SOCS

A COMPUTATIONAL LOGIC MODEL FOR THE DESCRIPTION, ANALYSIS AND VERIFICATION OF GLOBAL AND OPEN SOCIETIES OF HETEROGENEOUS COMPUTEES

IST-2001-32530

Computees and welfare

Project number:	IST-2001-32530
Project acronym:	SOCS
Document type:	IN (information note)
Document distribution:	I (internal to SOCS and PO)
CEC Document number:	IST32530/ICSTM/72/IN/I/b
File name:	172-b[welfare].pdf
Editor:	F. Toni
Contributing partners:	ICSTM, DIPISA, CITY
Contributing workpackages:	WP5
Estimated person months:	N/A
Date of completion:	11 March 2005
Date of delivery to the EC:	18 March 2005
Number of pages:	22

ABSTRACT

This is a report on preliminary work on welfare-related properties for computees, providing an annex to deliverable D13.

Computees and welfare

P. Mancarella[1], **F. Sadri**[2], **K. Stathis**[3], **F. Toni**[2], **A. Bracciali**[1]

[1] Dip. di Informatica, Università di Pisa, Email: `paolo@di.unipi.it`

[2] Dept. of Computing, Imperial College London, Email: {`fs,ft`}`@doc.ic.ac.uk`

[3] School of Informatics, City University London, Email: `kostas@soi.city.ac.uk`

ABSTRACT
This is a report on preliminary work on welfare-related properties for computees, providing an annex to deliverable D13.

Contents

1 Introduction

In this report, we set the ground and provide some initial definitions and results for properties of individual computees which are defined in terms of the *individual welfare* of computees. We consider two notions of individual welfare:

1. a notion of *happiness*, referring solely to the achieved and unachievable top-level goals of computees;

2. a notion of *progress*, referring to the whole state of computees, including their planned actions and the recorded observations in their knowledge bases.

These notions may depend upon quantitative measures (e.g. the number of achieved goals), but are fundamentally qualitative (e.g. in that they rely upon declarative notions of goal achievement). In particular, if the agent is equipped with a single top-level goal, individual welfare via happiness corresponds to success in achieving the goal.

In general, the notion of individual welfare might be *subjective*, referring to the beliefs of the computee (even though these may be assessed from the outside, i.e. not by the computee itself) or *objective*, referring to the perception of the welfare of the computee by an outsider, that can measure, e.g., how many goals the computee has achieved in reality, with respect to the actual environment in which it is situated. In this report we basically take a subjective view of individual welfare: goals are achieved if they can be proven subjectively by the computee via its Temporal Reasoning capability. However, it is an external observer that establishes when to make use of this capability, from the outside, by looking at the state-sequence and the current time.

In this report, we set the ground for specifying and proving properties of computees in terms of their individual welfare. In particular, we define five welfare-based orderings amongst computees'states, a notion of improvement over the life-cycle of computees, parametric wrt any ordering, and instantiate the general notion of improvement wrt the various orderings. We prove that computees can be shown always to improve according to one given ordering (defined in terms of happiness as the number of achieved top-level goals), never to improve according to another given ordering (defined in terms of unhappiness as the number of unachievable top-level goals).

2 Recap of some notions underlying the KGP model

Internal state. This is a tuple $\langle KB, Goals, Plan, TCS\rangle$, where:

- KB is the knowledge base of the computee, and describes what the computee knows (or believes) of itself and the environment. KB consists of modules supporting different reasoning capabilities:
  - KB_{plan}, for Planning,
  - KB_{pre}, for the Identification of Preconditions of actions,
  - KB_{TR}, for Temporal Reasoning,
  - KB_{GD}, for Goal Decision,

– KB_{react}, for Reactivity, and
– KB_0, for holding the (dynamic) knowledge of the computee about the external world in which it is situated (including past communications).

Syntactically, KB_{plan},KB_{react} and KB_{TR} are abductive logic programs with constraint predicates KB_{pre} is a logic program, KB_{GD} is a logic program with priorities and KB_0 is a set of facts in logic programming, and it is (implicitly) included in all the other modules.

- *Goals* is the set of properties that the computee wants to achieve, each one explicitly time-stamped by a time variable. Syntactically, each goal is a pair of the form $\langle l[t], G' \rangle$ where

  – $l[t]$ is the *fluent literal* of the goal, referring to the time t;
  – G' is the *parent* of G.

- *Plan* is a set of actions scheduled in order to satisfy goals. Each is explicitly time-stamped. Syntactically, each action is a triple of the form $\langle a[t], G, C \rangle$ where

  – $a[t]$ is the *operator* of the action, referring to the execution time t;
  – G the parent goal, towards which the action contributes (i.e., the action belongs to a plan for the goal G). G may be a post-condition for A (but there may be other such post-conditions).
  – C are the *preconditions* which should hold in order for the action to take place successfully; syntactically, C is a conjunction of (timed) fluent literals.

- *TCS* is a set of constraint atoms (referred to as *temporal constraints*) in some given underlying constraint language with respect to some structure $\Re$ equipped with a notion of constraint satisfaction $\models_\Re$. In the sequel, given a set C of sentences built from constraint atoms, $\models_\Re C$ will stand for C is $\Re$-satisfiable, and $\not\models_\Re C$ will stand for C is $\Re$-unsatisfiable.

  Temporal constraints bound the time variables of goals and actions, thus implicitly defining when goals are expected to hold and when actions should be executed. Also, via the temporal constraints, actions are partially ordered.

Goals and actions are uniquely identified by their associated time, which is implicitly existentially quantified within the overall state. To aid revision and partial planning, *Goals* and *Plan* form two *trees*, whose roots are represented by $\perp^{nr}$ and $\perp^{r}$, and indicated respectively as the *non-reactive tree*, containing *non-reactive* goals and actions, and the *reactive tree*, containing *reactive* goals and actions. All the top-level non-reactive goals are either assigned to the computee by its designer at birth, or they are determined by the Goal Decision capability. All the top-level reactive goals and actions are determined by the Reactivity capability. When clear from the context, we will refer to the state as a single tree with root $\perp$. The tree is given implicitly by associating with each goal and action its parent (the second element in the corresponding tuple). *Top-level* goals and actions are children of the root of the tree, $\perp$.

Valuation of temporal constraints. Given a state $S = \langle KB, Goals, Plan, TCS \rangle$, we denote by $\Sigma(S)$ (or simply Σ, when S is clear from the context) the valuation:

$$\Sigma(S) = \{t = \tau \,|\, executed(a[t], \tau) \in KB_0\} \cup \{t = \tau \,|\, observed(l[t], \tau) \in KB_0\}$$

Intuitively, $\Sigma(S)$ extracts from KB_0 in S the instantiation of the (existentially quantified) time variables in *Plan* and *Goals* in S, derived from having executed (some of the) actions in *Plan* and having observed that (some of the) fluents in *Goals* hold (or do not hold). KB_0 in S provides a "virtual" representation of $\Sigma(S)$.

Transitions. The state of a computee evolves by applying transition rules, which employ capabilities and the constraint satisfaction $\models_{\Re}$. The transitions are:

- Goal Introduction (GI), changing the top-level *Goals*, and using Goal Decision.
- Plan Introduction (PI), changing *Goals* and *Plan*, and using Planning and Introduction of Preconditions.
- Reactivity (RE), changing *Goals* and *Plan*, and using the Reactivity capability.
- Sensing Introduction (SI), changing *Plan* by introducing new sensing actions for checking the preconditions of actions already in *Plan*, and using Sensing.
- Passive Observation Introduction (POI), changing KB_0 of KB by introducing unsolicited information coming from the environment, and using Sensing.
- Active Observation Introduction (AOI), changing KB_0 of KB, by introducing the outcome of (actively sought) sensing actions, and using Sensing.
- Action Execution (AE), executing all types of actions, and thus changing KB_0 of KB.
- State Revision (SR), revising *Goals* and *Plan*, and using Temporal Reasoning and Constraint Satisfaction.

SR merges the revision transitions of the original KGP model as given in [4, 7, 6], i.e. GR and PR. This avoids having to repeatedly applying the GR and PR to get chained revisions. The specification of the new transition is given in the main deliverable D13.

Cycle theory. Formally, a cycle theory $\mathcal{T}_{cycle}$ consists of the following parts.

- An *initial* part $\mathcal{T}_{initial}$, that determines the possible transitions that the agent could perform when it starts to operate (*initial cycle step*). More concretely, $\mathcal{T}_{initial}$ consists of rules of the form

  $$*T(S_0, X) \leftarrow C(S_0, \tau, X), now(\tau)$$

  sanctioning that, if the conditions C are satisfied in the initial state S_0 at the current time τ, then the initial transition should be T, applied to state S_0 and input X, if required. Note that $C(S_0, \tau, X)$ may be empty, and $\mathcal{T}_{initial}$ might simply indicate a fixed initial transition T_1.

  The notation $*T(S, X)$ in the head of these rules, meaning that the transition T can be potentially chosen as the next transition, is used in order to avoid confusion with the notation $T(S, X, S', \tau)$ that we have introduced earlier to represent the actual application of the transition T.

- A *basic* part $\mathcal{T}_{basic}$ that determines the possible transitions (*cycle steps*) following other transitions, and consists of rules of the form

  $$*T'(S', X') \leftarrow T(S, X, S', \tau), EC(S', \tau', X'), now(\tau')$$

  which we refer to via the "name" $\mathcal{R}_{T|T'}(S', X')$. These rules sanction that, after the transition T has been executed, starting at time τ in the state S and ending at the current time τ' in the resulting state S', and the conditions EC evaluated in S' at τ' are satisfied, then transition T' could be the next transition to be applied in the state S' with the (possibly empty) input X', if required. The conditions EC are called *enabling conditions* as they determine when a cycle-step from the transition T to the transition T' can be applied. In addition, they determine the input X' of the next transition T'. Such inputs are determined by calls to the appropriate selection functions.

- A *behaviour* part $\mathcal{T}_{behaviour}$ that contains rules describing dynamic priorities amongst rules in $\mathcal{T}_{basic}$ and $\mathcal{T}_{initial}$. Rules in $\mathcal{T}_{behaviour}$ are of the form

  $$\mathcal{R}_{T|T'}(S, X') \succ \mathcal{R}_{T|T''}(S, X'') \leftarrow BC(S, X', X'', \tau), now(\tau)$$

  with $T' \neq T''$, which we will refer to via the "name" $\mathcal{P}^{T}_{T' \succ T''}$. Recall that $\mathcal{R}_{T|T'}(\cdot)$ and $\mathcal{R}_{T|T''}(\cdot)$ are (names of) rules in $\mathcal{T}_{basic} \cup \mathcal{T}_{initial}$. Note that, with an abuse of notation, T could be 0 in the case that one such rule is used to specify a priority over the *first* transition to take place, in other words, when the priority is over rules in $\mathcal{T}_{initial}$. These rules in $\mathcal{T}_{behaviour}$ sanction that, at the current time τ, after transition T, if the conditions BC hold, then we prefer the next transition to be T' over T'', namely doing T' has *higher priority* than doing T'', after T. The conditions BC are called *behaviour conditions* and give the behavioural profile of the agent. These conditions depend on the state of the agent after T and on the parameters chosen in the two cycle steps represented by $\mathcal{R}_{T|T'}(S, X')$ and $\mathcal{R}_{T|T''}(S, X'')$. Behaviour conditions are *heuristic* conditions, which may be defined in terms of *heuristic selection functions* (see [4] for details). For example, the heuristic action selection function may choose those actions in the agent's plan whose time is close to running out amongst those whose time has not run out.

- An *auxiliary part* including definitions for any predicates occurring in the enabling and behaviour conditions, and in particular for selection functions (including the heuristic ones, if needed).

- An *incompatibility part*, including rules stating that all different transitions are incompatible with each other and that different calls to the same transition but with different input items are incompatible with each other. These rules are facts of the form

  $incompatible(*T(S, X), *T'(S, X'))$

  for all T, T' such that $T \neq T'$, and of the form

  $incompatible(*T(S, X), *T(S, X')) \leftarrow X \neq X'$ expressing the fact that only one transition can be chosen at a time.

Hence, $\mathcal{T}_{cycle}$ is a logic program with priorities (P, H, A, I) where:

(i) $P = \mathcal{T}_{initial} \cup \mathcal{T}_{basic}$,

(ii) $H = \mathcal{T}_{behaviour}$,

(iii) A is the auxiliary part of $\mathcal{T}_{cycle}$, and

(iv) I is the incompatibility part of $\mathcal{T}_{cycle}$.

In the sequel, we will indicate with $\mathcal{T}^0_{cycle}$ the sub-cycle theory $\mathcal{T}_{cycle} \setminus \mathcal{T}_{basic}$ and with $\mathcal{T}^s_{cycle}$ the sub-cycle theory $\mathcal{T}_{cycle} \setminus \mathcal{T}_{initial}$. We will also assume a notion of preferential entailment $\models_{pr}$, which the underlying formalism of logic programming with priorities is equipped with. Finally, unless otherwise specified, we will assume that all cycle theories include rules that make the computee *interruptible*, as specified in [5].

Operational trace. A cycle theory $\mathcal{T}_{cycle}$ induces an *operational trace*, namely a (typically infinite) sequence of transitions
$T_1(S_0, X_1, S_1, \tau_1), \ldots, T_i(S_{i-1}, X_i, S_i, \tau_i)$, $T_{i+1}(S_i, X_{i+1}, S_{i+1}, \tau_{i+1}), \ldots$
(where each of the X_i may be empty), such that

- S_0 is the given initial state;
- for each $i \geq 1$, τ_i is given by the clock of the system, with the property that $\tau_i < \tau_{i+i}$;
- (*Initial Step*) $\mathcal{T}^0_{cycle} \wedge now(\tau_1) \models_{pr} *T_1(S_0, X_1)$;
- (*Cycle Step*) for each $i \geq 1$
  $\mathcal{T}^s_{cycle} \wedge T_i(S_{i-1}, X_i, S_i, \tau_i) \wedge now(\tau_{i+1}) \models_{pr} *T_{i+1}(S_i, X_{i+1})$
  namely each (non-final) transition in a sequence is followed by the most preferred transition, as specified by $\mathcal{T}^s_{cycle}$. If the most preferred transition determined by $\models_{pr}$ is not unique, we choose arbitrarily one.

We will refer to a state S_i in an operational trace as $\langle KB^i, Goals^i, Plan^i, TCS^i\rangle$. Note that, concerning the KB^i component of states S_i in an operational trace, only KB_0 may vary in it in the different states. We will rely upon this observation later on.

3 Improving individual welfare: basic definitions

In the sequel, given a (possibly infinite) operational trace for a computee:

$$T_1(S_0, X_1, S_1, \tau_1), \ldots, T_j(S_{j-1}, X_j, S_j, \tau_j), \ldots, T_l(S_{l-1}, X_l, S_l, \tau_l), \ldots$$

with $0 \leq j < l$, we refer to the (possibly infinite) sequence of states

$$S_0, S_1, \ldots, S_{j-1}, S_j, \ldots, S_{l-1}, S_l, \ldots$$

as the *state-sequence (of the trace)*, and to any (possibly infinite) sub-sequence

$$S_{j-1}, S_j, \ldots, S_{l-1}, S_l, \ldots$$

of a state-sequence as a *portion (of the state-sequence)*. We also refer to Σ simply as the union of all Σs corresponding to all states in the sequence or portion (note that this is equivalent to the Σ corresponding to the last state in the sequence, if this is finite).

Below, we will rely upon the informal concept of *environment-run*, namely a specific set of events (including actions by other computees) happened in the environment at given times, alongside state-sequences. We will also assume that computees have a perfect sensing capability which is capable of observing everything that happens in the environment within the specific environment run.

The following definition defines the criterium according to which we judge a state-sequence or portion as providing successive improvements over states. It is parametric wrt a notion of preference $\ll$ between states, for which sections 4 and 5 provide various alternative definitions in terms of various concepts of individual welfare. Note that this definition is somewhat naive, as, for example, it does not take into account changes of top-level goals (by the GI transition). The notion of improvement given here will be refined in section 6.

Definition 3.1 *Let $\ll$ be any notion of preference between states. Then, we say that an* infinite *state-sequence or portion* $S_0, S_1, \ldots, S_n, \ldots$ improves individual welfare wrt $\ll$ *iff for each* $j \geq 0$*, there exists* $l > j$ *such that* $S_j \ll S_l$*. We also say that a* finite *state-sequence or portion* $S_0, S_1, \ldots, S_n$ improves individual welfare wrt $\ll$ *iff for each* $j \geq 0$*,* $j < n$*, there exists* $l > j$*,* $l \leq n$ *such that* $S_j \ll S_l$*.*

Note that this definition does not impose any condition on intermediate states between S_j and S_l, and in particular any such state might actually bring the computee in a worse state than S_j, wrt $\ll$. Stronger notions could be adopted, for example that for each $j \geq 0$, for each $l > j$, $S_j \ll S_l$. However, we believe that such stronger notions would be too limiting in practice, as they would prevent computees by temporarily worsening their situation, before improving it again.

Definition 3.2 *Let $\ll$ be any notion of preference between states. Then, we say that a computee is* $\ll$-improving wrt some initial state and environment-run *iff the state-sequence corresponding to any operational trace induced by its cycle theory, from the given initial state and wrt the given environment-run, improves individual welfare wrt $\ll$, as in definition 3.1. We also say that a computee is* $\ll$-improving *(unconditionally) iff it is $\ll$-improving wrt any initial state and environment-run.*

Note that we could define a much weaker notion of (unconditional) $\ll$-improvement for a computee, requiring only for it to be $\ll$-improving wrt *some* given class of initial states and environment-runs.

4 Individual welfare in terms of top-level goals

In this section, we introduce various notions of *preference* $\ll$ between states of a computee, based upon different notions of *individual welfare*. All notions of individual welfare refer to the number of *(top-level) goals achieved* or to the number of *(top-level) unachievable goals* when the computee has reached a given state at a given time. In the remainder of this section we assume that all goals in any states are *non-reactive*, namely KB_{react} is always empty. We will see in section 6 how to extend the ideas presented here to deal with *reactive* goals as well. Note that the notions introduced in this section are somewhat naive, and will be refined in section 6.

We define the individual welfare of a computee in a given (finite) state-sequence $SS = S_0, S_1, \ldots, S_n$ $(n \geq 0)$ at a time τ in terms of the number of *(top-level) achieved goals*, referred to as $h^+(SS, \tau)$, and the number of *(top-level) unachievable goals*, referred to as $h^-(SS, \tau)$. [1] These quantities (h^+ and h^-) can be formally defined in terms of the Temporal

[1]Here, h^+ stands for *happiness* and h^- stands for *unhappiness*, as we will see later on.

Reasoning capability of the computee. For example, given a (portion of a) finite state-sequence $SS = S_0, S_1, \ldots, S_n$ and $S_i = \langle KB^i, Goals^i, Plan^i, TCS^i \rangle$, let

$$achieved(SS, \tau) = \{\langle l[t], \bot\rangle \in \bigcup_{0 \leq i \leq n} Goals^i \mid KB_{TR} \models_{TR} holds\_at(l, t) \wedge \bigcup_{0 \leq i \leq n} TCS^i \wedge \Sigma \wedge t \leq \tau\}$$
$$unachievable(SS, \tau) = \{\langle l[t], \bot\rangle \in \bigcup_{0 \leq i \leq n} Goals^i \mid \langle l[t], \bot\rangle \notin achieved(SS, \tau) \text{ and } \not\models_{\Re} \bigcup_{0 \leq i \leq n} TCS^i \wedge \Sigma \wedge t > \tau\}.$$

Then

$$h^+(SS, \tau) = \#achieved(SS, \tau)$$
$$h^-(SS, \tau) = \#unachievable(SS, \tau)$$

Note that we could adopt a different notion of unachievable goals, e.g. unachieved goals that are either timed-out or for which no plan exists.
Note also that we assume that top-level goals are all equally preferred, and that preferences amongst them are all taken care of by the Goal Decision capability.
Finally, note that we basically take a subjective view of individual welfare: goals are achieved if they can be proven subjectively by the computee via its Temporal Reasoning capability; however, it is an objective observer that establishes when to make use of this capability, from the outside, by looking at the state-sequence and the current time. So, it could happen that a computee is actually holding a goal g in S_n in SS at τ, whereas g already belongs to $achieved(SS, \tau)$ (see g_1 after T_2 in example 1). Similarly for unachievable goals. Alternatively, we could adopt a fully subjective view, whereby only at State Revision (SR) time the computee can "count" its welfare, or a fully objective view, whereby the observer evaluates whether goals are achievable or not via its own "temporal reasoning capability" with respect to its full knowledge of the environment.
The following example illustrates h^+ and h^-. Here and in the remainder of this report, if no confusion arises, we will write $h^+(S_n)$ in place of $h^+(SS, \tau_n)$, where $SS = S_0, \ldots, S_n$ and τ_i is the time at which S_i is generated.

Example 1 *Assume that a given computee is equipped with an initial pool of timed fluent literals with temporal constraints* $\{g_1, g_2, g_3, g_4, g_5\}$, *where* $g_i = l_i[t_i] \wedge TC_i$, $i, j = 1, \ldots, 5$. *Any (new variant of any) such* g_i *may be part of a top-level goal in any state in any operational trace of the computee (with the corresponding variant of the temporal constraint being part of the* TCS *component in the state).*
The following is a possible (finite portion of a) state-sequence for a given computee, starting with S_0 *with* $KB_0 = Goals = Plan = \{\}$, *with the associated values of* h^+ *and* h^-. *The time* τ_i *is the time at which transition* T_i *in the trace has generated state* S_i *from state* S_{i-1} *in the state-sequence corresponding to the trace.*

$S_0 = \langle \{\}, \{\}, \{\}, \{\} \rangle$ $\quad h^+(S_0) = 0$, $h^-(S_0) = 0$

T_1 *is GI and:*
$S_1 = \langle \{\}, \{\langle l_1[t_1], \bot\rangle, \langle l_2[t_2], \bot\rangle\}, \{\}, \{TC_1, TC_2\} \rangle$ $\quad h^+(S_1) = 0$, $h^-(S_1) = 0$

T_2 *is POI, changing* KB_0 *and leading to* g_1 *holding:*

$S_2 = \langle \_, \{\langle l_1[t_1], \bot\rangle, \langle l_2[t_2], \bot\rangle\}, \{\}, \{TC_1, TC_2\}\rangle$ $\quad h^+(S_2) = 1$
$h^-(S_2) = 0$

T_3 is PI, changing Plan (omitted as irrelevant):
$S_3 = \langle \_, \{\langle l_1[t_1], \bot\rangle, \langle l_2[t_2], \bot\rangle\}, \_, \{TC_1, TC_2\}\rangle$ $\quad h^+(S_3) = 1$
$h^-(S_3) = 0$

T_4 is SR, changing Goals (eliminating g_1 as satisfied and g_2 as timed-out) and Plan:
$S_4 = \langle \_, \{\}, \{\}, \{TC_1, TC_2\}\rangle$ $\quad h^+(S_4) = 1$
$h^-(S_4) = 1$

T_5 is GI, adding g_4:
$S_5 = \langle \{\}, \{\langle l_4[t_4], \bot\rangle\}, \{\}, \{TC_1, TC_2, TC_4\}\rangle$ $\quad h^+(S_5) = 1$
$h^-(S_5) = 1$

We give below various definitions of preference between states, in terms of h^+ and h^-.

Definition 4.1 *Given states S and S' in a (portion of a) trace of a computee, such that S comes before S' in the trace:*

- *$S \ll_1 S'$ iff $h^+(S) \leq h^+(S')$;*
- *$S \ll_2 S'$ iff $h^+(S) < h^+(S')$;*
- *$S \ll_3 S'$ iff $h^-(S) > h^-(S')$;*
- *$S \ll_4 S'$ iff $h^+(S) < h^+(S')$ or $h^+(S) = h^+(S')$ and $h^-(S) > h^-(S')$.*

Intuitively, (the designer of) a computee adopting $\ll_1$ believes that its welfare can be improved by never decreasing the number of top-level achieved goals, whereas a computee adopting $\ll_2$ wants to strictly increase the number of such goals. According to $\ll_3$, the computee wants to decrease the number of unachievable goals. Finally, according to $\ll_4$, a computee wants to strictly increase the number of achieved goals or, if this number is unchanged, at least decrease the number of unachievable goals. $\ll_4$ is a lexicographic order over pairs $(h^+(SS, \tau), h^-(SS, \tau))$. The following example illustrates the various notions of preferences, wrt the trace in example 1.

Example 2

- *Every state S_i in the trace in example 1 is better that any earlier state wrt $\ll_1$, namely, for each S_i, $i = 1, \ldots, 5$, for each S_j, $0 \leq j < i$, $S_j \ll_1 S_i$.*
- *$S_0 \ll_2 S_2$, but $S_0 \not\ll_2 S_1$.*
- *For every S_i, $i = 1, \ldots, 5$, for each S_j, $0 \leq j < i$, $S_j \not\ll_3 S_i$.*
- *$S_1 \ll_4 S_2$.*

Note that the needs of concrete applications for which computees are meant to be deployed might also need to be taken into account when specifying the notions of individual welfare and preferences over states. Such needs might be coded up into the goals of the corresponding computees in many cases (and thus our approach focussing upon goals is not overly restrictive). As an example, when designing a computee to be deployed as a book-buyer we might want for the computee to buy as many books as possible amongst those specified by the user, at the cheapest possible prices, and definitely by remaining within budget. In this setting, we might privilege states with lower numbers of books still remaining to be purchased and with higher amounts of money available.

5 Individual welfare in terms of the whole state

Up until now we have compared computees' states by looking solely at their top-level goals, and in particular at how many such goals are achieved or have become unachievable. This approach implicitly equates the computees' *happiness* to states with a high number of achieved goals and a low number of unachievable goals. This notion of happiness is rather coarse, in that it ignores the *progress* made by computees towards maximising the number of achieved goals and minimising the number of unachievable goals. This progress can be expressed in terms of all elements of $Goals$ (including non-top-level goals, belonging to partial plans for top-level goals), all elements of $Plan$ (the actions in partial plans for top-level goals) and the dynamically changing part of KB, KB_0 (the recorded changes in the environment that need to be taken into account when planning to achieve top-level goals). For example, if a goal becomes unachievable because of some event occurred in the environment, then the computee should not be deemed as non-improving, and could actually be much more effective in achieving its goals than another computee situated in a more friendly environment.

In this section we consider an alternative, finer-grained way of comparison amongst states, based upon looking at the overall computees states and the progress they make. We will refer to this comparison notion as $\prec$, to distinguish it from the various notions of happiness given in section 4 ($\ll_1, \ldots, \ll_4$). However, note that $\prec$ is another concrete instance of the generic concept of improvement $\ll$ given in section 3.

In this report, for simplicity we will define the notion of progress first with respect to a simplified scenario, where the computee is non-reactive and there are no critical deadlines for the execution of top-level actions or for the achievement of top-level goals of the computee. This scenario .....

Intuitively, absence of critical deadlines means absence of temporal constraints such as $t < 10$, $t < t' + 5$ in TCS, where t, t' is the temporal variable of some top level goal or reactive top-level action. Such constraints are however allowed when t is the time of non-top-level goals or (non-top-level reactive) actions. We will adopt the following definition for absence of critical deadlines:

Definition 5.1 (Absence of critical deadlines) *A* state $\langle KB, Goals, Plan, TCS\rangle$ *has* no critical deadline *iff TCS contains no constraint Tc with occurrences of a temporal variable t and a time constant, where t is the temporal variable of a top-level goal in $Goals$ or of a reactive top-level action in $Plan$.*
A computee *has* no critical deadline *iff every state in every operational trace induced by its cycle theory has no critical deadline.*

Note that, if a computee has no critical deadline, then $h^-(S) = \{\}$ for every state S in every operational trace induced by the cycle theory of the computee. Namely, no top-level goal of the computee can ever become unachievable.

No-reactivity for a computee means that the computee will never hold reactive goals or actions. This can be guaranteed by imposing conditions on the environment and/or the sensing capability of the computee, or otherwise by imposing that the computee holds no reactive knowledge. We will follow this last option, namely:

Definition 5.2 (No reactivity) *A computee is* non-reactive *iff* KB_{react}=$\{\}$.

In the remainder of this section, we will define the notion of progress $\prec$ for the simplified scenario with no critical deadlines and no reactivity. Future work is required to deal with the more general case.

Definition 5.3 *Given states*

$$S = \langle KB, Goals, Plan, TCS \rangle$$

and

$$S' = \langle KB', Goals', Plan', TCS' \rangle$$

in a (portion of a) trace of a computee such that S comes before S' in the trace, at times τ and τ', respectively, S' shows more progress than S, denoted $S \prec S'$, iff at least one of (a)-(d) below holds:

(a) $KB = KB'$, $Goals \subseteq Goals', Plan \subseteq Plan'$, $TCS \subseteq TCS'$,
and either $Goals \neq Goals'$ *or* $Plan \neq Plan'$ *(or both);*

(b) $Goals = Goals'$, $Plan = Plan'$, $TCS = TCS'$, *and* $KB \subset KB'$.

(c) $KB = KB'$, $Goals \supset Goals'$, *and* $Plan \supseteq Plan'$
and for every goal $G \in Goals - Goals'$ *it holds that* $KB \models^{\tau'}_{TR} G'$, *where* G' *is* G *or an ancestor of* G,
and for every action $\langle A, G, \_ \rangle \in Plan - Plan'$ *it holds that* $KB \models^{\tau'}_{TR} G'$, *where* G' *is* G *or an ancestor of* G*;*

(d) there exists a state S^* *such that* $S \prec S^*$ *and* $S^* \prec S'$.

In case (a), since we are assuming that the computee is non-reactive, the conditions

$$Goals \subseteq Goals', Goals \neq Goals'$$

amount to the following: there exists $G = \langle l[t], G' \rangle \in Goals'$ such that G' was a leaf node in *Goals*. Similarly, the conditions

$$Plan \subseteq Plan', Plan \neq Plan'$$

amount to the following: there exists $A = \langle l[t], G', \_ \rangle \in Plan' - Plan$ such that G' was a leaf node in *Goals*.

In case (b), the condition $KB \subset KB'$ amounts to $KB_0 \subset KB'_0$. This would be true, for example, if the computee has executed more actions (and thus recorded their executions in S') since state S, or if it has recorded more observations in S', through POI or AOI, since state S. Note that no information is ever deleted from the KB_0 component over an operational trace. More formally, $KB_0 \subset KB'_0$ amounts to one of more of the following:

- there exists $executed(A[T'], T) \in KB'_0 - KB_0$,
- there exists $observed(G[T'], T) \in KB'_0 - KB_0$,
- there exists $observed(C, A[T'], T) \in KB'_0 - KB_0$.

Note that an alternative, stronger definition for (b) could be obtained by imposing that

- $Goals = Goals'$, $Plan = Plan'$, and $TCS = TCS'$, and
- there exists $G = \langle l[t], G' \rangle \in Goals'$ and a total valuation σ' for the variables in S' such that $KB' \models_{TR} l[t]\sigma'$ and $\sigma' \models_{\Re} TCS'$ but
  there exists no total valuation σ for the variables in S such that $KB \models_{TR} l[t]\sigma$ and $\sigma \models_{\Re} TCS$.

We will refer to this stronger condition as (b$'$).

Case (c) occurs when between states S and S' SR has taken place. Note that the absence of critical deadlines implies that if SR has taken place then the goals of the state have necessarily changed, whereas actions may be the same. One of the advantages of state S' compared with S is that the selection functions have fewer actions/goals to consider. Note that, since we are assuming that deadlines are non-critical, goals and actions are deleted by SR if they (or some of their ancestors) are achieved.

Condition (d) is added to guarantee that $\prec$ is transitive. Indeed, conditions (a)-(c) alone do not guarantee $\prec$ to be transitive, as the following example shows.

Example 3 *Consider the following portion of a state-sequence:*

$S_0 = \langle \{\_\}, \{\langle g_1, \perp \rangle\}, \{\}, \{\_\} \rangle$
$S_1 = \langle \{\_\}, \{\langle g_1, \perp \rangle\}, \{observed(g_1, \tau_1)\}, \{\_\} \rangle$ *(after POI)*
$S_2 = \langle \{\_\}, \{\}, \{observed(g_1, \tau_1)\}, \{\_\} \rangle$ *(after SR)*

Without condition (d), $S_0 \prec S_1$ (by (b)) and $S_1 \prec S_2$ (by (c)) but $S_0 \not\prec S_2$. However, intuitively, S_2 shows more progress than S_0, as it results from the observation of a relevant and useful property of the environment.

6 Improving individual welfare: refinements

In section 3 we have assumed that any state in a state-sequence or portion of it should be taken into account to determine improvement of individual welfare. This is inappropriate when the Goal Introduction transition modifies the top-level goals in a state, as illustrated by the following example.

Example 4 *Assume to have the following (portion of a) state-sequence (with associated h^+):*

$S_0 = \langle \{\_\}, \{g_1\}, \{\_\}, \{\_\} \rangle$	$h^+(S_0) = 0$ *(g_1 not achieved yet)*
$S_1 = \langle \{\_\}, \{g_1\}, \{\_\}, \{\_\} \rangle$	$h^+(S_1) = 0$ *(g_1 not achieved yet)*
$S_2 = \langle \{\_\}, \{g_3, g_4\}, \{\_\}, \{\_\} \rangle$	$h^+(S_2) = 0$ *(g_1 dropped as non-preferred, g_3, g_4 introduced and not achieved yet)*
$S_3 = \langle \{\_\}, \{g_3, g_4\}, \{\_\}, \{\_\} \rangle$	$h^+(S_3) = 1$ *(g_1 achieved)*

According to definition 3.1, $S_0, \ldots, S_3$ is $\ll_2$-improving, which is counter-intuitive, since the computee should not be better off at achieving goals that it has dropped in favour of other, more preferred goals. Achieving g_1 might actually render g_3 unachievable, and thus should be avoided.

The notion of $\ll$-improvement should then be refined by looking at portions related to the same top-level goals. For $\ll$ seen as happiness, this can be achieved by modifying the notion of achieved and unachievable goals, as follows. Given a (portion of a) state-sequence $SS = S_0, S_1, \ldots, S_n, \ldots$, let $GI(S_i, SS)$, $0 \leq i$, be defined as follows:

- $GI(S_i, SS) = S_k$, $0 \leq k \leq n$, such that
  - either a step of Goal Introduction has occurred in the trace that has generated the state sequence, and the latest such step before S_i has occurred in state S_{k-1},
  - or $S_k = S_0$, otherwise.

Namely, $GI(S, SS)$ is the latest state in the sequence where new goals have been introduced. Below, we will refer to the top-level goals in $GI(S, SS)$ as the *most recent goals* in state S in the (portion of the) state-sequence SS, represented as $MRG(S, SS)$.

We can now re-define the notions of achieved and unachievable goals:

$$\begin{aligned}
achieved(SS,\tau) &= \{\langle l[t],\bot\rangle \in \textstyle\bigcup_{k\leq i\leq n} Goals^i \mid KB_{TR} \models_{TR} holds_at(l,t) \wedge \\
&\qquad\qquad \textstyle\bigcup_{0\leq i\leq n} TCS^i \wedge \Sigma \wedge t \leq \tau, \\
&\qquad\qquad GI(S_n, SS) = S_k\} \\
&= \{\langle l[t],\bot\rangle \in \textstyle\bigcup_{0\leq i\leq n} Goals^i \cap MRG(S_n,SS) \mid KB_{TR} \models_{TR} holds_at(l,t) \wedge \\
&\qquad\qquad \textstyle\bigcup_{0\leq i\leq n} TCS^i \wedge \\
&\qquad\qquad \Sigma \wedge t \leq \tau\}
\end{aligned}$$

$$\begin{aligned}
unachievable(SS,\tau) &= \{\langle l[t],\bot\rangle \in \textstyle\bigcup_{k\leq i\leq n} Goals^i \mid \langle l[t],\bot\rangle \notin achieved(SS,\tau) \text{ and} \\
&\qquad\qquad \not\models_{\Re} \textstyle\bigcup_{k\leq i\leq n} TCS^i \wedge \Sigma \wedge t > \tau, \\
&\qquad\qquad GI(S_n, SS) = S_k\} \\
&= \{\langle l[t],\bot\rangle \in \textstyle\bigcup_{0\leq i\leq n} Goals^i \cap MRG(S_n,SS) \mid \\
&\qquad\qquad \langle l[t],\bot\rangle \notin achieved(SS,\tau) \text{ and} \\
&\qquad\qquad \not\models_{\Re} \textstyle\bigcup_{k\leq i\leq n} TCS^i \wedge \Sigma \wedge t > \tau\}.
\end{aligned}$$

Then, h^+ and h^-can be defined via the new notions of *achieved* and *unachievable*. Wrt these new notions, in example 4, the trace $S_0, \ldots, S_3$ is not $\ll_2$-improving.

Similarly, $\prec$ should be re-defined by looking solely at states S, S' in portions related to the same top-level goals. We omit the formal definition for simplicity.

Note that other improvements of the original notions would have been possible, e.g. by considering the *ratio* between achieved and unachievable goals and most recent goals (and taking care of the cases in which there are no most recent goals appropriately). We might explore this improvement in the future.

Even with this refined notion of improvement, problems might still occur, due to the fact that, after all most recent goals have been dealt with, there is no possible further improvement and there should be none, as illustrated by the following example.

Example 5 *Assume to have the following (portion of a) state-sequence (with associated h^+):*

$S_0 = \langle\{\text{-}\}, \{g_1\}, \{\text{-}\}, \{\text{-}\}\rangle$ $\quad h^+(S_0) = 0$ *(g_1 not achieved yet)*
$S_1 = \langle\{\text{-}\}, \{g_1\}, \{\text{-}\}, \{\text{-}\}\rangle$ $\quad h^+(S_1) = 0$ *(g_1 not achieved yet)*
$S_2 = \langle\{\text{-}\}, \{\}, \{\text{-}\}, \{\text{-}\}\rangle$ $\quad h^+(S_2) = 1$ *(g_1 achieved and dropped)*
$S_3 = \langle\{\text{-}\}, \{\}, \{\text{-}\}, \{\text{-}\}\rangle$ $\quad h^+(S_3) = 1$
...
$S_9 = \langle\{\text{-}\}, \{g_3\}, \{\text{-}\}, \{\text{-}\}\rangle$ $\quad h^+(S_9) = 0$ *(g_3 introduced and not achieved yet)*

According to the refined notions, $S_0, \ldots, S_3$ is not $\ll_2$-improving, which si counter-intuitive, since the computee has achieved all its top-level goals within the sequence, and is simply idle (and will remain so until new goals are introduced, in state S_9).

We can further refine the notion of $\ll$-improvement, as follows. Here, we do so for happiness-oriented notions of $\ll$. First, let us define the notion of *idle* state S in a (portion of a) state-sequence SS, ending with S:

- $idle(S; SS)$ holds iff for each goal $G \in MRG(S, SS)$:
  - either $G \in achieved(SS, \tau)$
  - or $G \in unachievable(SS, \tau)$

  where τ is the time at which S is generated in SS.

Intuitively, an idle state is a state in which nothing can be done actively, as there is no pending goal. Then, we can re-define the notion of improvement, as follows.

Definition 6.1 *Let $\ll$ be any notion of preference between states. Then, we say that an* infinite *state-sequence or portion* $S_0, S_1, \ldots, S_n, \ldots$ improves individual welfare wrt $\ll$ *iff, for each $j \geq 0$, if it is not the case that $idle(S_j; S_0, S_1, \ldots, S_j)$, namely S_j is not idle, then there exists $l > j$ such that $S_j \ll S_l$. We also say that a* finite *state-sequence or portion* $S_0, S_1, \ldots, S_n$ improves individual welfare wrt $\ll$ *iff for each $j \geq 0$, $j < n$, if it is not the case that $idle(S_j; S_0, S_1, \ldots, S_j)$, then, there exists $l > j$, $l \leq n$ such that $S_j \ll S_l$.*

With this new notion, the portion in example 5 is $\ll_2$-improving.

Below, when referring to the notion of improvement, we will consider solely the refined notions presented in this section.

Finally, note that one additional refinement of the earlier notions would amount to taking into account reactive goals, if present, when measuring the individual welfare of computees. This would amount to including reactive goals into $MRG(S, SS)$.

7 Some cycle-independent properties

In this section, we make some initial considerations about proving that computees are $\ll$-improving (either wrt some initial state or in general), for various notions of $\ll$ given earlier. In particular, we separate out properties for the various notions of happiness-related welfare given in section 4 and the various scenaria and notions of progress-related welfare given in section 5.

7.1 Happiness

Trivially:

Theorem 7.1 *Any computee is $\ll_1$-improving.*

This result holds trivially because of the features of the KGP model, according to which goals, once achieved, can never become "unachieved". This is due to the fact that goals are existentially quantified in the KGP model.

The analogous result for $\ll_2$ does not hold as illustrated by any example where before Action Execution takes place, to execute actions that render a goal achieved, other transitions need to take place, thus rendering the actions and the corresponding goal timed-out, and thus the goal unachievable.

Again trivially, we can prove that:

Theorem 7.2 *No computee is $\ll_3$-improving.*

Theorem 7.3 *A computee is $\ll_4$-improving iff it is $\ll_2$-improving.*

Indeed, by definition, the number of unachievable goals in a state-sequence can never decrease (theorem 7.2). and thus $\ll_2$ and $\ll_4$ always coincide (theorem 7.3). If we adopted a different notion of unachievable goals, e.g. unachieved goals that are either timed-out or for which no plan exists, then $\ll_2$ and $\ll_4$ would differ, and $\ll_3$ would make sense.

We are currently attempting to identify concrete cycle theories which can be proven to be $\ll_2$-improving, or $\ll$-improving for some other notion of $\ll$, different from the ones considered earlier on. A trivial result for $\ll_2$ is that a cycle theory that introduces no goal, either because of its Goal decision capability or because it never calls the Goal Introduction transition, would be $\ll_2$-improving.

7.2 Progress

The following property links the notion of progress $\prec$ to the notion of happiness according to $\ll_1$ (in the restricted scenario we are considering), by stating that the maximal element of the $\prec$ ordering corresponds to the maximal element of the $\ll_1$ ordering, in that in a state that is maximally good according to $\prec$ as many top-level goals as possible are achieved. The same property should hold for the ordering $\ll_2$.

Conjecture 1 *If there exists a state sequence $S_0, \ldots, S_l, \ldots S_m \ldots$ such that, for every $m > l$, $S_l \not\prec S_m$, then for every $m > l$, $h^+(S_l) = h^+(S_m)$.*

We are also considering additional properties, defined in the case of *fair cycle theories*, namely cycle theories where all transitions occur at some stage in every operational trace. Then, given any state S in an operational trace induced by a fair cycle theory, in finite time we will reach a better state, according to $\prec$, if one exists, i.e. there is no infinite sequence of equally good states according to $\prec$ in any operational trace induced by a fair cycle theory. Therefore starting in any state S, we will eventually reach an optimal state—provided $\prec$ is transitive and provided the following additional conditions hold:

(i) KB_{plan} generates a finite plan for each goal.

(ii) There are no exogenous actions in the environment.

We hope to prove that there can never be an infinitely improving sequence of states given a fair cycle and conditions (i)–(ii) above.

8 Some cycle-dependent properties

This section describes some possible future work making use of the definitions given earlier in the report.

8.1 Fixed versus flexible cycles

We have made some preliminary investigations on proving that flexible cycles (induced by cycle theories) are better than given fixed cycle.
Consider a fixed cycle [2]

$$POI, SR, GI(0), PI, AE$$

and a flexible cycle [3]

$$POI, SR, GI(0), (PI \uparrow, AE \uparrow) \uparrow$$

where PI is to be repeated till there are goals to be planned for, and AE is to be repeated till there are actions that can be executed. We aim at proving that the flexible cycle is "better" than the fixed cycle at improving welfare, e.g. according to $\ll_2$.

8.2 Interruptible versus non-interruptible cycles

We could consider proving that the interruptible variant of any cycle [5] is "better" than the original non-interruptible cycle at improving welfare, e.g. according to $\ll_2$ or some other notion of $\ll$, e.g. taking into account events happened in the environment of the computee.

8.3 Profiles of behaviour

We aim at proving that some profiles of behaviour, given in a separate annex to D13. are better than others at improving welfare, according to some $\ll$, either amongst the ones given earlier or novel. For example, we believe and aim at proving that the Actively Cautious computee will do as well as the Cautious computee (but could take longer time) whereas the reverse is not true.

[2] With $T(0)$ we indicate that T is the initial transition.

[3] With $T \uparrow$, for some transition T, we indicate that T is to be repeated, if some (omitted) conditions hold, and with $R \uparrow$ we indicate that the routine (sequence of transitions) R is to be repeated.

9 Some Informal Notes on Welfare

9.1 Individual agent welfare

There is no universally accepted definition for the notion of *individual welfare* but we can informally think of it as the *state of doing well*, especially in respect to good fortune, happiness, well-being, security or prosperity. In this informal view the "state of doing well" implies a mental state and/or an external environment containing other agents and objects that ground agent's interactions, and in particular the notion that an agent is doing well (or bad for that matter).

9.2 Objective vs Subjective Welfare

Broadly speaking, in seeking to understand the notion of welfare we find two theories as the main contenders: *utilitarianism* and *freedom philosophy.* In the utilitarian view, welfare resides in how well people live. This is for each person a matter of his or her own experience of how well he or she lives. Resources, goods and so on are of different utility to different persons depending on their preferences and tastes. The balance of utility and disutility in a person's situation is pulled together by him or her in an experience of happiness. Things are good or bad for persons depending on their consequences for their happiness.

In the philosophy of freedom, welfare resides in people's freedom to pursue their own life strategies as they see them. Resources, goods and so on are instruments of choice. People live well if the life they may wish to live is available to them. What needs to be established is what opportunities people have for living as they might wish, rather than how they in fact live or choose to live. Things are good for people to the degree that they make them free and bad to the degree that freedom is restricted.

There are strong similarities between these two philosophies. Both are compatible with a rationalist view of human nature, that persons have good sense and competence to understand and pursue their own good. Both are non-dictatorial; each person is his or her own best judge of their best interest. Both are individualistic; welfare resides in persons and social welfare is some aggregate of individual welfares (not to be confused with the notion of social services to disadvantaged groups or communities). The main difference is that the utilitarian view is a theory of *subjective welfare* - people live well if they feel well - while the freedom view is a theory of *objective welfare* - people live well if they are free.

9.3 A Taxonomy of Welfare Theories

There are different ways in which we may classify theories of welfare, for one thing, the classification in subjective vs objective may not do the job for us. We give below a more fine-grained classification than that of the previous section with the aim to include what we want to do in our approach.

- Mental State Views vs. Non-Mental-State Views:
  - Mental state views: Assume some mental state that is intrinsically desirable, i.e., pleasure, happiness, or contemplation;
  - Non-Mental-State Views: One's welfare depends on more than just one's mental states, i.e., achievements, goods, or health.

- Formal vs. Substantive:
  - Substantive: A substantive theory says what things are intrinsically good for people and give reasons for preferring one state of affairs to another.
  - Formal: Formal theories specify how one finds out what things are intrinsically good for people, but they do not say what those things are (e.g. welfare as preference satisfaction).

The economics and social science literature on this is quite huge.

10 Parameters based on Agents and MAS

We are interested in identifying the important parameters that may affect individual welfare.

10.1 Agent's personalities

We have defined in [4] a set of different personalities based on different patterns of behaviours. How do these affect the welfare of the agent is an important issue. For example, an agent that is not focused may delay the achievement of its goals, which may in turn impact on the agent's individual welfare. However, to what extend existing theories of personality from psychology can be translated in our KGP model of agency, is an open issue that deserves further investigation. Moreover, to what extend is the notion of personality an adequate level of abstraction or whether addition refinements are required, such as those of *thinking styles* [3], is also an open issue.

10.2 Classification of agents

The informal definition of individual welfare provided earlier, suggests that we might find useful to classify agents into different classes whose properties are based on the notions of good fortune, happiness, well-being, or prosperity. We may call an agent as being:

- *fortunate* - if events that happen in the environment help the agent achieve its goals and more (e.g. consider an agent that buys a book on Amazon and that the transaction wins the agent a free coupon which allows the agent to exchange it with more books for free).
- *happy* - according to some definition of happiness, for example, achievement of set goals.
- *healthy* - if all its sensors, effectors, mind and body function properly (or according to their specifications).
- *prosperous* - if the agent is successful, especially economically (i.e. how many resources can access/has control over/owns including currency).
- *well-being* - if the agent is happy and/or healthy and/or prosperous.

10.3 The role of the Environment

The role of the environment is important in that it might affect the welfare of an agent. As different agent personalities will give rise to different cognitive states for agents, we would probably need to classify environments, in order to prove properties for specific agents assuming being situated in different kinds of environments.

10.3.1 An existing environment classification

Russell and Norvig in [8] provide a classification for agent environments, as follows:

- accessible vs inaccessible;
- deterministic vs non-detrministic;
- episodic vs non-episodic;
- static vs dynamic;
- discrete vs continuous;

Although this classification can be quite useful, other dimensions in the literature exist that may be related. For example, a number of articles in the literature [9, 1, 2] suggest that accessibility of an environment might be further be divided in those environment that are accessible because they are *fully observable* and those that are *partially observable*. Also, a distinction between an environment where there is a single-agent only versus environments where there are multiple agents may make a difference.

10.3.2 Social Environment classification

Russel's and Norvig's classification of agent environments does not take into consideration social notions. A more social-centric classification could give rise to social environments classification:

- private vs public (open vs closed);
- hierarchical vs flat;
- cooperative vs competitive (hostile vs amicable);
- liberal vs autocratic;
- ordered vs anarchic.

Other classifications maybe useful.

11 Conclusions

This report summarises some initial progress on the specification of properties of computees, based upon notions of individual welfare. It provides a number of definitions of individual welfare and its improvements, for (some restrictions of) the full KGP model. It also discusses a number of alternative such notions throughout.

We have made some initial attempts at formally defining various notions of individual welfare, that support the definition of operational trace improving welfare of computees, and to state and prove (or disprove) properties, e.g. whether certain computees manage to improve their welfare.

Future work includes the further specification and formal verification of such properties, as well as studying the impact of different computee profiles (cycle theories) on their welfare.

References

[1] R. Ashri, M. Luck, and M. d'Inverno. On identifying and managing relationships in multi-agent systems. In *Proceedings of the 18th International Conference of Artificial Intelligence (IJCAI-03)*, pages 743–748, 2003.

[2] A. Bracciali, P. Mancarella, K. Stathis, and F. Toni. On declarative semantics of multi-agent systems. In *Proceedings of DALT'04 Workshop*, New York, 2004.

[3] A. F. Harrison and R. M. Bramson. *The Art of Thinking*. Berkley, 1982.

[4] A. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. A logic-based approach to model computees. Technical report, SOCS Consortium, 2003. Deliverable D4.

[5] A. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. Declarative agent control. In *Proc CLIMAV*, 2004.

[6] A. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. The KGP model of agency. In *Proc. ECAI2004*, 2004. To appear.

[7] A. C. Kakas, E. Lamma, P. Mancarella, P. Mello, K. Stathis, and F. Toni. Computational model for computees and societies of computees. Technical report, SOCS Consortium, 2003. Deliverable D8.

[8] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 1995.

[9] M. Wooldridge and A. Lomuscio. A logic of visibility, perception, and knowledge: completeness and correspondence results. *Journal of the IGPL*, 9(2), March 2001.