A Policy with Updated Information State and Dialogue Acts

1. <ask_question_to_usr>
Information state:
[Agent/BEL: {NULL}]
[Agent/DES: {nonconfusion(usr)}]
[Agent/AGENDA: <notify confusion(usr)>]
[Public/QUD: <?a.wordproblem(agent)>]
[Public/COM: {State A1}]
[Public/LU: {Speaker: agent;
Move: {answer(wordproblem(usr))}]
Dialogue Act: {notify confusion(usr)}

2. <ask_question_to_usr_A1>
Information state 2.1:
[Agent/BEL: {State A1(usr), tohelp(agent)}]
[Agent/DES: {nonconfusion(usr), tohelp(agent)}]
[Agent/AGENDA:
<restate(wordproblem(agent))>]
[Public/QUD: <?a.wordproblem(agent)>]
[Public/COM: {State A1}]
[Public/LU:
{Speaker: usr; Move: {cofirm(help)}]
Dialogue Act 2.1: {confirmation(usr)}

Information state 2.2:
[Agent/BEL: {State A1(usr), tohelp(agent)}]
[Agent/DES: {nonconfusion(usr), tohelp(agent)}]
[Agent/AGENDA:
<inforExten(wordproblem(agent))>]
[Public/QUD: <?a.wordproblem(agent)>]
[Public/COM: {State A1}]
[Public/LU: {Speaker: agent;
Move: {restate(wordproblem(agent))}]
Dialogue Act 2.2: {restatement(agent)}

A.1 Simulating a Multimodal Dialogue

3. <ask_question_to_usr_insufficient>
Information state:
[Agent/BEL: {State A1(usr), tohelp(agent)}]
[Agent/DES: {nonconfusion(usr), tohelp(agent)}]
[Agent/AGENDA:
<inforSply(wordproblem(agent))>]
[Public/QUD: <?a.wordproblem(agent)>]
[Public/COM: {State A1}]
[Public/LU: {Speaker: usr;
Move: {inforExten(wordproblem(agent))}]
Dialogue Act: {information extension(agent)}

4. <ask_question_to_usr_complex>
Information state:
[Agent/BEL: {State A1(usr), tohelp(agent)}]
[Agent/DES: {nonconfusion(usr), tohelp(agent)}]
[Agent/AGENDA:
<cofirm(answer(wordproblem)),
affirm(wordproblem(agent))>]
[Public/QUD: <?a.wordproblem(agent)>]
[Public/COM: {State A1}]
[Public/LU: {Speaker: agent; Move: {inforSply(wordproblem(agent))}]
Dialogue Act: {information supplement(agent)}

5. <ask_question_to_usr_complex_2>
Information state:
[Agent/BEL: {State A2(usr), tohelp(agent)}]
[Agent/DES: {nonconfusion(usr), tohelp(agent)}]
[Agent/AGENDA: <downdate(wordproblem))>]
[Public/QUD: <?a.wordproblem(agent)>]
[Public/COM: {State A2}]
[Public/LU: {Speaker: usr;
Move: {cofirm(answer(wordproblem)),
affirm(wordproblem(agent))}]
Dialogue Act: {feedbackrequest(usr),
affirm(agent)}

6. <ask_question_to_usr_state A2>
Information state:
[Agent/BEL: {State A2(usr), tohelp(agent)}]
[Agent/DES: {nonconfusion(usr), tohelp(agent)}]
[Agent/AGENDA:
<notify confusion(usr), ack(answer(urs))>]
[Public/QUD: <?a.wordproblem(agent)>]
[Public/COM: {State A2}]
[Public/LU: {Speaker: agent; Move: {downdate(wordproblem))}]
Dialogue Act: {notify confusion(usr)}

7. <ask_question_to_usr_state B>
Information state:
[Agent/BEL: {State B(usr)}]
[Agent/DES: {nonconfusion(usr)}]
[Agent/AGENDA: <NULL>]
[Public/QUD: <?a.wordproblem(agent)>]
[Public/COM: {State B}]
[Public/LU:
{Speaker: usr; Move: {ack(answer(urs))}]
Dialogue Act: {notify confusion(usr)}