aeronautics legislation and solar power management for on-board systems.

ACKNOWLEDGEMENTS

We thanks to the Secretariat of Energy (SENER) through the Mexican National Council for Science and Technology (CONACYT)[grant number 581317/412566] for their support.

REFERENCES

Chandra, A. (2010). Synergy between biology and systems resilience. *Scholars' Mine*.

de l'Aviation Civile., D. G. (1992). *Manuel du pilote d'avion Vol à vue.*

Delgrande, J. and Schaub, T. (2003). On the relation between reiter's default logic and its (major) variants. *Seventh European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*.

Delgrande, J. and Schaub, T. (2005). Expressing default logic variants in default logic. *Journal of Logic and Computation*.

Doncescu, A. and Siegel, P. (2015). *DNA Double-Strand Break-Based Nonmonotonic Logic.* Computational Biology, Bioinformatics and Systems Biology. Elservier.

El-Azhary, Edrees, A., and Rafea, A. (2002). Diagnostic expert system using non-monotonic reasoning. *Expert Systems with Applications*, pages 137–144.

Fossen, T. (2011). Mathematical models for control of aircraft and satellites. *Department of Engineering Cybernetics, NTNU*.

Goerger, S., Madni, A., and Eslinger, O. (2014). Engineered resilient systems: A dod perspective. *Procedia Computer Science*, pages 865–872.

Grigoris, A. (1999). A tutorial on default logics. *ACM Computing Surveys*.

Le, T., Doncescu, A., and Siegel, P. (2013). Utilization of default logic for analyzing a metabolic system in discrete time. *ICCSA*.

Lifschitz, V. (1999). Success of default logic. *Logical Foundations for Cognitive Agents: Contributions in Honor of Ray Reiter*.

Lukaszewicz, W. (1988). Consideration on default logic: an alternative approach. *Comput. Intell.*

McCarthy, J. (1980). Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*.

McCarthy, J. (1986). Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*.

Reiter, R. (1980). A logic for default reasonig. *Artificial Intelligence*, pages 81–132.

Sombé, L. (1990). *Reasoning under incomplete information in artificial intelligence: a comparison of formalisms using a single example.* John Wiley and Sons Inc.

Suchenek, M. (2006). On undecidability of non-monotonic logic. *Studia Informatica*.

Toulgoat, I. (2011). *Modélisation du comportement humain dans les simulations de combat naval.* PhD thesis, Laboratoire SNC/DCNS.

Toulgoat, I., Siegel, P., and Doncescu, A. (2011). Modelling of submarine navigation by nonmonotonic logic. *BWCCA*.