6.0 Acknowledgment

The author is being supported by a joint scholarship grant of the Engineering and Science Education Program, Department of Science and Technology of the Philippine government and De La Salle University. This work would not have been possible without the support and supervision of Sharon Wood.

References

[Bulo94a] Bulos, R. dD, (1994). Goal Formulation as an AI Research Issue. In the *7th White House Paper*, de Bourcier, P., Lenmen, R. & Thompson, A. (Eds.), School of Cognitive and Computing Science, University of Sussex, UK.

[Bulo94b] Bulos, R. dD., (1994). Goal Formulation as an AI Research Issue*. In the *Proceedings of the 6th De La Salle University Computer Conference,* De La Salle University, Manila, Philippines.

[Bulo95] Bulos, R. dD. (1995). Goal Detection Representation Mechanisms and Architecture. In the *Proceedings of the 8th Florida AI Research Symposium,* Stewman, J. (Ed.), Florida: Florida AI Research Society.

[GI89] Georgeff, M.P. & Ingrand, F.F. (1989). Decision-making in an Embedded Reasoning System. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence,* pp. 972-978, Detroit, Michigan.

[GL87] Georgeff, M.P. & Lansky, A.L. (1987). Reactive Reasoning and Planning. In *Proceedings of AAAI 1987*, pp. 677-682.

[GS93] Goodwin, R. & Simmons, R. (1993). *Rational Handling of Multiple Goals for Mobile Robots.* Technical Report CMU-CS-93-204, School of Computer Science, Carnegie Mellon University.

[HLetal94] Huber, M.J., Lee, J., Kenny, P. & Durfee, E.H. (1994). *UM-PRS V2.0 Programmer User Guide.* Department of Electrical Engineering and Computer Science, The University of Michigan.

[KBetal94] Kenny, P.G., Bidlack, C.R., Kluge, K.C., Lee, J., Huber, M.J., Durfee, E.H., & Weymouth, T. (1994). Implementation of a Reactive Autonomous Navigation System on an Outdoor Mobile Robot. In the *Proceedings of AUV '94.*

[Lehr93] Lehr, N.B. (1993). *DARPA/Rome Laboratory Planning and Scheduling Initiative Knowledge Representation Specification Language (KRSL) Version 2.0.2.*

[LHetal94] Lee, J., Huber, M.J., Durfee, E.H. & Kenny, P.G (1994). UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications. In the *Proceedings of the Conference on Intelligent Robotics in Field, Factory, Service and Space*, pp. 842-849, Houston Texas.

[MB93] McFarland, D & Bosser, T. (1993). *Intelligent Behavior in Animals and Robots.* Cambridge, MA: MIT Press

[Mcca94] McCarthy, J. (1994). *Making Robots Conscious of their Mental States.* Computer Science Department, Stanford University. URL: http/www-formal.stanford.edu/jmc/.

[SS79] Sassone, P.G. & Schaffer, W.A. (1979). *Cost-Benefit Analysis A Handbook.* Academic Press Inc.

[Wile83] Wilensky, R. (1983). *Planning and Understanding A Computational Approach to Human Reasoning*. Reading, MA: Addison-Wesley Publishing Company.

[WM94] Wilkins, D.E. & Myers, K.L. (1994). *A Common Knowledge Representation for Plan Generation and Reactive Execution.* SRI AI Center Technical Note 532R, SRI International, Menlo Park, CA.

[WMetal95] Wilkins, D.E., Myers, K.L., Lowrance, J.D., & Wesley, L.P. (1995). Planning and Reacting in Uncertain and Dynamic Environments. *Journal of Experimental and Theoretical AI*, 7(1).

the utility of the plan generates a net total benefit to the overall performance of the agent, that is, positive results outweigh the negative ones.

This work has emphasized the need for assessing the feasibility of a goal. It has identified the requirements, provided a system architecture and a developed a control strategy for goal feasibility assessment. Future work will involve the development of an enhanced automated reasoning mechanism for estimating the urgency of the goal and the development of a reasoning mechanism to evaluate the priority of the goal by taking into consideration the several factors that affect a goal's priority such as urgency, expected utility, goal conflict and goal coordination.

Figure 4 Goal Feasibility Assessment Sample Process Run

THE GOAL IS: goal(goalDescriptor(achieve), motive(selfPreservation),desire(objectName(powerMeter),objectProperty(capacity), objectPropertyValue(full)), stimuli([currentBelief(objectName(powerSupply), objectProperty(availability), objectPropertyValue(insufficient), beliefConfidence(1.0), status(new))]), mechanism([dataDriven(2, achieve), dataDriven(1, achieve)]))
Evaluating the urgency of the goal ... in progress. The URGENCY of the goal is 0.
Searching for a new plan ... in progress. NO NEW (OTHER) PLAN GENRERATED
Searching for an existing plan in the plan library... in progress
An old PLAN/ROUTINE (ANOTHER) : labelName(rechargePowerSupply(1))EXISTS FOR THE GOAL
NO (other) EXISTING PLAN IS GENERATED/FOUND
Now searching whether the goal is in the CAUSAL RULE ....(i.e.CAUSAL SEARCH .... in progress)
NO (other) PLAN FOUND THRU CAUSAL SEARCH
Now evaluating whether plans generated/found are feasible
The plan with label name labelName(rechargePowerSupply(1))is being considered
HARD CONSTRAINTS ARE: [checkVerity(not(currentBelief(objectName(recharger), objectProperty(operationalStatus), objectPropertyValue(notOk), beliefConfidence(1.0), status(_1)))), checkVerity(not(currentBelief(objectName(powerMeter), objectProperty(capacity), objectPropertyValue(full), beliefConfidence(1.0), status(_2))))]
NOW SATISFYING THE HARD CONSTRAINTS
The agent does not have a current belief: currentBelief(objectName(recharger), objectProperty(operationalStatus), objectPropertyValue(notOk), beliefConfidence(1.0), status(_1))
THEREFORE: checkVerity(not(currentBelief(objectName(recharger), objectProperty(operationalStatus), objectPropertyValue(notOk), beliefConfidence(1.0),status(_1)))) HAS BEEN MET
The agent does not have a current belief: currentBelief(objectName(powerMeter), objectProperty(capacity), objectPropertyValue(full), beliefConfidence(1.0), status(_2))
THEREFORE: checkVerity(not(currentBelief(objectName(powerMeter), objectProperty(capacity), objectPropertyValue(full), beliefConfidence(1.0), status(_2)))) HAS BEEN MET
ALL HARD CONSTRAINTS ARE SATISFIED
Computing for Cost-Benefit of the plan.... in progress
Procedure is : [execute(freeRobot), execute(goto(_3, h)), execute(putDown(_4)), execute(turnOff(rechargerSwitch)), execute(plugOnto(rechargerSource, powerSupplyPlug)), execute(turnOn(rechargerSwitch)), waitUntil(object(objectName(powerMeter),objectProperty(capacity), objectPropertyValue(full)))]
Now computing the costs and time in executing plan
The operator is : execute(freeRobot)
Observe the value for: objectName(robot),objectProperty(attachment),objectPropertyValue is ? : ? free, beliefConfidence is ? : ? 1.0
ITS ENERGY COST IS : 0 ; ITS TIME PERIOD OF EXECUTION IS: 0 seconds
The operator is : execute(goto(_3, h))
Observe the value for:objectName(robot),objectProperty(location),objectPropertyValue is ? : ? a, beliefConfidence is ? : ? 1.0
ITS ENERGY COST IS: 40; ITS TIME PERIOD OF EXECUTION IS : 40 seconds

............(computing costs of the other operators in the procedure)......

Total Cost of execution is (Net present Value) : 76.0396; Total time of execution is : 649
Computing the benefits of goal achievement ..... in progress
Total reward points due to goal achievement (Net Present Value): 209.73242
Computing for net cost-benefit; Net CBA or Expected Utility is: 133.69282
Resources used are: [energy(48), time(649)]
Resources generated are: []
Resources Provided are: []
Now Checking the FEASIBILITY of the plan:
Hard Constraints are satisfied;
Utility is GREATER than ZERO
ALL the Resources needed for Plan Execution are AVAILABLE (energy Available = 50 Needed = 48 ; time Available = 50400 Needed = 649)
THIS PLAN IS FEASIBLE TO PURSUE
Has Completed Evaluation of all generated/found plans
Now selecting the best plan option for the goal
BEST PLAN OPTION is: labelName(rechargePowerSupply(1))
UPDATED GOAL is: goal(goalDescriptor(achieve), motive(selfPreservation), desire(objectName(powerMeter), objectProperty(capacity), objectPropertyValue(full)),stimuli([currentBelief(objectName(powerSupply), objectProperty(availability), objectPropertyValue(insufficient), beliefConfidence(1.0),status(new))]), mechanism([dataDriven(2, achieve), dataDriven(1, achieve)]), expectedUtility(133.69282), urgency(0), resourcesUsed([energy(48), time(649)]), resourcesGenerated([]), resourcesProvided([]), checkHC(yes), goalStatus(forExecution)).

terminated.

If a plan is found, a simultaneous satisfaction of hard constraints and the estimation of the expected utility of every generated/found plan commence. *The goal is evaluated as doable if a plan is found and if the hard constraints of the plan are satisfied.* Satisfaction of hard constraints and assessment of expected utility are treated as parallel processes.

Early planners generally regard a precondition as a state of the world that must be true before an operation can take place and it is usually achievable by another plan or operator. However, separating hard constraints from ordinary preconditions and prioritizing their satisfaction prior to plan execution avoid unnecessary waste of resources and effort attributed to uninformed execution. *Obtaining prior information through the satisfaction of hard constraints enables the agent to obtain an early decision about the fate of the plan.*

Only plans with positive expected utilities are considered. The plan which has the highest expected utility is chosen. A sample process run illustrating the control algorithm/strategy for assessing the feasibility of a goal, is shown in Figure 4.

Figure 3 Outline of the Goal Feasibility Assessment Control Algorithm

1st stage: Concurrent processing of:
1) urgency evaluation
2) new plan generation
3) search for an existing plan from the plan library (direct search)
4) search for an existing plan from the plan library (causal search)

If a plan is not constructed in the first stage, the goal is rendered non-doable; otherwise, control shifts to the second stage;

2nd stage: Concurrent processing of:
5) satisfaction of hard constraints
6) estimation of expected utility (cost-benefit analysis using Net Present Value method)

If any of the hard constraints included in the plan is non-satisfiable or the computed expected utility of the goal is negative, then the plan is dropped; otherwise, control shifts to the third stage;

3rd stage: 7) selection of the best alternative plan

The plan with the highest expected utility value is considered.
If a plan with positive expected utility is not found, then the goal is terminated.

5.0 Summary, Conclusion and Future Work

To address the issue of providing an agent the consciousness to determine the goals it can currently achieve and discern its choices of actions, this work has put forward the idea of goal feasibility assessment, a reasoning mechanism that evaluates whether a goal is feasible to achieve. For a goal to be feasible, it must be both doable and rational.

An autonomous intelligent agent may achieve its given goal in a feasible way, if: a) it has the capability to affect the world; that is, if it has the built-in operators that will construct the plan to produce the intended effects; b) it has the resources or means to effectively carry out the actions; c) the environmental situations that satisfy the hard constraints are existing; and d) existing, insufficient, not functioning or being used by another agent or otherwise unavailable. A situational constraint is unsatisfiable if it does not exist.

Hard constraints are represented as relations with a predicate descriptor called "*checkVerity*" and a list of arguments that must be satisfied. The predicate descriptor *checkVerity* calls a procedural function that tests the verity of the argument.

The procedure is the most important component of the action plan. It embodies the set of subgoals or set of actions needed to achieve the goal. It reflects the capability and knowledge of the agent to achieve the goal. It is described as a directed graph with a partially ordered list of nodes representing the subgoals and/or actions (see Figure 2). A list of nodes indicates sequential processing. An "*or*" predicate descriptor denotes disjunctive branching while an "*and*" predicate signifies conjunctive branching.

The procedure contains a list of relations that may have varied predicate descriptors such as *verifyMaintainAchieve*, *checkVerity*, *execute*, *executeWhile*, *executeUntil, wait, waitWhile,* and *waitUntil*. The *verifyMaintainAchieve* predicate descriptor either achieves, maintains or verifies the argument (which is a goal), depending on the verity of the desire. The checkVerity predicate descriptor tests the verity of its argument. The *execute*, *executeWhile*, and *executeUntil* predicate descriptors specify actions while the *wait, waitWhile, waitUntil* predicate descriptors check for situations that are expected to occur. In the example shown in Figure 2 (see plan), the predicate descriptor used is "execute".

Lastly, the "effects" component of the plan contains a list of relations that need to be updated and reflected in the agent's current beliefs. Such effects are the result of successful plan execution. The predicate descriptor *updateBeliefs* [9]is used for this purpose.

4.0 Control Strategy

The goal feasibility assessment control algorithm is outlined in Figure 3. It consists of three stages. Plan generation/search and urgency evaluation are processed in the first stage; hard constraints satisfaction and the computation of expected utility value for each generated alternative plan are undertaken during the second stage; and selection of the best plan for the goal is done in the third stage.

Once a goal has been detected by the agent, a simultaneous process of generating a new plan, searching for an existing plan and determining the urgency of a detected goal is undertaken. Since plan generation and plan search are independent processes, they can be executed in parallel. If no plan is generated or found, then further processing of the goal is

9. When updating its beliefs the agent reasons about whether to store the facts or events that have transpired. *Some facts are important for the agent to know, others are extremely changeable so that confidence in their verity is highly questionable.* Storing insignificant facts would just be a waste of memory usage. To be able to reason whether a fact is significant enough to be retained, the agent is provided with a list of "must be stored" object properties.

to be maintained if the desire's verity is true. As illustrated in Figure 2, the goal descriptor is classified as "achieve" because the desire of "having full power capacity" is opposite the agent's belief that "the power meter capacity is in the warning level".

A goal also has a corresponding expected utility value and an urgency. The expected utility measures the net benefits gained by the agent when the goal is successfully achieved while the urgency indicates the latest start time for executing the plan to achieve the goal.

3.4 Plans

In AI Planning, reasoning about the achievability of a goal has often been associated with the production of a correct *plan*, that is, the identification of a *sequence of actions* or the determination of a *series of subgoals* to achieve a goal, given some initial conditions in the environment. Today, as agents have become more sophisticated and as domains have gone complex, the concept of a plan has become more elaborate. Although different planners may have different constructions, there appears to be a set of plan components that are common among them. Figure 2 shows an action plan representation that is described using Prolog like statements. It has most of the features found in other plan formalisms [GI89, GL87, HLetal94, KBetal94,Lehr94, LHetal94, WM94, WMetal95][8]; however, it has also some features (particularly the predicate descriptors) which differentiate it from the rest.

Although the use of terminology varies among planners, a plan can be described as consisting of a *label name*, a *goal, hard constraints, procedure* and *effects*. The label name is for documentation purposes and the "plan number" suggests that there can be more than one action plan to achieve the same goal. The goal is the primary effect of the plan. It consists of a goal descriptor and a desire.

Hard constraints are states of the world that must be satisfied prior to the execution of the plan, otherwise the plan will fail. They may be considered to be stricter types of preconditions in the sense that either the agent does not have the capability and know-how to make them happen or the agent does not act in order for them to become true. They are satisfied by querying about their verity in the environment and are not acted upon. They are regarded as primary requirements and their non-satisfaction would lead to automatic plan failure. Most hard constraints can be categorized as resource constraints or situational conditions generated by the environment, ones which the agent has no control. A *resource* is something required to execute a plan. Time, consumable functional objects, nonconsumable functional objects, and abilities are classes of resources. [Wile83] A resource constraint is unsatisfiable if it is not

---

7. This indicates non-knowledge about the verity of the desire.
8. The planners described by these authors (except Lehr94) were based from Georgeff's Procedural Reasoning System (PRS). Wilkins [WM94,WMetal95] developed the ACT formalism based on the formalism of SIPE and PRS.

Goals are stored and organized according to a *dynamic hierarchy*, that is, as the agent adapts to the constantly changing environment, goals become temporarily superordinate or subordinate in the hierarchy. They are chosen depending on the *relative position* they occupy in the hierarchy, their *urgency* and their *potential for realization*.They are implemented by the agent through the formation and execution of plans (sequence of actions). And as the agent executes the plans to achieve its goals, goals may either be eventually achieved, suspended by some internal or external events that results in the temporary taking precedence of another goal, or evaluated to be unattainable, at least for the meantime. [Bulo94a, Bulo94b].

Figure 2
Nature and Representation of Knowledge
that is Relevant to Goal Feasibility Assessment

BELIEF

```
currentBelief(
    objectName(powerMeter),
    objectProperty(capacity),
    objectPropertyValue(warning),
    beliefConfidence(1.0)).
```

CAUSAL RULE

```
causalRule(
    cause([currentBelief(
        objectName(powerMeter),
        objectProperty(capacity),
        objectPropertyValue(full),
            beliefConfidence(1.0))]),
    effect([currentBelief(
        objectName(powerSupply),
        objectProperty(availability),
        objectPropertyValue(sufficient),
        beliefConfidence(1.0))])).
```

GOAL

```
goal(
    goalDescriptor(achieve),
    motive(selfPreservation),
    desire(
        objectName(poweMeter),
        objectProperty(capacity),
        objectPropertyValue(full)),
    stimuli([ currentBelief(
        objectName(powerMeter),
        objectProperty(capacity),
        objectPropertyValue(warning),
        beliefConfidence(1.0),status(new))]),
    mechanism([]),
    expectedUtility(844), urgency(0)).
```

PLAN

```
actionPlan(
    labelName(rechargePowerSupply(1)),
    goal(goalDescriptor(achieve),
        desire(objectName(powerMeter),objectProperty(capacity),objectPropertyValue(full))),
    hardConstraints([or([checkVerity(currentBelief(objectName(recharger),
        objectProperty(operationalStatus),objectPropertyValue(ok),
        beliefConfidence(1.0))),...]),
        checkVerity(...),...]),
    procedure([execute(freeRobot), execute(goto(_3, h)), execute(putDown(_4)),
        execute(turnOff(rechargerSwitch)), execute(plugOnto(rechargerSource, powerSupplyPlug)),
        execute(turnOn(rechargerSwitch)), waitUntil(object(objectName(powerMeter),
        objectProperty(capacity), objectPropertyValue(full)))]
    effects(updateBeliefs(currentBelief(objectName(powerMeter),objectProperty(capacity),
        objectPropertyValue(full),beliefConfidence(1.0))))).
```

The representational structure of a goal is illustrated in Figure 2 (see the predicate goal). A goal is described as having a *goal descriptor, motive, desire, stimuli, expected utility and urgency.* A motive compels the agent to act in order to achieve a desired state. It may be described as a high level form of desire that drives the agent to satisfy other lower level desires. Associated with motives are desires. Desires are descriptions of states that the agent wishes to be achieved (or avoided) when performing its role. They are activated and transformed into goals when the conditions (called stimuli) that stimulate the agent to satisfy a motive emerge. Both motive and desire are represented as object-property-value trios, while a stimuli is represented as a list of arguments that depicts time or situations[6].

The goal descriptor indicates whether a goal should be *achieved, verified or maintained.* A goal is to be achieved if the desire is not a current belief of the agent or it is said to be currently false; it is to be verified if the agent does not know whether the desire is true or false[7]; and it is

6. These situations are in the form of current beliefs.

acted upon at the earliest possible time). Goals with specified start times have urgency values equivalent to their start times (e.g. if the goal of cooking for lunch is scheduled at twelve noon, then the urgency value of the goal will be 1200). The urgency values for communicated goals are explicitly assigned by the external agent.

3.0 Knowledge Base: Nature and Representation

To endow an agent with the capability of assessing the feasibility of its goals, it must possess and reason with: *beliefs* it has about the world, *causal effects* of its beliefs, *goals* that drive it to change its current world into a more desirable one, and *plans* that are constructed and then executed to achieve those goals. Figure 2 illustrates the nature and structural representation of knowledge that is relevant to Goal Feasibility Assessment.

3.1 Beliefs

The agent's beliefs are characterized as explicit or derived facts about the present status of object properties in the environment. They can be described as a collection of *object-property-value* trios (which can also be called a *state*), that is, each known property of an object in the environment has a corresponding value associated with it. Also, associated with each object-property-value trio is the *degree of confidence* for its verity. Figure 2 illustrates the representation and instantiation of an agent's belief (see currentBelief predicate). The example provided simply means that the agent believes (100 percent) that the current capacity of the power meter is in the "warning" level.

3.2 Causal Rules

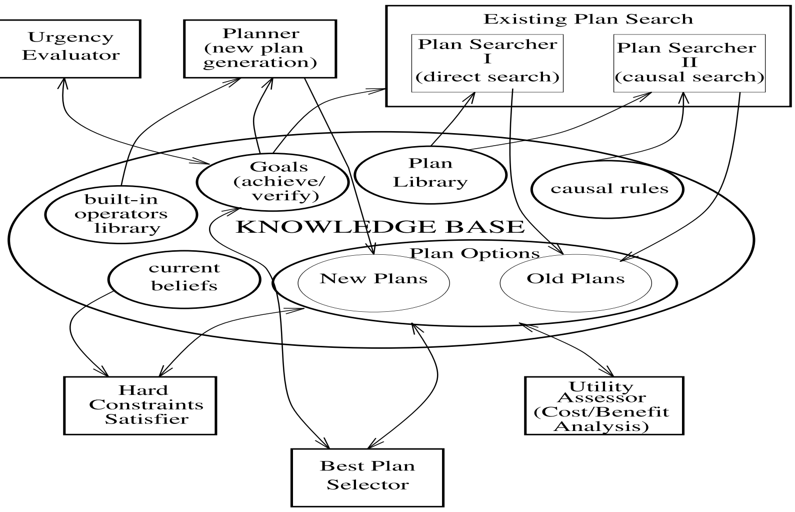
A causal rule indicates a cause-and-effect relationship among the beliefs of the agent, such that the causes bear a direct influence on the effects. It is represented as consisting of two arguments, namely: causes and effects. If the list of causes are currently true, then it can be said that the listed effects are also true (see the causalRule predicate of Figure 2).

3.3 Goals

Aside from possessing a set of beliefs, an agent is also regarded as a *purposive system* that is being driven behaviorally by a set of goals. *Goals* are defined as the end results that an agent would like to achieve or avoid. They are specific and measurable outcomes toward which an agent's efforts are directed within specified time and cost constraints. They are primarily acquired by an agent through association with other agents, objects, events, symbols and behavior. They are either *prescribed* (by another agent) or *self-formulated*. They are often *multiple*, *occur simultaneously* and may *interact with each other*.They are detected at a specific time intervals by the stimuli generated by the environment. According to [Bulo95], goals can be detected by four reasoning mechanisms namely: current data-driven, goal-driven, anticipated data-driven and through communication.

conditions that must be present in the environment for a plan to be considered as implementable. Their non-satisfaction automatically leads to plan failure and consequently renders the goal non-doable. In the system architecture, the Hard Constraints Satisfier is the component that tests whether the enumerated hard constraints in the plan are satisfied.

Figure 1 Goal Feasibility Assessment System Architecture



Reasoning about the rationality of pursuing a goal involves: a) the computation of costs attributed to the plan as well as the estimation of benefits derived in implementing the plan and b) selection of the best plan for the goal. The analysis of costs and benefits for the plan produces an expected utility. The Utility Assessor, which runs in parallel with the Hard Constraints Satisfier, is the component that determines the expected utility derived from successful plan implementation. It uses the net present value[4] decision criterion in the computation of the expected utility. The expected utility is the net result of the discounted benefits and costs (which include real and opportunity costs).

The Best Plan Selector compares the various alternative plans generated and chooses the plan that returns the highest utility value. It is undertaken after all alternative plans have been generated/searched and their utility values computed. If a plan with a positive utility value cannot be selected, then the goal is considered to be infeasible.

The Urgency Evaluator is an independent processor that computes for the latest start time needed in order to achieve the goal. It may run concurrently with the plan generation components. Goals that are activated in response to stimuli generated by the dynamic environment are assigned with urgency values[5] of zero (which means that they have to be

4. The net present value method is the most widely used and by far adjudged to be superior to other cost-benefit analysis metrics. [GS93, SS79]
5. Urgency is expressed as an absolute time (e.g. 1000 means 10 o'clock; 0 means "now").

possible to achieve), deliberate whether the utility derived in achieving the goal outweighs the costs incurred in carrying out the plan to achieve the goal.

2.0 System Architecture

The system architecture for goal feasibility assessment is depicted in Figure 1. The control strategy of the system was implemented in PROLOG using a serial machine. It simulates a hypothetical robot doing some household chores.The system architecture consists of a knowledge base and seven processors, namely: Urgency Evaluator, Planner, Plan Searcher I, Plan Searcher II, Hard Constraints Satisfier, Utility Assessor and Best Plan Selector. The knowledge base contains the nature and structure of knowledge that the agent requires for assessing the feasibility of a goal. This includes the detected goals, causal rules, existing plan library, current beliefs, built-in operators library and active plan options[3].

Goal Feasibility Assessment involves a two-step reasoning process namely: 1) reasoning about the doability of the goal and 2) reasoning about the rationality to pursue a goal. Assessing the doability of a goal is likewise a two-step reasoning activity. The first step involves reasoning about how to achieve the goal, given the agent's knowledge about the world and its capability to affect it. If successful, this reasoning process produces a *plan of action*; otherwise, the goal is considered non-doable (that is, no plan is constructed).

In the system architecture shown in Figure 1, the Planner, Plan Searcher I and Plan Searcher II are the plan generation components of the system that function in parallel. The Planner generates new plans for the goal using non-linear search. The Plan Searcher I and Plan Searcher II search for existing plans that are available in the plan library. The Plan Searcher I uses direct search in looking for an existing plan while Plan Searcher II makes use of causal rules to search for an existing plan for the goal. Direct search for an existing plan involves matching of the goal's and a plan's (accessed in the plan library) two arguments namely, goal descriptor and desire. If a match is found, the plan is considered as an alternative solution to accomplish the goal. On the other hand, causal search for an existing plan involves the use of causal rules by determining whether a goal is considered as an effect of a causal condition. If the desire argument of the goal matches one of the effects listed in the causal rule, then the causal condition becomes the new goal; subsequently a direct search for an existing plan to achieve the new goal is undertaken.

The second step in evaluating the doability of a goal involves the *satisfaction of hard constraints* of the generated plan. Hard constraints are descriptions of the world whose existence are verifiable. They are only tested for their verity (by checking current beliefs) and are never made or acted upon to become true. They specify the minimum requirements/

3. The active plan options are the plans generated or searched during the goal feasibility assessment process.

1.0 Motivation

Among several others, agents operating in complex and dynamic domains are confronted with two major problems, namely: *1) goals of the agent are (often) multiple and may occur simultaneously, and 2) agents are resource-bounded.* Resource constraints, as well as dynamic situations that occur in the environment, compel the agent to reason and eventually choose the goals it considers possible and practical to achieve.To avoid or minimize the waste of resources and exertion of useless effort in pursuing goals that *are destined to fail or prove to be inconsequential to the greater benefit of the agent and its environment*, the agent must be conscious about its goals and not just blindly execute them.

McCarthy suggests that in order to perform various and complicated tasks, an agent such as a robot must possess some form of self-consciousness in its mental processes. He discusses that one property of consciousness that agents (e.g. robots) should have is *"knowing what goals it can currently achieve and what its choices are for action"* [p. 2, Mcca94]. To address the issue of providing an agent with the consciousness to determine the goals it can currently achieve and discern its choices of actions, this paper posits the idea of *goal feasibility assessment*[1].

Goal feasibility assessment is a reasoning process for evaluating whether a goal is both *doable* and *rational.* A goal is considered to be doable, if the requirements for successful goal achievement match the agent's capabilities, available resources, and knowledge about the world as well as the presence of favorable situational conditions. This means that the agent must possess both the *capability* and *knowledge* to pursue the goal as well as satisfy the resource and situational constraints pertinent to the achievement of the goal. On the other hand, a goal is said to be rational to achieve if it is evaluated to *contribute positively*[2]or it has been assessed to prevent any harm or damage (negative effects) to the agent and its environment after considering all available options.

Primarily, Goal Feasibility Assessment is undertaken in order to avoid unnecessary waste of effort, time and resources when a goal proves to be unattainable. Thus, before committing the necessary resources and expending effort to effectively carry out the tasks, the agent must first intelligently reason whether a goal is possible to achieve and subsequently (if found

1. To be truly called "intelligent", an agent must not only be capable of knowing how to achieve its given goals; preferably, it must also have the capability to formulate its own goals. It must reason and decide what goals to pursue and when to achieve them. It must be able to identify its own goals (Goal Detection), assess their feasibility (Goal Feasibility Assessment), prioritize them (Goal Prioritizing), evaluate or validate their status as to whether they have to be continued, terminated, suspended, or modified (Goal Evaluation/Validation), and modify them in the light of present circumstances (Goal Modification). Of the five integrated processes mentioned above that constitute goal formulation, this paper focuses on Goal Feasibility Assessment [Bulo94a, Bulo94b, Bulo95].
2. This means that the net total benefit or net utility value that the agent gains for successfully accomplishing a goal should be greater than zero.

Goal Feasibility Assessment: Architecture, Representation and Control Strategy

Remedios de Dios Bulos

School of Cognitive and Computing Sciences
University of Sussex

Falmer, Brighton BN1 9QH

United Kingdom

email: remedios@cogs.susx.ac.uk

CSRP 403

January, 1996

Abstract

This paper posits the idea of goal feasibility assessment, a reasoning mechanism that is intended to give the agent the consciousness to decide early and rationally on the fate of its goals. It may be described as a reasoning process for evaluating whether an agent's detected goal is possible (or doable) and rational (practical) to achieve. The doability of a goal is evaluated based on the capability and knowledge of the agent to pursue the goal as well as the satisfaction of resource and situational constraints pertinent to the achievement of the goal. On the other hand, the rationality of pursuing a goal is assessed based on the net total benefit (or net utility value) attributed to the agent and its environment when a goal is successfully achieved.

In addressing the issue of goal feasibility assessment, this paper seeks to determine the nature and representation of knowledge that the agent requires in assessing the feasibility of its goals. It also aims to define a control strategy for goal feasibility assessment and provide a supporting architecture. In brief, goal feasibility assessment shall be addressed through architecture, representation and control strategy.

The system architecture consists of a knowledge base and seven processors, namely: Urgency Evaluator, Planner, Plan Searcher I, Plan Searcher II, Hard Constraints Satisfier, Utility Assessor and Best Plan Selector. The knowledge base contains the nature and structure of knowledge that the agent requires for assessing the feasibility of a goal. This includes the detected goals, causal rules, existing plan library, current beliefs, built-in operators library and active plan options.

The goal feasibility assessment control algorithm consists of three stages. Plan generation/search and urgency evaluation are processed in the first stage; hard constraints satisfaction and computation of expected utility value for each alternative plan are undertaken during the second stage; and selection of the best plan for the goal is done in the third stage.