KING'S COLLEGE LONDON.

DEPARTMENT OF INFORMATICS.

UK.

PROFESSORSHIP

BY

DR. FRANK APPIAH[1]

2020/21.

[1] This person is on a fellowship of King's College London, 2020/21.

ON

- Project title

– **Phenomenon of Procurement Agent: Reasoning and Planning**

- Proposed mentor, and host department

– **Prof. Micheal Luck: Trusted Autonomous Systems Hub, Department of Informatics.**

- Abstract

This reseach is about need by an agent to determine a generalized position of procurement enacts in an attractive interests or attractive attention. Agent parties are responsible to act by a chain of commands and that create an enactment game determined to be a win or loss.

SUPERVISOR:

Prof. Costas Illoupolous.

INTRODUCTION

This current research is mature in the sense of agents now run an engagement program via enact function. Agents now ranks their actions in any engagement function of enact. Agents are now behaving in relation to their agent interest. Agents will find dominance in their behavior relationship. Agents has an art to act in accords with other actor. Agents action is a linear process. Actors in agency with less interest value will have less attention in relationship.

This research started with the description of information and the further introduction of policy exchange information. The simple terms of policy and its exchange is composed. The investigation of policy exchange information brought out in two main purposes –Choice and Information. The price inference phenomenon is exploited in this novelty research. The term, rational exchange as an explicit specification of current inference and its exchange outcomes were enumerated.

Proposed research project

- **Aims**

- **This research project aims to undertake an intelligent autonomous agent from the perspective of procurement world. Then reason with its logically basics to create an autonomous agent.**

- **Objectives**

- **The main objective is to use a generalized enact functions that are programmed as logical functions with an enact program.This is executed to run the enact parameters against the enact facts described as engagement functions of enact. The behavior relationships of chain of commands are represented mathematically to create behavior dominance of agent interest. This creates autonomous objects.**

- **Methodology**

- **An abstract view of agent is used in representing the agent environment of procurement enact function. Mathematical logic tools are used to describe the procurement agent view.**

- **Planned Outputs**

- **A console application written in c++ called LEEMapper. A report on demonstration of execution run on enact programs.**

A PROCEDURE RUN ON ENGAGEMENT FUNCTIONS AND BEHAVIOR RELATIONSHIPS.

FRANK APPIAH.

KING' COLLEGE LONDON, CENTRE OF DOCTORAL STUDIES, ENGLAND, UNITED KINGDOM.

appiahnsiahfrank@gmail.com.

Extended Abstract[+]. A generalized enact functions are programmed as logic function with an enact program. This is executed to run the enact parameters against the enact facts described as engagement functions of enact. The behavior relationships of chain of commands are represented mathematically to create behavior dominance of agent interest.

Keywords. enactmen, agent parties, enaction, attention, enact, chain of command, dominance relation, behavior relationship, precedence relation.

Year of Study: 2013 **Year of Publication: 2020**

1 ***AFFILIATE. UNIVERSITY OF LONDON, KING'S COLLEGE LONDON, DEPARTMENT OF INFORMATICS, LONDON, UK.**

1 INTRODUCTION

The behavior relationships of the agent interest and attention is determined from the linear order of the enact functions of engagement. The engagement functions[2] of enact are:

(i) $enact\left(a_1, a_2, rank_1\right)$,

(ii) $enact\left(a_2, a_3, rank_2\right)$,

(iii) $enact\left(a_3, a_4, rank_3\right)$,

(iv) $enact\left(a_4, a_5, rank_4\right)$,

(v) $enact\left(a_n, a_{n+1}, rank_n\right)$: **Generalized Enact Functions.**

The engagement functions of enact can be read as:

(i) An enactment will run if the enact parameters are a_1 , a_2 and $rank_1$ respectively.

(ii) An enactment will run if the enact parameters are a_2 , a_3 and $rank_2$ respectively.

(iii) An enactment will run if the enact parameters are a_3 , a_4 and $rank_3$ respectively.

(iv) An enactment will run if the enact parameters are a_4 , a_5 and $rank_4$ respectively.

(v) An enactment will run if the enact parameters are a_n , a_{n+1} and $rank_n$ respectively.

2 ENGAGEMENT PROGRAM

The engagement program[1, 2, 3, 5] will run procedurally from (i) to (v). The enact function can be an assertion. That is a true or false evaluation.

PROGRAM

enact(A, B, C);

In executing the program, if the engagement functions (i) to (v) be evaluated on an enact parameter A= a_2 , B= a_4 , r= $rank_2$

then

enact (A= a_2 *, B=* a_4 *, r=* $rank_2$ *)=***FALSE**.

It is false because B value is a_3 according to enact fact (ii) and not a_4 even though A and R have correct values.

A second execution on A= a_1 , B= a_2 , r= $rank_1$:

enact(A= a_1 , B= a_2 , R= $rank_1$)=**RUN**
enact(A= a_1 , B= a_2 , R= $rank_1$)=**TRUE.**

It evaluates to true because it matches the enact fact according to the engagement function of enact(i).

A second execution on A= a_1 , B= a_2 , r= $rank_1$:

enact(A= a_1 , B= a_2 , R= $rank_1$)=**RUN**
enact(A= a_1 , B= a_2 , R= $rank_1$)=**TRUE.**

It evaluates to true because it matches the enact fact according to the engagement function of enact(i).

A third execution on A= a_1 , B= a_2 , r= $rank_4$:

enact(A= a_1 , B= a_2 , R= $rank_4$)=**RUN**
enact(A= a_1 , B= a_2 , R= $rank_4$)=**FALSE.**

It evaluates to false because it does not again match the enact fact according to the engagement function of enact(i).

A fourth execution on A= a_2 , B= a_5 , r= $rank_3$:

enact(A= a_2 , B= a_5 , R= $rank_3$)=**RUN**
enact(A= a_2 , B= a_5 , R= $rank_3$)=**FALSE.**

It evaluates to false because it does not match the enact facts according to the engagement function of enact(i) to (v). A, B and R are all mismatch for the enact database.

3 BEHAVIOR RELATIONSHIP OF AGENT INTEREST

The linearizing ordering of the chain of commands[2] is as follows:

$$a_1 \xrightarrow{R_1} a_2 \xrightarrow{R_2} a_3 \xrightarrow{R_3} a_4 \xrightarrow{R_4} a_5 \xrightarrow{R_5} a_n \xrightarrow{R_n} a_{n+1}$$

There is an implication[7] row for $a_5 \quad \rightarrow \quad a_n$ ranked at the fifth position in the linear process. The behavior relationship[6, 8] can be represented by replacing the arrow with a < (less than) relation. The behavior relationship of the agent[3] interest and attention can be as:

$$a_1 < a_2 < a_3 < a_4 < a_5 < a_n < a_{n+1} \quad .$$

The behavior relationship of the agent interest are dominance relation and behavior dominance.

Dominance Relation	**Behavior Dominance**
(i) $a_1 < a_2$	Actor 2 has interest that dominates Actor 1's interest.
(ii) $a_2 < a_3$	Actor 3 has interest that dominates Actor 2's interest.
(iii) $a_3 < a_4$	Actor 4 has interest that dominates Actor 3's interest.
(iv) $a_4 < a_5$	Actor 5 has interest that dominates Actor 4's interest.
(v) $a_n < a_{n+1}$	Actor n+1 has interest that dominates Actor n's interest.

The dominance relation of the agent's interest and attention can be read as:

(i) Actor 1's interest value is less than Actor 2's interest.

(ii) Actor 2's interest value is greater than Actor 1's interest.

(iii) Actor 1's interest value is less than Actor 4's interest.

(iv) Actor 3's interest value is greater than Actor 1's interest.

(v) Actor n's interest value is less than Actor (n+1)'s interest.

$Dom := \{A^2 : < ; a_{1,} a_2 \epsilon A\}$: Domain expression.

$DomSet := \{a_{1,} a_{2,} a_{3,} a_{4,} a_{5,} a_n , a_{n+1}\}$.

The chain of commands demands a precedence relation[3] for procurement enact. This can be written as:

Precedence Relation	**Precedence Behavior**
(i) $a_1 < a_2$	Actor 1's interest causes a precedence relation to Actor 2's interest. Actor 1 precedes before Actor 2 in

Precedence Relation	Precedence Behavior
	command.
(ii) $a_2 < a_3$	Actor 1's interest causes a precedence relation to Actor 2's interest. Actor 1 precedes before Actor 2 in command.
(iii) $a_3 < a_4$	Actor 1's with interest, a_3 precedes before Actor 2's interest.
(iv) $a_4 < a_5$	Actor 1's having interest a_4 will precede before Actor 2 with interest a_5 .
(v) $a_n < a_{n+1}$	Actor 1's *n* interest causes a precedence relation to Actor 2's *(n+1)* interest. Actor 1 precedes before Actor 2 in command.

3 CONCLUSION

In conclusively remarks, the findings of this research includes the following:

- Agents now run an engagement program via enact function.
- Agents now ranks their actions in any engagement function of enact.
- Agents are now behaving in relation to their agent interest.
- Agents will find dominance in their behavior relationship.
- Agents has an art to act in accords with other actor.
- Agents action is a linear process.
- Actors in agency with less interest value will have less attention in relationship.
- Agents will now demand precedence relation to procurement enact.
- Actor in agency with higher interest will precede without the other.
- Actor in agency having higher interest will precede before the other.

Compliance with Ethical Standards:

(In case of funding) Funding: This is research is funded by King's Alumni Group, Association of Engineering with ISAreference grant number: 204424 20821845.

Conflict of Interest:

Author, Dr. Frank Appiah declares that he has no conflict of interest .

REFERENCES

1. Appiah F. (2009/10), RuleML for Policy Exchange in Agent Commerce, King's College London, Msc Dissertation.
2. Appiah F. (2013/20), A General Position of Enactment Paradigm: Phenomenon of Procurement Enact, Centre of Doctoral Studies, King's College London, England, UK.
3. Woodridge M. (2009), Introduction to Multi-agent Systems, Wiley Publication, USA.
4. Mark Ashcraft H. and Klein R. (2010), Cognition, Pearson Education Canada.
5. Richard Stark W.(1990), Lisp, Lore, Logic, Springer Verlag, New York.
6. Ballard D. (1997), Introduction to Natural Computations, Cambridge, MA: MIT Press.
7. Andrew P. (1986), An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof, Academic Press, New York.
8. Russell S. J. (1989), The Use of Knowledge in Analog and Induction, San Mateo, CA: Morgan Kauffman.

A GENERAL POSITION OF ENACTMENT PARADIGM: PHENOMENON OF PROCUREMENT ENACT.

FRANK APPIAH.

KING' COLLEGE LONDON, CENTRE OF DOCTORAL STUDIES, ENGLAND, UNITED KINGDOM.

appiahnsiahfrank@gmail.com.

Extended Abstract[+]. There is a need by an agent to determine a generalized position of procurement enacts in an attractive interests or attractive attention. Agent parties are responsible to act by a chain of commands and that create an enactment game determined to be a win or loss.

Keywords. enactment game, agent parties, enaction, attention, enact, chain of command, PEEP, CPED, decision ratio, attractive interest.

Year of Study: 2013 **Year of Publication: 2020**

2 ***AFFILIATE. UNIVERSITY OF LONDON, KING'S COLLEGE LONDON, DEPARTMENT OF INFORMATICS, LONDON, UK.**

1 INTRODUCTION

In describing the abstract view of agent[2], it is assumed that an environment may be in any of a finite set E of discrete, instantaneous states:

$$E=\{e,e',\ldots.\}$$

Agents are assumed to have a repertoire of possible actions available to them, which transform the state of the environment. Let

$$Ac=\{\alpha,\alpha',\ldots..\}$$: Finite set of actions

The basic model of agents interacting with their environment is as follows:
A state is initiated in the environment and agent begins by choosing an action to perform on that state. As a result of action, the environment responds with a number of possible states. There is only one actual state to result. The agent chooses on a second state, the action to perform. The environment in *state-action cycle* responds with one of a set of possible states and agent then chooses another action to run and so on.

A run, of an agent in an environment is thus a sequence of interleaved environment states and actions :

$$r{:}e_0 \rightarrow^{\alpha_0} e_1 \rightarrow^{\alpha_2} e_2 \rightarrow^{\alpha_3} e_3 \rightarrow^{\alpha_1} e_4 \rightarrow^{\alpha_{n-1}} e_n$$

- Le t R be the set of all such possible finite sequence.
- R^{AC} be the subset of these that end with (over E and Ac)
- $\mathfrak{R}$ be the subset of these that end with an environment state.

2 CHAIN OF COMMAND

Agent formulates a chain of command strategy to determine the phenomenon of enaction[3] or enactment[1]. It is given as:

Chain of Commands

Actor	Command	Actor 2	Rank
a_1	enacts	a_2	1
a_2	enacts	a_3	2
a_3	enacts	a_4	3
a_4	enacts	a_5	4
a_n	enacts	a_{n+1}	n

$Rank{:}Ac_i \; x \; Ac_j \rightarrow \Omega$, with respect to decision action(Ac):= { C, D} against decision outcomes (Ω):={P, E}.

***CPED** Abbreviation:*

cooperate **:=**C *propose:*=P

defect:= D *enact:*=E

In tabulated decision form:

Outcome/Action	Cooperate	Defect
Propose	Ω_1	Ω_2
Enact	Ω_3	Ω_4

CPED Decision Ratio= Outcome over Action.

There are four decision outcomes in the command strategy. There are represented as P and E only. The possible results with incentive measure will be as:

Ω_1 *:* Propose over Cooperate = *P*

Ω_2 *:*Propose over Defect = *E*

Ω_3 *:* Enact over Cooperate *:= E*

Ω_4 *:Enact over Defect :=P*

PEEP Strategy:

1. A propose over cooperate has P attractive interest.
2. A propose over defect has E attractive interest.
3. A enact over cooperate has E attractive interest.
4. A enact over defect has P attractive interest.

In this chain of command, there are only 2 parties in the interaction, namely A_1 and A_2 with 5 attractive interests. Agent party, A_1 (Actor 1) interests can be represented as:

$$Interest\left(I_1\right)=\left\{a_1,a_2,a_3,a_4,\ldots..a_n\right\}.$$

and A_2 (Actor 2) can also have its interest to be represented as:

$$Interest\left(I_2\right)=\left\{a_2,a_3,a_4,a_5,\ldots..a_{n+1}\right\}$$

Actor 1 has only one eliminated interest from the set Actor 2 in the enactment.
The chain of command strategy will be read as:

(1) Actor 1 has a_1 interest to enact Actor 2 interest, a_2 ,

(2) Actor 1 has a_2 interest to enact Actor 2 interest, a_3 ,

(3) Actor 1 has a_3 interest to enact Actor 2 interest, a_4 ,

(4) Actor 1 has a_4 interest to enact Actor 2 interest, a_5 ,

(5) Actor 1 has a_n interest to enact Actor 2 interest, $a_{(n+1)}$.

The scenario of procurement enact creates a tension needing a negogiation strategy. What Actor 1 or 2 procures is not quite known? What is known is their interest and the chains of commands. If Actor, A_1 enacts cause an interest to Actor, A_2 , what should be done in the situation? Procurement is based on effort made by actor to do business in an agent place but the chain of commands does not need to conflict with resources available to them.

The implication[4, 6] of chain of commands are:

1. Rank 1: $a_1 \overset{enact}{\rightarrow} a_2$,
2. Rank 2: $a_2 \overset{enact}{\rightarrow} a_3$,
3. Rank 3: $a_3 \overset{enact}{\rightarrow} a_4$,
4. Rank 4: $a_4 \overset{enact}{\rightarrow} a_5$,
5. Rank n: $a_n \overset{enact}{\rightarrow} a_{n+1}$.

The implication row is a chain of commands because for each rank of enactment Actor 1 has an interest a_n that is positioned at n than Actor 2 interest which is positioned at n+1.

Simply Actor 1 interest must cause Actor 2 interest in the procurement engagement. Actor engagement is established because Actor 1 interest fits into Actor 2 interest part and they start to ran together. The implication row of interest is to attract and keep other parties in agent interest and attention. An agent interest and attention in an engagement functions properly if there is a determined place/ location with a temporal notation.

Secondly, actor engagement in procurement is established again because Actor 1 employs Actor 2 to keep an agent interest and attention. Actor 1 cannot solely do business with the agent's conflict of interest. Actor 2 cannot also do so. There is a need for engagement in enactment handshake.

The negotiation strategy is communicated to agent parties before hand in any procurement enact. The legalized position in procurement scenario demands that agent parties know of the agent's interests and attention. What interest value of Actor 1 influences which interest value of Actor 2?

3 CONCLUSION

The behavior relationships of the agent interest and attention is determined from the linear order of the enact functions of engagement. The engagement functions[5] of enact are:

(vi) $enact\left(a_1, a_2, rank_1\right)$,

(vii) $enact\left(a_2, a_3, rank_2\right)$,

(viii) $enact\left(a_3, a_4, rank_3\right)$,

(ix) $enact\left(a_4, a_5, rank_4\right)$,

(x) $enact\left(a_n, a_{n+1}, rank_n\right)$: **Generalized Enact Functions.**

The engagement functions of enact can be read as:

(vi) An enactment will run if the enact parameters are a_1 , a_2 and $rank_1$ respectively.

(vii) An enactment will run if the enact parameters are a_2 , a_3 and $rank_2$ respectively.

(viii) An enactment will run if the enact parameters are a_3 , a_4 and $rank_3$ respectively.

(ix) An enactment will run if the enact parameters are a_4 , a_5 and $rank_4$ respectively.

(x) An enactment will run if the enact parameters are a_n , a_{n+1} and $rank_n$ respectively.

Compliance with Ethical Standards:

(In case of funding) Funding: This is research is funded by King's Alumni Group, Association of Engineering with ISAreference grant number: 204424 20821845.

Conflict of Interest:

Author, Dr. Frank Appiah declares that he has no conflict of interest .

REFERENCES

1. Appiah F. (2009/10), RuleML for Policy Exchange in Agent Commerce, King's College London, Msc Dissertation.
2. Woodridge M. (2009), Introduction to Multi-agent Systems, Wiley Publication, USA.
3. Mark Ashcraft H. and Klein R. (2010), Cognition, Pearson Education Canada.
4. Richard Stark W.(1990), Lisp, Lore, Logic, Springer Verlag, New York.
5. Ballard D. (1997), Introduction to Natural Computations, Cambridge, MA: MIT Press.
6. Andrew P. (1986), An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof, Academic Press, New York.
7. Russell S. J. (1989), The Use of Knowledge in Analog and Induction, San Mateo, CA: Morgan Kauffman.

THE MAPPING OF THE LOGICAL STRUCTURE, ENACT TO THE ENGAGEMENT STRUCTURE OF ENACT.

FRANK APPIAH.

appiahnsiahfrank@gmail.com.

Extended Abstract[+]. This work researches on the fitness of logically mapping an enact structure to an engagement structure. In this mapping, a language of logical(L) engage(E) enactment(E) is described in the context of both logic and structures, LEE. An underscore language is created based on words like implies and is, An is_ language and implies_language of logical engage enactment are made in four sentences.

Keywords. enactment, composites, environment, abbreviation, model, logic, theory, language, sentence.

1 INTRODUCTION

The engagement function[1] of enact has its structure: $enact(a_i, a_j, rank_i)$ where $i, j > 1$. The parameters of this structure takes two interest variables and one rank indicator. The logical structure of enact is represented as:

enact(a, l, t) , where a: action, l:location and t: time.

The exact mapping of engagement, enact to a logical enact is represented as:

$$\begin{array}{cccc} \text{enact(} & a_i, & a_j, & rank_i\text{).} \\ & \downarrow & \downarrow & \downarrow \\ \text{enact(} & \text{a,} & \text{l,} & \text{t).} \end{array}$$

Here, the interest a_i maps to action a, interest a_j also maps l and $rank_i$ maps to time, t. The physical dimensions of the structures does not work properly. This research looks at ways to addressing the issue by a language[6] in the context of engagement. The language of logical engage enactment describes the context of both logical an engagement structures:

(a) $enact_E \to enact_L$,

(b) $a_i \to a$,

(c) $a_j \to l$,

(d) $rank_i \to t$.

The language of the logical engage-enactment is:

(i) The enact of business implies the enact of machinery, partonomy, interest, attention and more logic.

(ii) The enact of business interest implies the enact of business action or event.

(iii) The enact of business interest implies the enact of interest in a location.

(iv) The enact rank of business interest implies the enact of the linear/temporal ordering of interest.

The *is_language* of the logical engage enactment(LEE^{is}) is:

(i) The enact of business is the enact of logic.

(ii) The enact of business interest is the implication of business action.

(iii) The enact of business interest is in a location.

(iv) The enact of business rank is based on temporal dimensions.

The *implies_ language* of LEE($LEE^{\to}$) is

(i) $Enact_E$ implies $Enact_L$

(ii) Interest a_i implies Action a

(iii) Interest a_j implies Action in location l

(iv) Rank interest $rank_i$ implies Ordering in temporal dimension t.

The linear ordering of the mapping LEE structures is as follows:

$$enact_E \to enact_L \to a_i \to a \to a_j \to l \to rank_i \to t \quad .$$

Given $\alpha \to \beta$ can be abbreviated to $\neg\alpha \vee \beta$.

The LEE structures (a) to (d) can be abbreviated as:

(a) $enact_E \to enact_L$ abbreviates to $\neg enact_E \vee enact_L$

(b) $a_i \to a$ abbreviates to $\neg a_i \vee a$.

(c) $a_j \to a$ abbreviates to $\neg a_j \vee a$.

(d) $rank_i \rightarrow t$ abbreviates to $\neg rank_i \vee t$.

The LEE structures now becomes:

(a) $enact_E \Leftrightarrow enact_L$

(b) $a_i \Leftrightarrow a$

(c) $a_j \Leftrightarrow l$

(d) $rank_i \Leftrightarrow t$

Given $\alpha \Leftrightarrow \beta$ can be abbreviated to $(\alpha \Leftrightarrow \beta) \vee (\beta \vee \alpha)$.

The double arrow LEE($LEE^{\Leftrightarrow}$) structures now becomes:

(a) $enact_E \Leftrightarrow enact_L$ abbreviates to:

$(enact_E \rightarrow enact_L) \quad \wedge \quad (enact_L \rightarrow enact_E)$.

(b) $(a_i \Leftrightarrow a)$ abbreviates to $(a_i \rightarrow a) \wedge (a \rightarrow a_i)$.

(c). $(a_j \Leftrightarrow a)$ abbreviates to $(a_j \rightarrow a) \wedge (a \rightarrow a_j)$.

(d) $rank_i \Leftrightarrow t$ abbreviates to $(rank_i \rightarrow t) \wedge (t \rightarrow rank_i)$.

The $LEE^{\rightarrow}$ structure can be read as:

(i) $enact_E \xrightarrow{\vec{r}} enact_L$: A run(r) of an agent in an engaged environment is thus a sequence of inter-logic environment of states and actions.

(ii) $a_i \xrightarrow{\vec{r}} a_j$: A run of an agent by a sequence of inter-logic environment of states and actions in an engaged environment.

(iii) $a_j \xrightarrow{\vec{r}} L$: A run of an agent by sequence of inter-logic states and actions is localized in an engaged environment.

(iv) $rank_i \xrightarrow{\vec{r}} t$: A state is initially ranked in an engaged environment before choosing an action to execute (run in a specific time).

The $LEE^{\Leftrightarrow}$ abbreviation structures can also be read as:

(1) $enact_E \Leftrightarrow enact_L$: An $enact_E$ will exist to be true implies $enact_L$ is true and $enact_L$ existence is true implies $enact_E$ is true.

(2) $(a_i \Leftrightarrow a)$: An interst a_i will exist to be true implies a_i is true and a hold to be true in existence implies a_i is existentially true.

(3) $(a_j \Leftrightarrow a)$: An interest enaction , a_j is existentially true implies location exist to be true and location, l is very true in existence implies interest holding is truly in existence.

(4) $rank_i \Leftrightarrow t$: A rank for interest will exist true implies a chosen priority in time holds to be true.

A run on LEE structure will consequent the following:

(D) $enact_E \rightarrow enact_L \rightarrow a_i$

(DI) $enact_L \rightarrow a_i \rightarrow a$

(DII) $a_i \rightarrow a \rightarrow a_j$

(DIII) $a \rightarrow a_j \rightarrow l$

(DIV) $a_j \rightarrow l \rightarrow rank_i$

(DV) $l \rightarrow rank_i \rightarrow t.$.

2 RESULTS OF WORK

The formulas of propositional enactment[1, 2] consist of:

(1) *propositional constants*; T and F.

(2) *propositional variables*; a , a_i , a_j , l , t , $rank_i$, $enact_E$ and $enact_L$.

(3) *propositional composites*;

Composites	Propositional
or-Composites	(i) $\neg enact_E \vee enact_L$ (ii) $\neg a_i \vee a$ (iii) $\neg a_j \vee l$ (iv) $\neg rank_i \vee t$
not-Composites	(i) $\neg enact_E$

Composites	Propositional
	(ii) $\neg a_i$ (iii) $\neg a_j$ (iv) $\neg rank_i$
and-composites	(i) $(a_i \rightarrow a) \wedge (a \rightarrow a_i)$ (ii) $(a_j \rightarrow l) \wedge (l \rightarrow a_j)$ (iii) $(enact_E \rightarrow enact_L) \wedge (eanct_L \rightarrow enact_E)$ (iv) $(rank_i \rightarrow t) \wedge (t \rightarrow rank_i)$
implies-composites	(i) $enact_E \rightarrow enact_L$ (ii) $a_i \rightarrow a$ (iii) $a_j \rightarrow l$ (iv) $rank_i \rightarrow t$

$\vee$ -composites are V-clauses of disjunctive literals. $\wedge$ - composites are $\wedge$ clauses of conjunctive literals. The literal is a logical constant or the negation of a constant or variable. Enactment logic[1] is the term for the formulas of the propositional enactment.

4 CONCLUSION

This section concludes work on research done in terms of LEE language. LEE is a logical engage enactment in abbreviation and a coined name[4]. The engagement function[4] is made from inspection of enact function[1] and its structural parameters.

Engagement function is a structure that takes variables of two interests and one rank. An exact mapping of engagement enact to logical enact is represented in this research. A four sentences of *is_* language is written to bring normal meaning to the whole language process. Again, four sentences of *implies_* language is also written.

A linear ordering of the LEE structure maps are generated. LEE , $LEE^{\rightarrow}$ and $LEE^{\Leftrightarrow}$ abbreviations are generated with logical implication. A run is carried out on each LEE structures. In concluding remarks, four propositional composites of literal are made from enact and engagement functions.

Compliance with Ethical Standards:

(In case of funding) Funding: This is research is funded by no organisation or institution.

Conflict of Interest:

Author, Frank Appiah declares that he has no conflict of interest .

REFERENCES

8. Appiah F. (2009/10), RuleML for Policy Exchange in Agent Commerce, King's College London, Msc Dissertation.
9. Appiah F. (2020). A general position of enactment paradigm; Phenomenon of Procurement Agent. KCL Centre of Doctoral Studies, London, England, UK.
10. Richard Stark W.(1990), Lisp, Lore, Logic, Springer Verlag, New York.
11. Ballard D. (1997), Introduction to Natural Computations, Cambridge, MA: MIT Press.
12. Andrew P. (1986), An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof, Academic Press, New York.
13. Russell S. J. (1989), The Use of Knowledge in Analog and Induction, San Mateo, CA: Morgan Kauffman.

SUPPLEMENTARY MATERIAL

LEE Run- (L)ogical (E)ngage (E)nactment Program : Demonstration of Enactment Logic in Console Application.

(LEEMapper Program)

by

Frank Appiah
appiahnsiahfrank@gmail.com

This work describes a console application on the fitness of logically mapping an enact structure to an engagement structure. In this mapping, a language of logical(L) engage(E) enactment(E) is described in the context of both logic and structures, LEE. An underscore language is created based on words like implies and is, An is_ language and implies_language of logical engage enactment are made in four sentences. The created LEE language is programmatically described as console character outputs from a C++ program execution or run.

Terms. console, output, environment, character, language, enactment, c++, sentence.

Console Character Outputs

LEE Run- (L)ogical (E)ngage (E)nactment Program

LEE Language and Composites
Enactment Formalizations
Developed by: Frank Appiah

Enter[ok] to start>>

LEE Language and Composites
Enactment Formalizations

Developed by: Frank Appiah

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

===========================

Enactment Logic Solution Menu

===========================

(1):: Context Language
(2):: LEE Implications
(3):: LEE Imply Language
(4):: LEE Linear Ordering
(5):: LEE Abbreviation(Abb)
(6):: LEE AbbStruct Language
(7):: LEE Abbreviation Language
(8):: LEE Double Arrow(DA)
(9):: LEE DA Structures
(10):: LEE is_Language
(11):: LEE Consequent Run
(12):: Enactment Formalizations

Building the Enactment Engagement Language.....

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

1. Structural Language Context(SLC)

(1) Enact_E ---> Enact_L ::=Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) ---> Enact_L(sell_pc,buy_pc,rank_9)

(2) a_i ---> a_j ::= buy--->sell_pc

(3) a_j ---> l ::= sell_pc --->tottenham

(4) rank_i ---> t ::= rank_9 --->Sat 12/09/2019 12:00pm

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

Contextual Language for LEE

(i)The enact of business implies the enact of machinery, partonomy, interest, attention and more logic.

(ii)The enact of business interest implies the enact of business action or event.

(iii)The enact of business interest implies the enact of interest in a location.

(iv)The enact rank of business interest implies the enact of the linear/temporal ordering of interest.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

2. LEE Implies(LEE->)

(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) implies Enact_L(sell_pc,buy_pc,rank_9)

(2)buy implies sell_pc

(3)sell_pc implies tottenham

(4)rank_9 implies Sat 12/09/2019 12:00pm

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

3. Contextual Language for LEE Implications

(i)The enact of business is the enact of logic.

(ii)The enact of business interest is the implication of business action.

(iii)The enact of business interest is in a location.

(iv)The enact of business rank is based on temporal dimensions.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

4. LEE Implies Language(LEE->)

(1)Enact_E(buy,tottenham,Sat1 2/09/2019 12:00pm) implies Enact_L (sell_pc,buy_pc,rank_9)

(2) Interest buy implies Action sell_pc

(3) Interest sell_pc implies Action in Location tottenham

(4) Rank interest rank_9 implies Ordering in temporal dimension Sat 12/09/2019 12:00pm

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

5. Contextual Language for LEE Implications

(i)The enact of business is the enact of logic.

(ii)The enact of business interest is the implication of business action.

(iii)The enact of business interest is in a location.

(iv)The enact of business rank is based on temporal dimensions.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

6. LEE Mapping Structure(LEE-Linear Ordering)

Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9) --> buy --> sell_pc --> sell_pc --> tottenham --> rank_9 --> Sat 12/09/2019 12:00pm

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

7. LEE Abbreviation(LEEAbb)::{LEE Implies are now abbreviated below:}

(1) not Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm)

or Enact_L(sell_pc,buy_pc,rank_9)

(2) not buy or sell_pc

(3) not sell_pc or tottenham

(4) not rank_9 or Sat 12/09/2019 12:00pm

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

LEE Structural Language(LEEStruct Language)

(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9)

Reading

::A run(r) of an agent in an engaged environment is thus a sequence of inter-logic environment of states and actions.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

(2)buy --> sell_pc

Reading

::A run of an agent by a sequence of inter-logic environment of states and actions in an engaged environment.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

(3)sell_pc --> tottenham

Reading

::A run of an agent by sequence of inter-logic states and actions is localized in an engaged environment.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

(4)rank_9 --> Sat 12/09/2019 12:00pm

Reading

::A state is initially ranked in an engaged environment before choosing an action to execute (run in a specific time).

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

LEE Abbreviation Language(LEEAbb Language)

(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) <==> Enact_L(sell_pc,buy_pc,rank_9)

Reading

::An Enact_E will exist to be true implies Enact_L is true and Enact_L existence is true implies Enact_E is true.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

(2)buy <==> sell_pc

Reading

::An interest a_i will exist to be true implies a_i is true and a hold to be true in existence implies a_i is existentially true.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

(3)sell_pc <==> tottenham

Reading

::An interest enaction, a_i is existentially true implies location exist to be true and location,l is very true in existence implies interest holding is truly in existence.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

(4)rank_9 <==> Sat 12/09/2019 12:00pm

Reading

::A rank for interest will exist true implies a chosen priority in time holds to be true.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

8. LEE Double Arrow(LEE<==>)

(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) <==> Enact_L(sell_pc,buy_pc,rank_9)

(2)buy <==> sell_pc

(3)sell_pc <==> tottenham

(4)rank_9 <==> Sat 12/09/2019 12:00pm

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

9. LEE Double Arrow Structures(LEE<-->)

(1) (Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9)) and (Enact_L(sell_pc,buy_pc,rank_9) --> Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm))

(2) (buy --> sell_pc) and (sell_pc --> buy)

(3) (sell_pc --> tottenham) and (tottenham --> sell_pc)

(4) (rank_9 ---> Sat 12/09/2019 12:00pm) and (Sat 12/09/2019 12:00pm ---> rank_9)

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

10. LEE is_LanguageContextual Language for LEE_is

(i) The enact of business is the enact of logic.

(ii) The enact of business interest is the implication of business action.

(iii) The enact of business interest is in a location.

(iv) The enact of business rank is based on temporal dimensions.

Enter number[go(1)/no(2)] to continue>>

Go Value:=#

11. LEE Consequent Runs(LEE-runs)

D::Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9) --> buy
Di::Enact_L(sell_pc,buy_pc,rank_9) --> buy --> sell_pc
Dii::sell_pc --> buy_pc --> buy

Diii::sell_pc --> buy_pc --> tottenham
Div::buy_pc --> tottenham --> rank_9
Dv::tottenham --> rank_9 --> Sat 12/09/2019 12:00pm

Enter number[go(1)/no(2)] to continue>>
Go Value:=#

12. LEE Propose Enactment (Enactment Formulas)

Propositional Constants::
T and F

Propositional Variables::

Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) , Enact_L(sell_pc,buy_pc,rank_9) , buy , sell_pc , sell_pc , tottenham , rank_9 , Sat 12/09/2019 12:00pm

Enter number[go(1)/no(2)] to continue>>
Go Value:=#

Propositional Composites::

or-Composites

(i) not Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) or Enact_L(sell_pc,buy_pc,rank_9)
(ii) not buy or sell_pc
(iii) not sell_pc or tottenham
(iv) not rank_9 or Sat 12/09/2019 12:00pm

not-Composites

(i) not Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm)

(ii) not buy

(iii) not sell_pc

(iv) not rank_9

and-Composites

(i) (Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9)) and (Enact_L(sell_pc,buy_pc,rank_9) --> Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm))

(ii) (buy --> sell_pc) and (sell_pc --> buy)

(iii) (sell_pc --> tottenham) and (tottenham --> sell_pc)

(iv) (rank_9 --> Sat 12/09/2019 12:00pm) and (Sat 12/09/2019 12:00pm --> rank_9)

implies-Composites

(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9)

(2)buy --> sell_pc

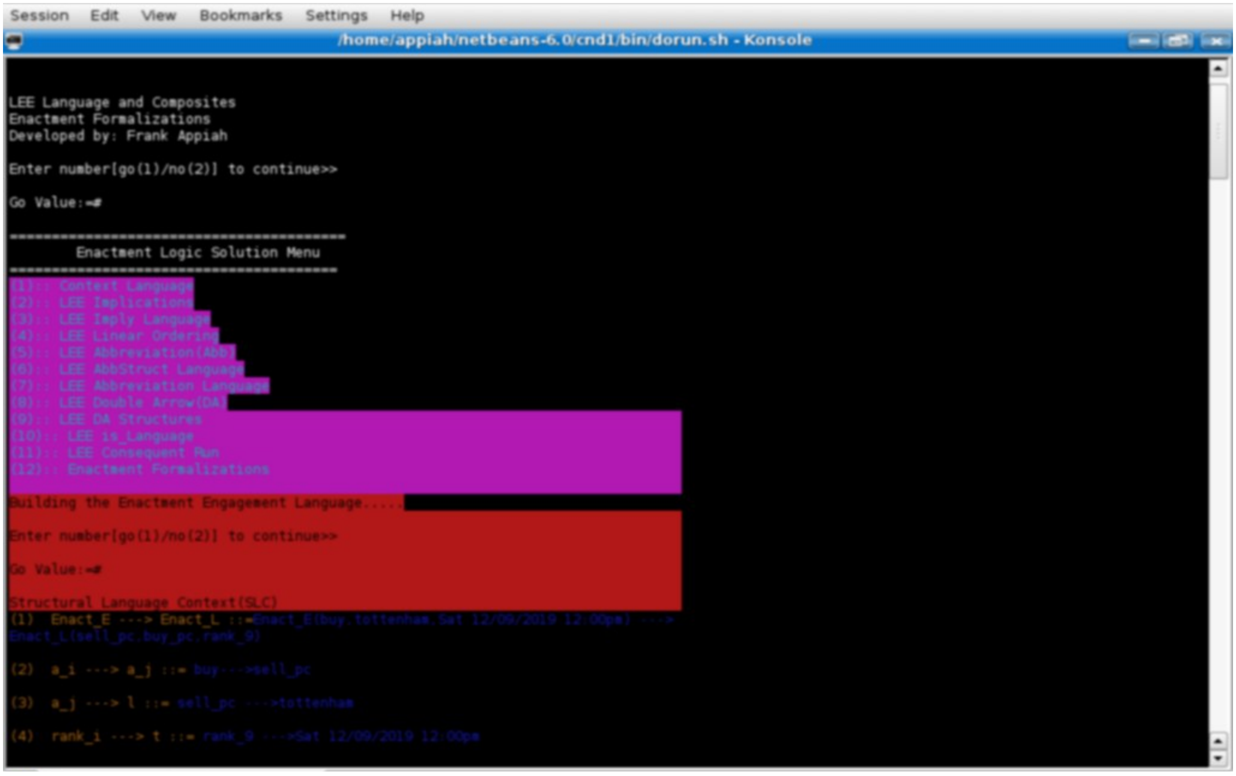
(3)sell_pc --> tottenham

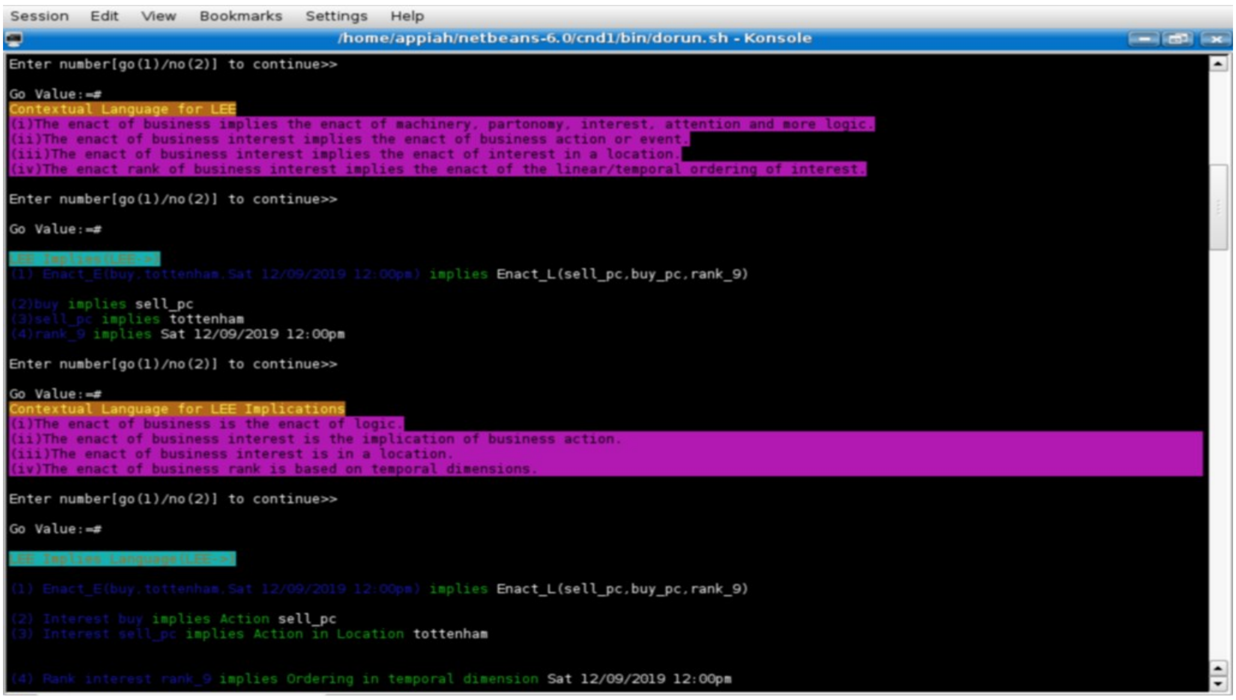
(4)rank_9 --> Sat 12/09/2019 12:00pm

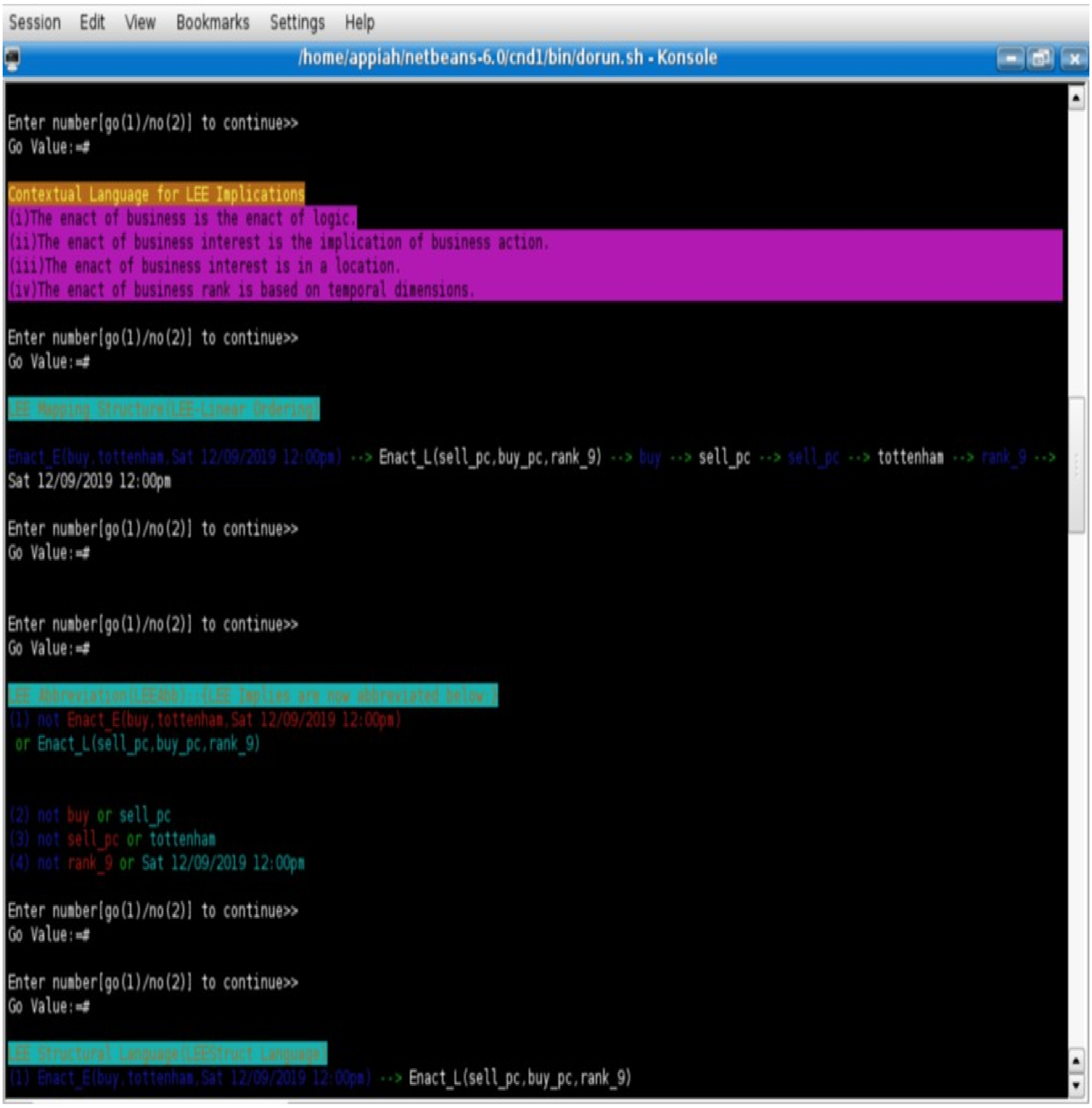
Enter number[go(1)/no(2)] to continue>>

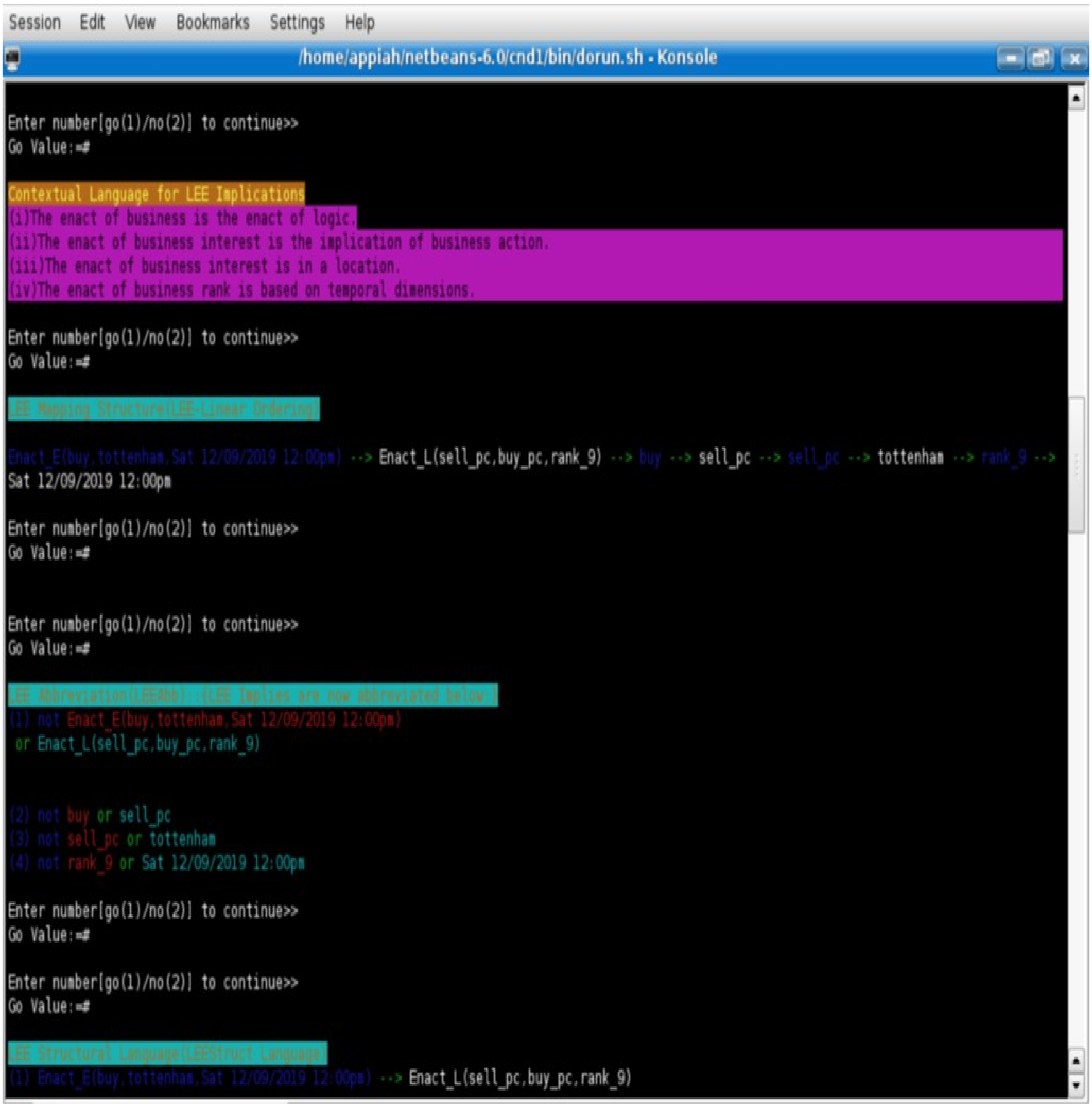
Program Exits.

ScreenShots on LEEMapper Console Application

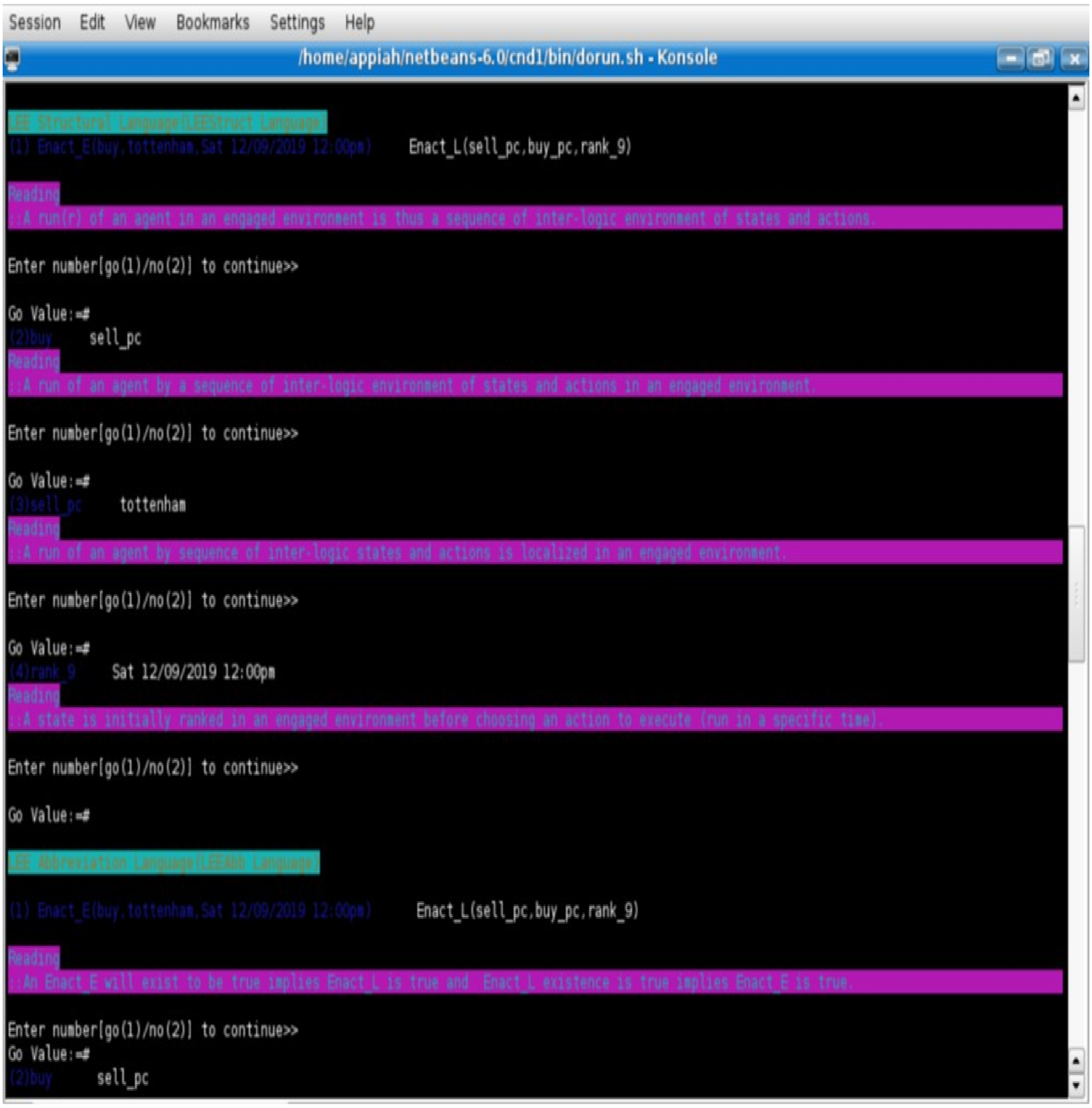


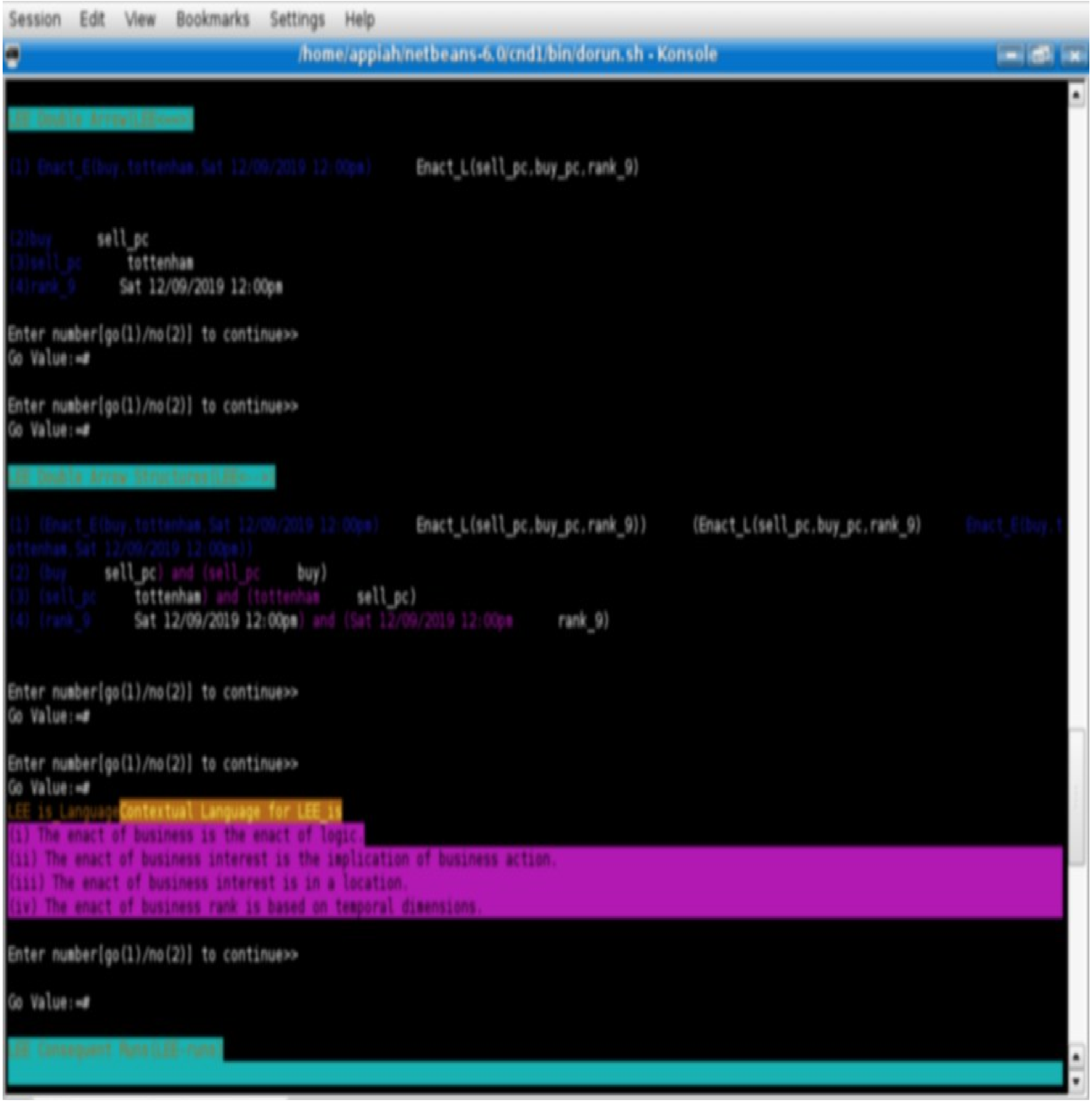


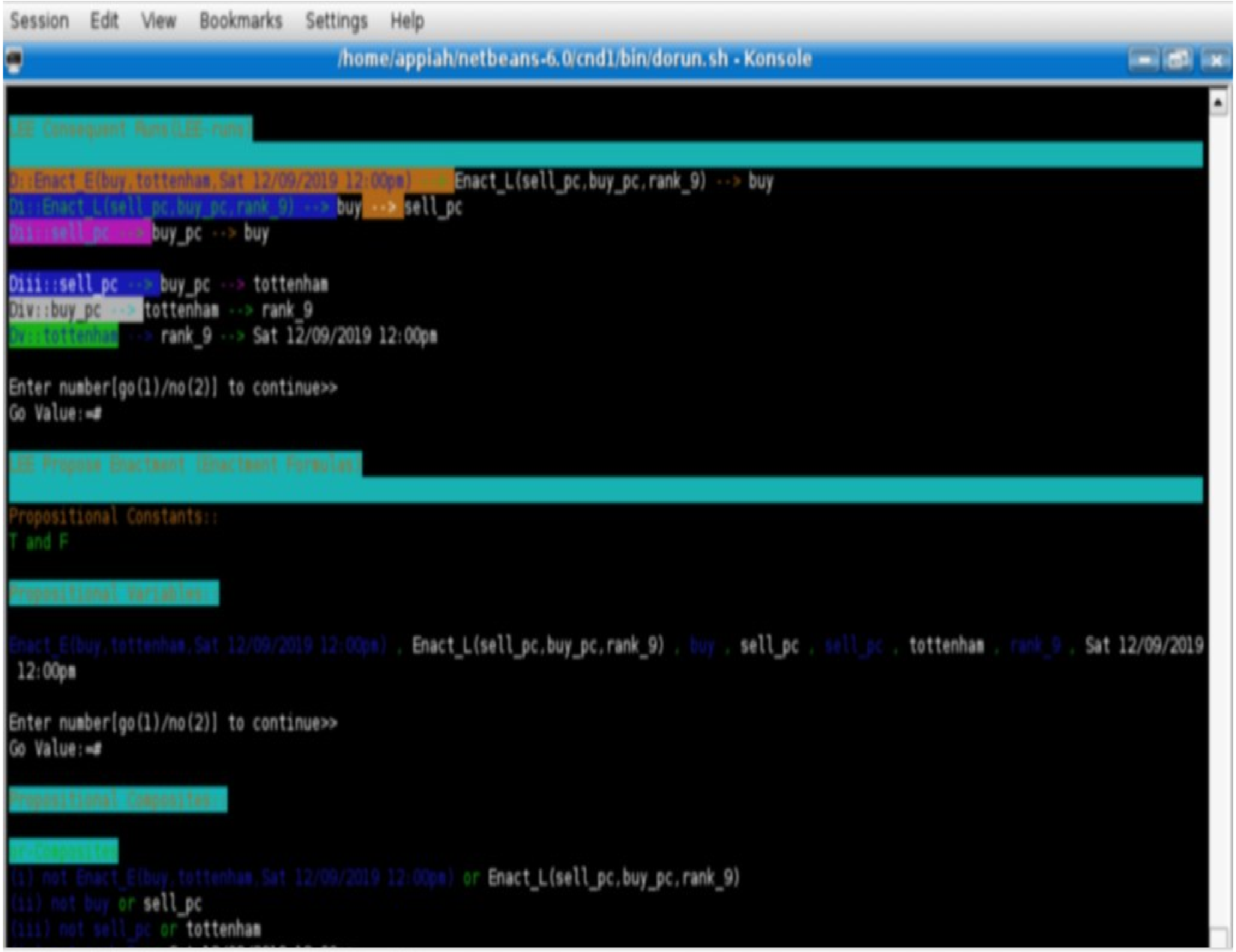

Session Edit View Bookmarks Settings Help
/home/appiah/netbeans-6.0/cnd1/bin/dorun.sh - Konsole
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Contextual Language for LEE Implications
(i)The enact of business is the enact of logic.
(ii)The enact of business interest is the implication of business action.
(iii)The enact of business interest is in a location.
(iv)The enact of business rank is based on temporal dimensions.
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
LEE Mapping Structure(LEE-Linear Ordering)
Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9) --> buy --> sell_pc --> sell_pc --> tottenham --> rank_9 -->
Sat 12/09/2019 12:00pm
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
LEE Abbreviation(LEEAbb):(LEE Implies are now abbreviated below)
(1) not Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm)
or Enact_L(sell_pc,buy_pc,rank_9)
(2) not buy or sell_pc
(3) not sell_pc or tottenham
(4) not rank_9 or Sat 12/09/2019 12:00pm
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
LEE Structural Language(LEEStruct Language)
(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9)

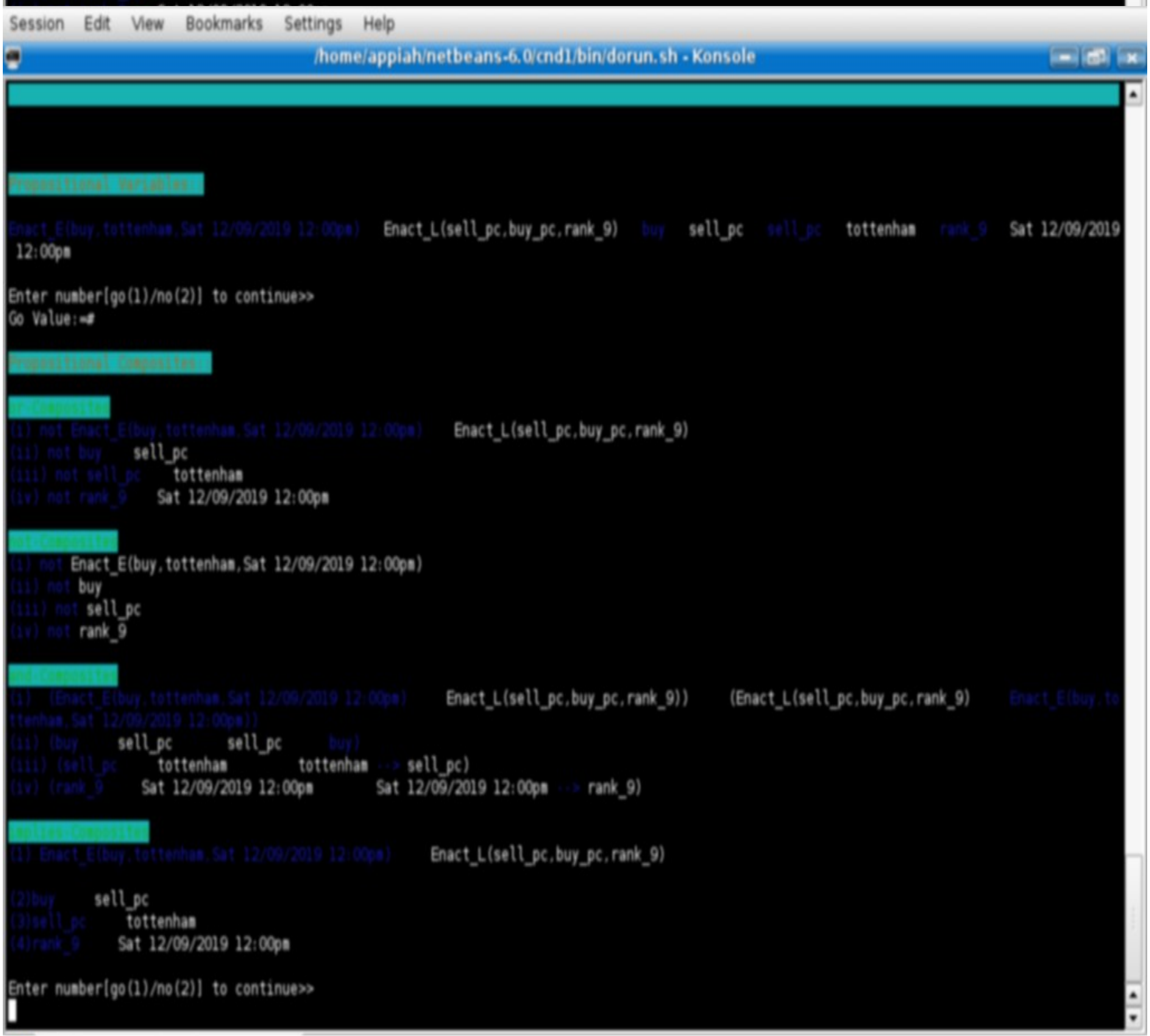


Session Edit View Bookmarks Settings Help
/home/appiah/netbeans-6.0/cnd1/bin/dorun.sh - Konsole
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Contextual Language for LEE Implications
(i)The enact of business is the enact of logic.
(ii)The enact of business interest is the implication of business action.
(iii)The enact of business interest is in a location.
(iv)The enact of business rank is based on temporal dimensions.
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
LEE Mapping Structure(LEE-Linear Ordering)
Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9) --> buy --> sell_pc --> sell_pc --> tottenham --> rank_9 -->
Sat 12/09/2019 12:00pm
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
LEE Abbreviation(LEEAbb): (LEE Implies are now abbreviated below)
(1) not Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm)
or Enact_L(sell_pc,buy_pc,rank_9)
(2) not buy or sell_pc
(3) not sell_pc or tottenham
(4) not rank_9 or Sat 12/09/2019 12:00pm
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
LEE Structural Language(LEEStruct Language)
(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9)


Session Edit View Bookmarks Settings Help
/home/appiah/netbeans-6.0/cnd1/bin/dorun.sh - Konsole
LEE Structural Language(LEEStruct Language)
(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) Enact_L(sell_pc,buy_pc,rank_9)
Reading
::A run(r) of an agent in an engaged environment is thus a sequence of inter-logic environment of states and actions.
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
(2)buy sell_pc
Reading
::A run of an agent by a sequence of inter-logic environment of states and actions in an engaged environment.
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
(3)sell_pc tottenham
Reading
::A run of an agent by sequence of inter-logic states and actions is localized in an engaged environment.
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
(4)rank_9 Sat 12/09/2019 12:00pm
Reading
::A state is initially ranked in an engaged environment before choosing an action to execute (run in a specific time).
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
LEE Abbreviation Language(LEEAbb Language)
(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) Enact_L(sell_pc,buy_pc,rank_9)
Reading
::An Enact_E will exist to be true implies Enact_L is true and Enact_L existence is true implies Enact_E is true.
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
(2)buy sell_pc


Session Edit View Bookmarks Settings Help
/home/appiah/netbeans-6.0/cnd1/bin/dorun.sh - Konsole
Enact_L(sell_pc,buy_pc,rank_9)
sell_pc
tottenham
Sat 12/09/2019 12:00pm
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Enact_L(sell_pc,buy_pc,rank_9))
(Enact_L(sell_pc,buy_pc,rank_9)
sell_pc) and (sell_pc buy)
tottenham) and (tottenham sell_pc)
Sat 12/09/2019 12:00pm) and (Sat 12/09/2019 12:00pm rank_9)
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Contextual Language for LEE is
(i) The enact of business is the enact of logic.
(ii) The enact of business interest is the implication of business action.
(iii) The enact of business interest is in a location.
(iv) The enact of business rank is based on temporal dimensions.
Enter number[go(1)/no(2)] to continue>>
Go Value:=#


Session Edit View Bookmarks Settings Help
/home/appiah/netbeans-6.0/cnd1/bin/dorun.sh - Konsole
D::Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) --> Enact_L(sell_pc,buy_pc,rank_9) --> buy
Di::Enact_L(sell_pc,buy_pc,rank_9) --> buy --> sell_pc
Dii::sell_pc --> buy_pc --> buy
Diii::sell_pc --> buy_pc --> tottenham
Div::buy_pc --> tottenham --> rank_9
Dv::tottenham --> rank_9 --> Sat 12/09/2019 12:00pm
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
Propositional Constants::
T and F
Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) , Enact_L(sell_pc,buy_pc,rank_9) , buy , sell_pc , sell_pc , tottenham , rank_9 , Sat 12/09/2019 12:00pm
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
(i) not Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) or Enact_L(sell_pc,buy_pc,rank_9)
(ii) not buy or sell_pc
(iii) not sell_pc or tottenham


Session Edit View Bookmarks Settings Help
/home/appiah/netbeans-6.0/cnd1/bin/dorun.sh - Konsole
Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) Enact_L(sell_pc,buy_pc,rank_9) buy sell_pc sell_pc tottenham rank_9 Sat 12/09/2019 12:00pm
Enter number[go(1)/no(2)] to continue>>
Go Value:=#
(i) not Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) Enact_L(sell_pc,buy_pc,rank_9)
(ii) not buy sell_pc
(iii) not sell_pc tottenham
(iv) not rank_9 Sat 12/09/2019 12:00pm
(i) not Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm)
(ii) not buy
(iii) not sell_pc
(iv) not rank_9
(ii) (buy sell_pc sell_pc buy)
(iii) (sell_pc tottenham tottenham --> sell_pc)
(iv) (rank_9 Sat 12/09/2019 12:00pm Sat 12/09/2019 12:00pm --> rank_9)
(1) Enact_E(buy,tottenham,Sat 12/09/2019 12:00pm) Enact_L(sell_pc,buy_pc,rank_9)
(2)buy sell_pc
(3)sell_pc tottenham
(4)rank_9 Sat 12/09/2019 12:00pm
Enter number[go(1)/no(2)] to continue>>