UNIVERSITY OF LEICESTER

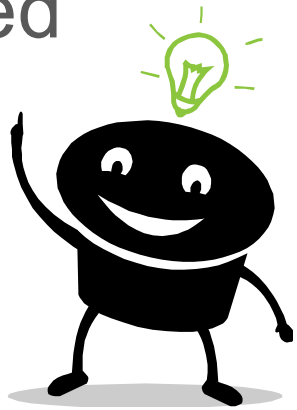
A Graph-based Game to Negotiate Features

Graphs as Models – ETAPS 2016
Eindhoven, The Netherlands

Mohammed Alabdullatif and Reiko Heckel
University of Leicester

Problem

- Service selection and configuration is a key challenge in today's business (e-commerce or B2B).
- Service requestors and providers have different interests and preferences.
- Competition and conflicting interests between requestors and providers, but also shared interests.
- Negotiation allows to explore both.

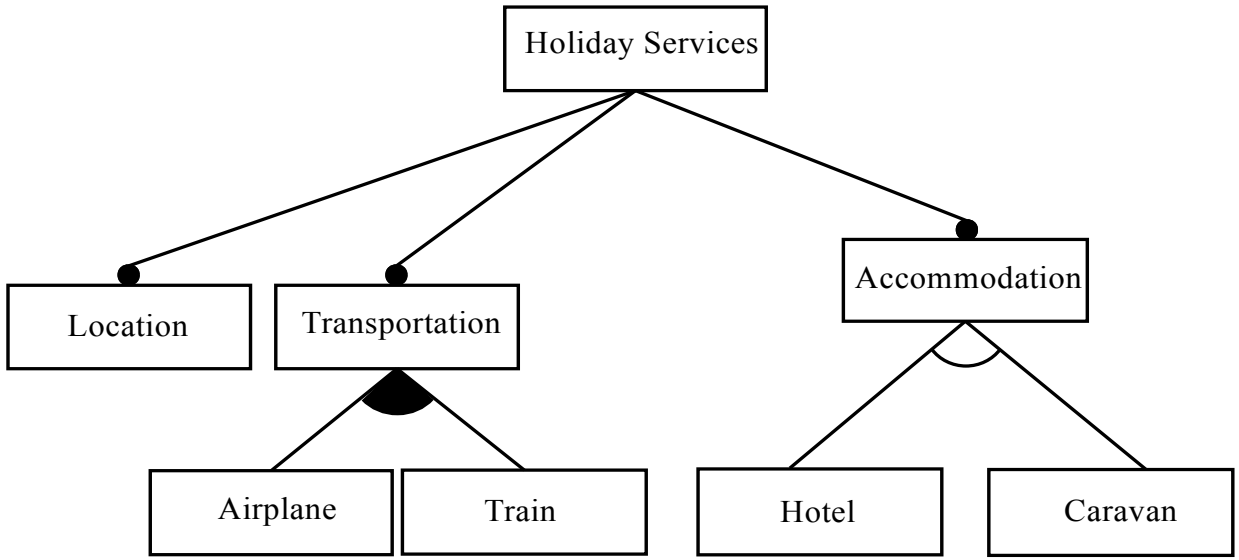


Feature Negotiation as Graph Transformation Game

Negotiation games based on feature models:

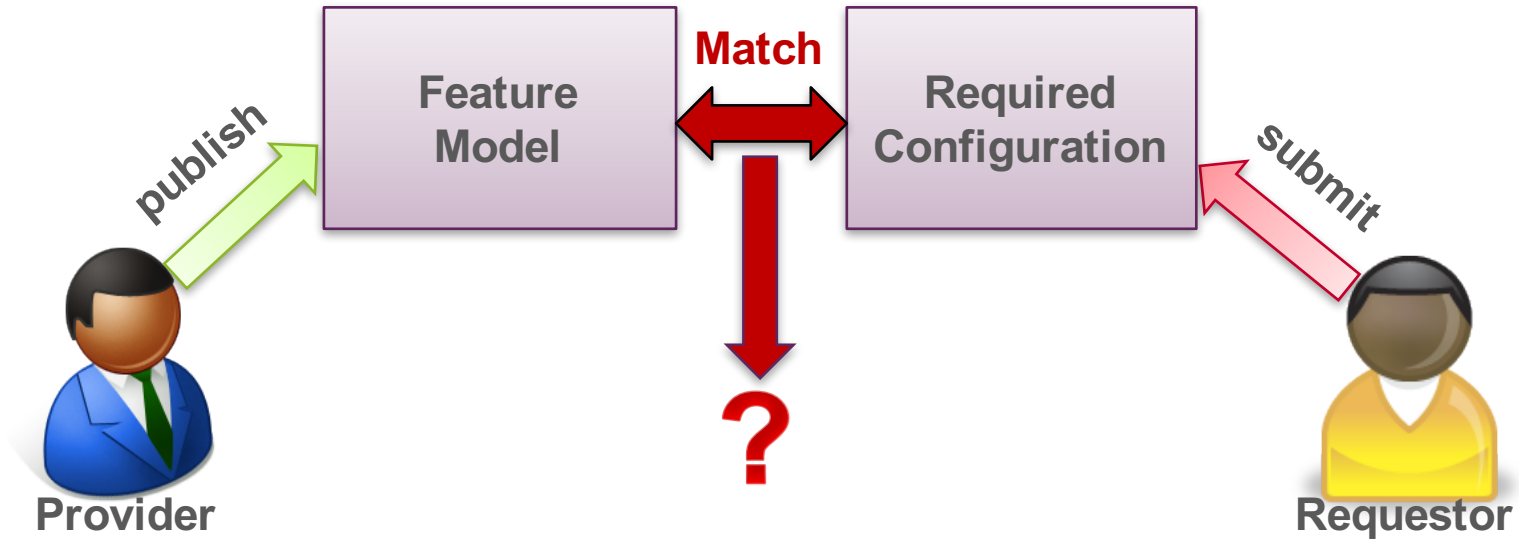
- Feature Models are used to model service selection and configuration.
- Games are used to design and analyse competitive and cooperative behaviour between.

Implemented as Graph Transformation Games.



Feature Negotiation

- Negotiation based on selecting and deselecting features representing service characteristics.
- Start with configuration matching requirements and offers.
- Proceed by proposing, accepting, rejecting change requests.



UNIVERSITY OF LEICESTER

Negotiation Game

For every match, play a game where

- **Negotiators** are the players of the game.
- **Interests** are measured by a payoff function.
- **Negotiation Protocol** defines the rules of the game.
- **Negotiation Strategy** determines the actions of the negotiators at any stage of the game.
- **Nash Equilibria** give a set of strategies such that no player can get better payoff by switching to another strategy.

Negotiation Game

We assume

- Two players *Pro*, *Req*.
- A feature model *FM* over set of features *F*.
- An initial feature configuration *C* over *FM*.
- Sets of available strategies S_{pro}, S_{req} determining final configuration.
- Payoff functions U_{Pro}, U_{Req} defined by assigning values for *price*, *cost* and *utility* to all features. Payoffs are calculated based on features in final configuration
  - Provider U_{Pro} : sum of (*price – cost*)
  - Requestor U_{Req} : sum of (*utility – price*)

Graph Transformation Game

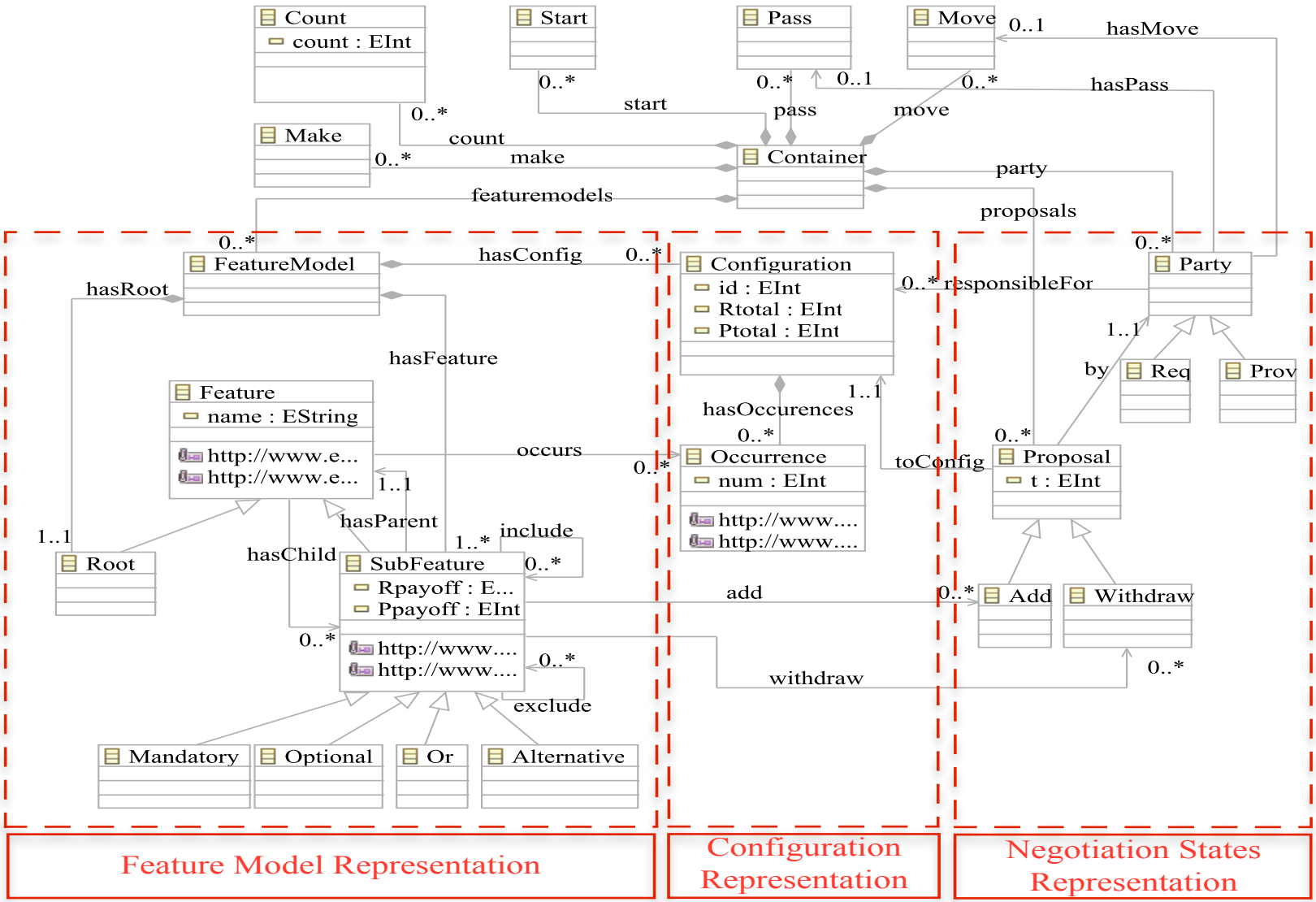
States of the game given by graphs representing

- The feature model.
- The current Configuration.
- The negotiation state (open requests for features to be (de)selected, next turn, etc.)

Moves of the players are defined by graph transformation rules.

- It is analysed as a two-player turn-based zero-sum stochastic game.

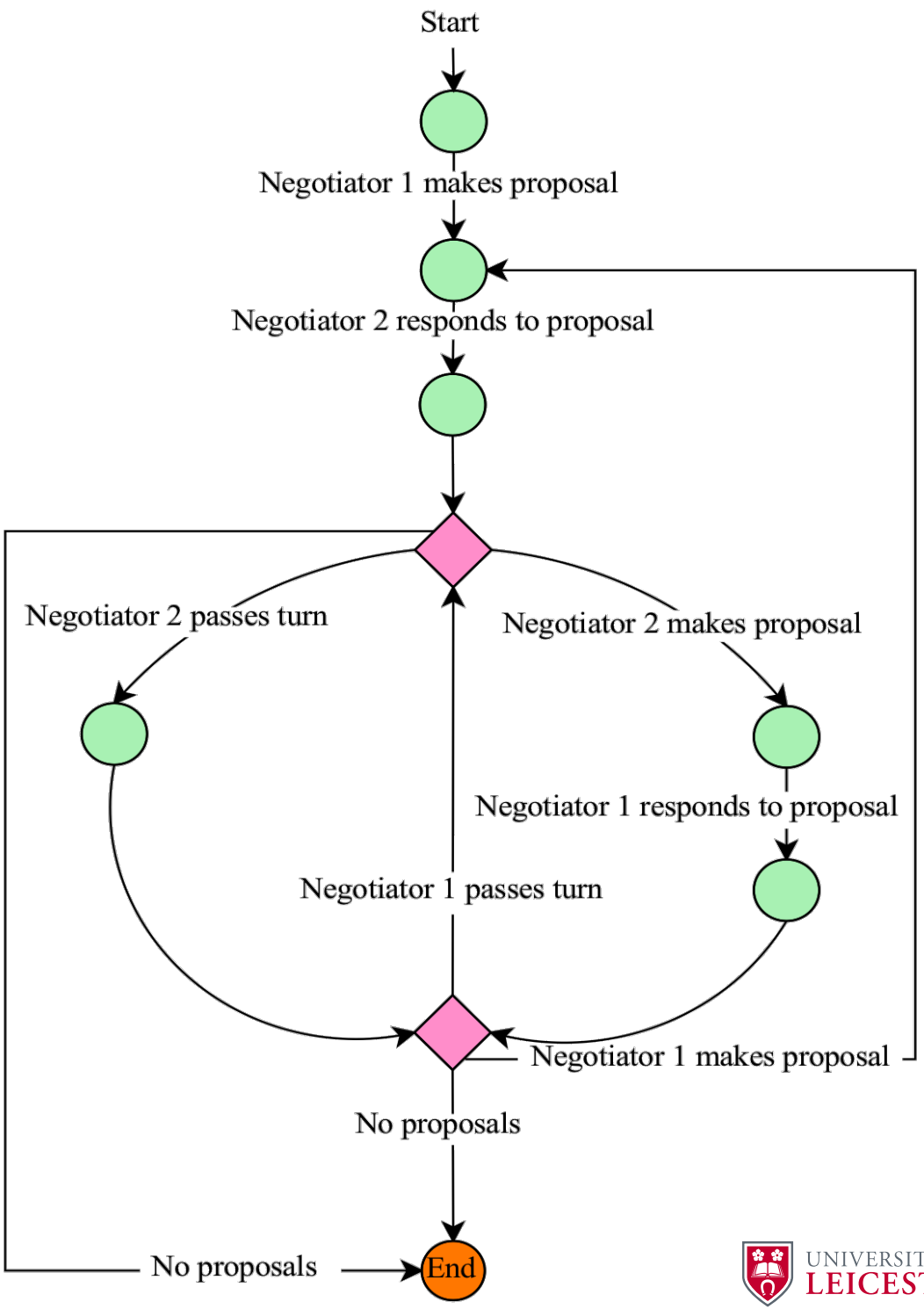
Type Graph



UNIVERSITY OF LEICESTER

Rules

Are designed so players take turns proposing changes, accepting or rejecting them.



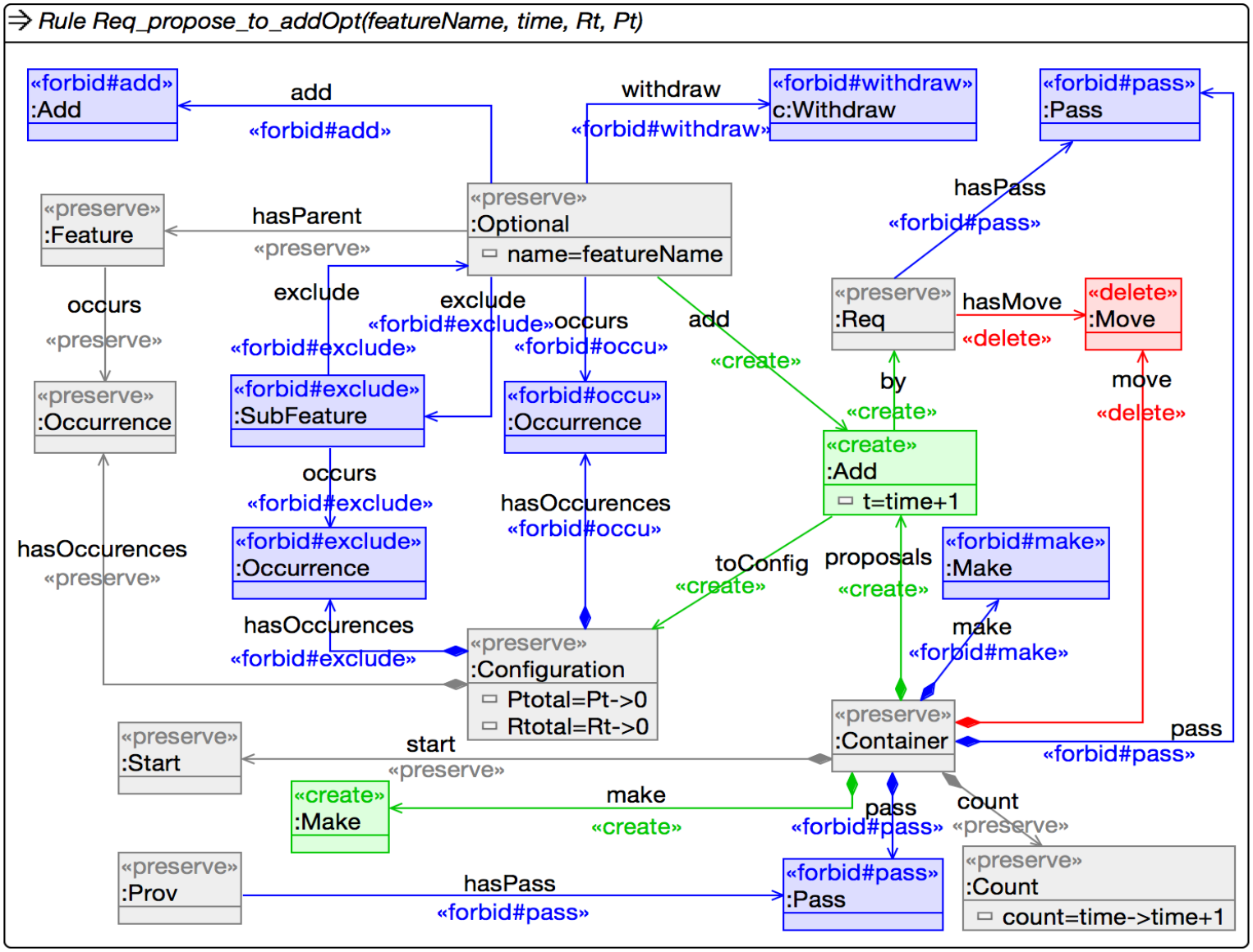
Rules: for each player

Five rules for making proposals:

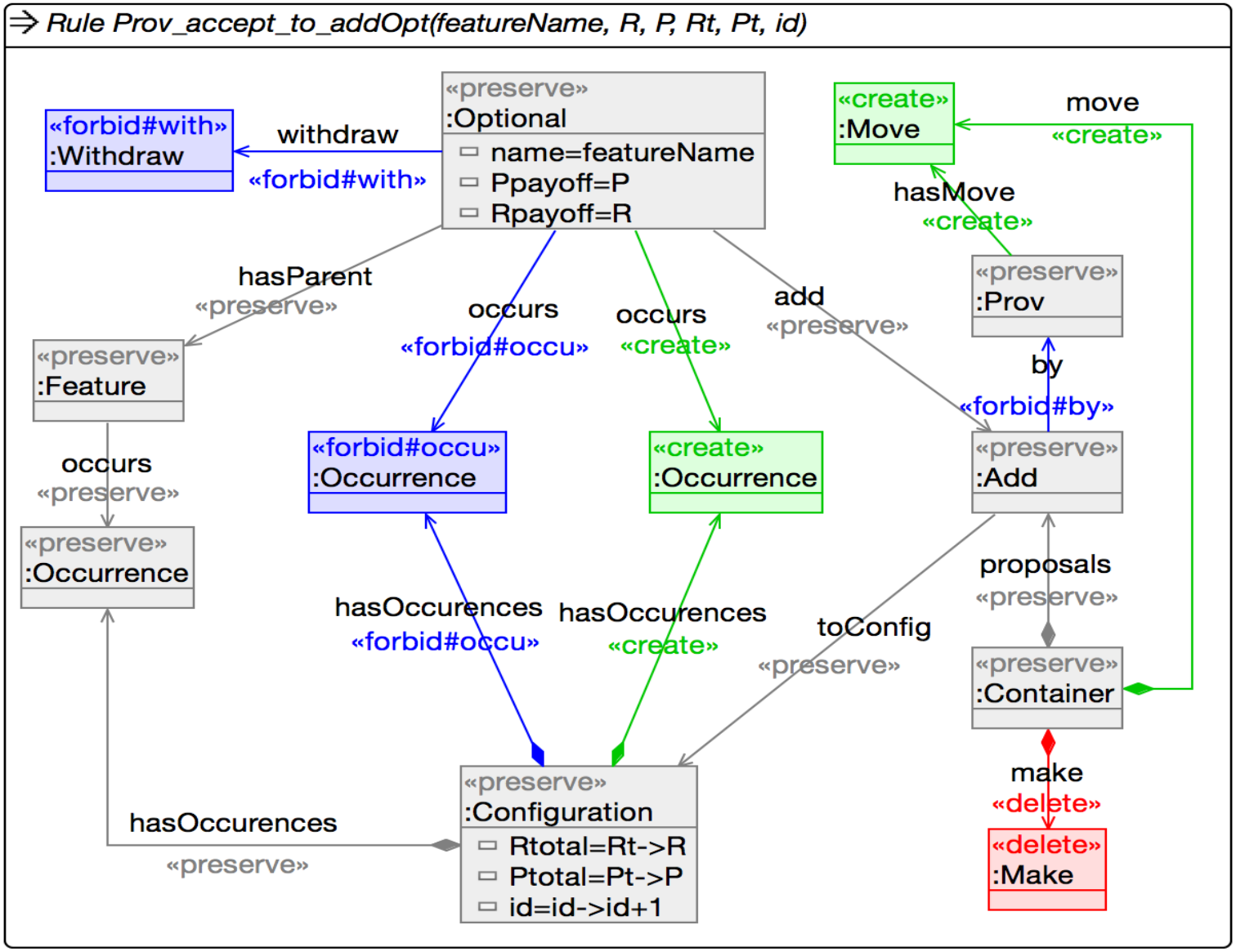
1. Propose to add Optional feature.
2. Propose to add Or feature.
3. Propose to withdraw Optional feature.
4. Propose to withdraw Or feature.
5. Propose to substitute Alternative feature.

Ten Rules for responding to such proposals by accepting or rejecting them.

In Henshin: Propose to Add Optional



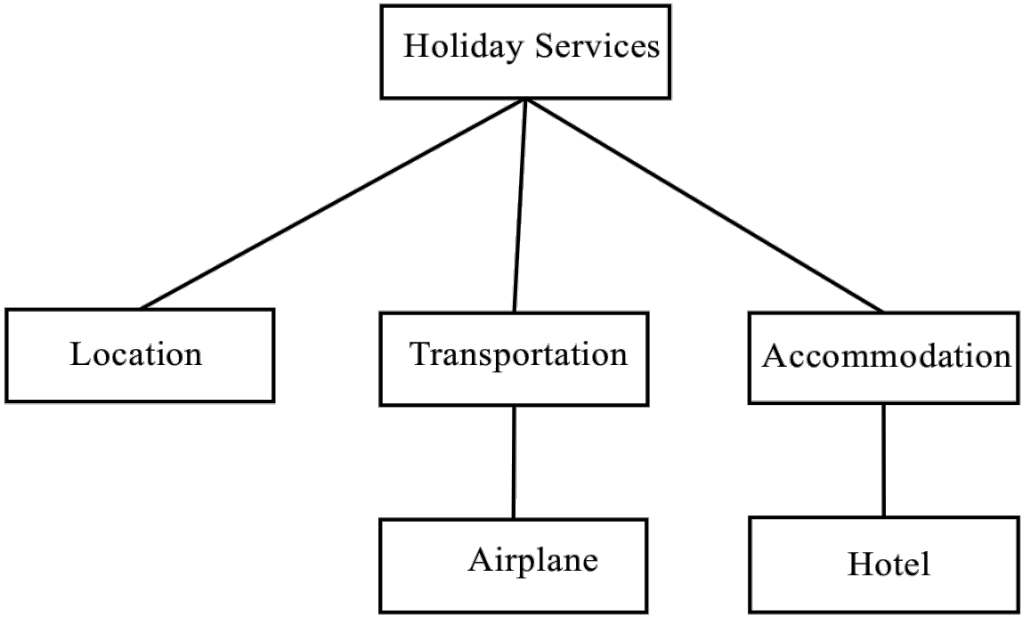
In Henshin: Accept to add Optional



Generating the Game

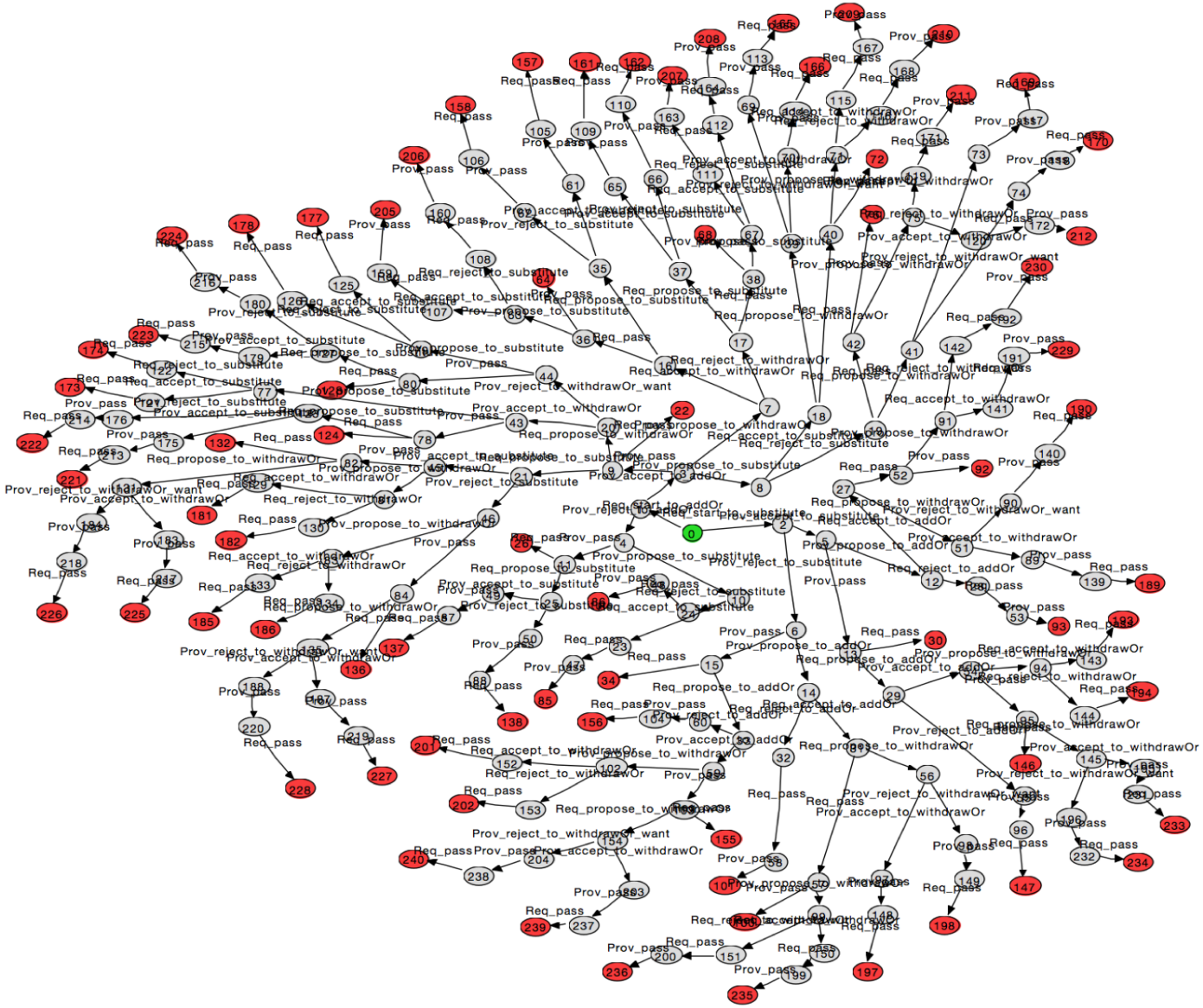
Using Henshin State Space tools we

- specify the initial state as an instance of our metamodel to encode the initial configuration



- generate the state space.

State Space by Henshin



UNIVERSITY OF LEICESTER

Analysing the Game

- Negotiators can behave erratically (to be less predictable or because their individual preferences are unknown)
- Hence the game is seen as a two-player, turn-based stochastic game.
- We use PRISM-games, based on Markov decision processes, to create
  - Players: provider and requestor.
  - Modules: transitions defining behaviour of each player.
  - Rewards: payoff values of the features confirmed in each state.

PRISM-games SMG from Henshin LTS

```
smg

player requestor
[Req_start_to_addOr], [Req_start_to_substitute],
[Req_propose_to_withdrawOr_Not_want],
[Req_propose_to_substitute_Not_want],
[Req_propose_to_addOr_Not_want],
          [Req_accept_to_withdrawOr_want],
[Req_reject_to_withdrawOr_want], [Req_accept_to_withdrawOr_Not_want],
[Req_reject_to_withdrawOr_Not_want], [Req_accept_to_substitute_want],
[Req_reject_to_substitute_want], [Req_accept_to_substitute_Not_want],
[Req_reject_to_substitute_Not_want], [Req_accept_to_addOr_want],
[Req_reject_to_addOr_want], [Req_accept_to_addOr_Not_want],
[Req_reject_to_addOr_Not_want], [Req_propose_to_substitute_want],
[Req_propose_to_withdrawOr_want]
endplayer
player provider
[Prov_accept_to_addOr_want], [Prov_reject_to_addOr_want],
[Prov_accept_to_addOr_Not_want], [Prov_reject_to_addOr_Not_want],
[Prov_accept_to_substitute_want], [Prov_reject_to_substitute_want],
[Prov_accept_to_substitute_Not_want],
[Prov_reject_to_substitute_Not_want],
[Prov_propose_to_withdrawOr_want], [Prov_propose_to_substitute_want],
[Prov_propose_to_addOr_want], [Prov_accept_to_withdrawOr_want],
[Prov_reject_to_withdrawOr_want], [Prov_accept_to_withdrawOr_Not_want],
[Prov_reject_to_withdrawOr_Not_want],
[Prov_propose_to_substitute_Not_want],
[Prov_propose_to_withdrawOr_Not_want]
endplayer
module M

          s : [0..128] init 0;
          [Req_start_to_addOr] (s=0) -> (s'=1);
          [Req_start_to_substitute] (s=0) -> (s'=2);
          [Prov_accept_to_addOr_want] (s=1) -> (s'=3);
          [Prov_reject_to_addOr_want] (s=1) -> (s'=4);
          [Prov_accept_to_addOr_Not_want] (s=1) -> (s'=5);
          [Prov_reject_to_addOr_Not_want] (s=1) -> (s'=6);
          [Prov_accept_to_substitute_want] (s=2) -> (s'=7);
          [Prov_reject_to_substitute_want] (s=2) -> (s'=8);
          [Prov_accept_to_substitute_Not_want] (s=2) -> (s'=9);
          [Prov_reject_to_substitute_Not_want] (s=2) -> (s'=10);
          [Prov_propose_to_withdrawOr_want] (s=3) -> (s'=11);
          [Prov_propose_to_substitute_want] (s=3) -> (s'=12);
          [Prov_propose_to_substitute_want] (s=4) -> (s'=13);
          [Req_propose_to_withdrawOr_Not_want] (s=5) -> (s'=14);
          [Req_propose_to_substitute_Not_want] (s=5) -> (s'=15);
          [Req_propose_to_substitute_Not_want] (s=6) -> (s'=16);
          [Prov_propose_to_addOr_want] (s=7) -> (s'=17);
          [Prov_propose_to_addOr_want] (s=8) -> (s'=18);
          [Req_propose_to_addOr_Not_want] (s=9) -> (s'=19);
          [Req_propose_to_addOr_Not_want] (s=10) -> (s'=20);
          [Req_accept_to_withdrawOr_want] (s=11) -> (s'=21);
          [Req_reject_to_withdrawOr_want] (s=11) -> (s'=22);
          [Req_accept_to_withdrawOr_Not_want] (s=11) -> (s'=23);
          [Req_reject_to_withdrawOr_Not_want] (s=11) -> (s'=24);
          [Req_accept_to_substitute_want] (s=12) -> (s'=25);
          [Req_reject_to_substitute_want] (s=12) -> (s'=26);
          [Req_accept_to_substitute_Not_want] (s=12) -> (s'=27);
          [Req_reject_to_substitute_Not_want] (s=12) -> (s'=28);
          [Req_accept_to_substitute_want] (s=13) -> (s'=29);
          [Req_reject_to_substitute_want] (s=13) -> (s'=30);
          [Prov_accept_to_withdrawOr_want] (s=14) -> (s'=31);
          [Prov_reject_to_withdrawOr_want] (s=14) -> (s'=32);
          [Prov_accept_to_withdrawOr_Not_want] (s=14) -> (s'=33);
          [Prov_reject_to_withdrawOr_Not_want] (s=14) -> (s'=34);
          [Prov_accept_to_substitute_want] (s=15) -> (s'=35);
          [Prov_reject_to_substitute_want] (s=15) -> (s'=36);
          [Prov_accept_to_substitute_Not_want] (s=15) -> (s'=37);
          [Prov_reject_to_substitute_Not_want] (s=15) -> (s'=38);
          [Prov_accept_to_substitute_want] (s=16) -> (s'=39);
          [Prov_reject_to_substitute_want] (s=16) -> (s'=40);
          [Req_accept_to_addOr_want] (s=17) -> (s'=41);
          [Req_reject_to_addOr_want] (s=17) -> (s'=42);
          [Req_accept_to_addOr_Not_want] (s=17) -> (s'=43);
          [Req_reject_to_addOr_Not_want] (s=17) -> (s'=44);
          [Req_accept_to_addOr_want] (s=18) -> (s'=45);
          [Req_reject_to_addOr_want] (s=18) -> (s'=46);
          [Req_accept_to_addOr_Not_want] (s=18) -> (s'=47);
          [Req_reject_to_addOr_Not_want] (s=18) -> (s'=48);
          [Prov_accept_to_addOr_want] (s=19) -> (s'=49);
          [Prov_reject_to_addOr_want] (s=19) -> (s'=50);
          [Prov_accept_to_addOr_Not_want] (s=19) -> (s'=51);
          [Prov_reject_to_addOr_Not_want] (s=19) -> (s'=52);
          [Prov_accept_to_addOr_want] (s=20) -> (s'=53);
          [Prov_reject_to_addOr_want] (s=20) -> (s'=54);
          [Prov_accept_to_addOr_Not_want] (s=20) -> (s'=55);
          [Prov_reject_to_addOr_Not_want] (s=20) -> (s'=56);
          [Req_propose_to_substitute_want] (s=21) -> (s'=57);
          [Req_propose_to_substitute_want] (s=22) -> (s'=58);
          [Prov_propose_to_substitute_Not_want] (s=23) -> (s'=59);
          [Prov_propose_to_substitute_Not_want] (s=24) -> (s'=60);
          [Req_propose_to_withdrawOr_want] (s=25) -> (s'=61);
          [Req_propose_to_withdrawOr_want] (s=26) -> (s'=62);
          [Prov_propose_to_withdrawOr_Not_want] (s=27) -> (s'=63);
          [Prov_propose_to_withdrawOr_Not_want] (s=28) -> (s'=64);
          [Prov_propose_to_substitute_want] (s=31) -> (s'=65);
          [Prov_propose_to_substitute_want] (s=32) -> (s'=66);
          [Req_propose_to_substitute_Not_want] (s=33) -> (s'=67);
          [Req_propose_to_substitute_Not_want] (s=34) -> (s'=68);
          [Prov_propose_to_withdrawOr_want] (s=35) -> (s'=69);
          [Prov_propose_to_withdrawOr_want] (s=36) -> (s'=70);
          [Req_propose_to_withdrawOr_Not_want] (s=37) -> (s'=71);
          [Req_propose_to_withdrawOr_Not_want] (s=38) -> (s'=72);
          [Req_propose_to_withdrawOr_want] (s=41) -> (s'=73);
          [Prov_propose_to_withdrawOr_Not_want] (s=43) -> (s'=74);
          [Req_propose_to_withdrawOr_want] (s=45) -> (s'=75);
          [Prov_propose_to_withdrawOr_Not_want] (s=47) -> (s'=76);
          [Prov_propose_to_withdrawOr_want] (s=49) -> (s'=77);
          [Req_propose_to_withdrawOr_Not_want] (s=51) -> (s'=78);
          [Prov_propose_to_withdrawOr_want] (s=53) -> (s'=79);
          [Req_propose_to_withdrawOr_Not_want] (s=55) -> (s'=80);
          [Prov_accept_to_substitute_want] (s=57) -> (s'=81);
          [Prov_reject_to_substitute_want] (s=57) -> (s'=82);
          [Prov_accept_to_substitute_want] (s=58) -> (s'=83);
          [Prov_reject_to_substitute_want] (s=58) -> (s'=84);
          [Req_accept_to_substitute_want] (s=59) -> (s'=85);
          [Req_reject_to_substitute_want] (s=59) -> (s'=86);
          [Req_accept_to_substitute_want] (s=60) -> (s'=87);
          [Req_reject_to_substitute_want] (s=60) -> (s'=88);
          [Prov_accept_to_withdrawOr_want] (s=61) -> (s'=89);
          [Prov_accept_to_withdrawOr_Not_want] (s=61) -> (s'=90);
          [Prov_accept_to_withdrawOr_want] (s=62) -> (s'=91);
          [Prov_reject_to_withdrawOr_want] (s=62) -> (s'=92);
          [Req_accept_to_withdrawOr_want] (s=63) -> (s'=93);
          [Req_reject_to_withdrawOr_want] (s=63) -> (s'=94);
          [Req_accept_to_withdrawOr_want] (s=64) -> (s'=95);
          [Req_reject_to_withdrawOr_want] (s=64) -> (s'=96);
          [Req_accept_to_substitute_want] (s=65) -> (s'=97);
          [Req_reject_to_substitute_want] (s=65) -> (s'=98);
          [Req_accept_to_substitute_want] (s=66) -> (s'=99);
          [Req_reject_to_substitute_want] (s=66) -> (s'=100);
          [Prov_accept_to_substitute_want] (s=67) -> (s'=101);
          [Prov_reject_to_substitute_want] (s=67) -> (s'=102);
          [Prov_accept_to_substitute_want] (s=68) -> (s'=103);
          [Prov_reject_to_substitute_want] (s=68) -> (s'=104);
          [Req_accept_to_withdrawOr_want] (s=69) -> (s'=105);
          [Req_reject_to_withdrawOr_want] (s=69) -> (s'=106);
          [Req_accept_to_withdrawOr_want] (s=70) -> (s'=107);
          [Req_reject_to_withdrawOr_want] (s=70) -> (s'=108);
          [Prov_accept_to_withdrawOr_want] (s=71) -> (s'=109);
          [Prov_reject_to_withdrawOr_want] (s=71) -> (s'=110);
          [Prov_accept_to_withdrawOr_want] (s=72) -> (s'=111);
          [Prov_reject_to_withdrawOr_want] (s=72) -> (s'=112);
          [Prov_accept_to_withdrawOr_want] (s=73) -> (s'=113);
          [Prov_reject_to_withdrawOr_want] (s=73) -> (s'=114);
          [Req_accept_to_withdrawOr_want] (s=74) -> (s'=115);
          [Req_reject_to_withdrawOr_want] (s=74) -> (s'=116);
          [Prov_accept_to_withdrawOr_want] (s=75) -> (s'=117);
          [Prov_reject_to_withdrawOr_want] (s=75) -> (s'=118);
          [Req_accept_to_withdrawOr_want] (s=76) -> (s'=119);
          [Req_reject_to_withdrawOr_want] (s=76) -> (s'=120);
          [Req_accept_to_withdrawOr_want] (s=77) -> (s'=121);
          [Req_reject_to_withdrawOr_want] (s=77) -> (s'=122);
          [Prov_accept_to_withdrawOr_want] (s=78) -> (s'=123);
          [Prov_reject_to_withdrawOr_want] (s=78) -> (s'=124);
          [Req_accept_to_withdrawOr_want] (s=79) -> (s'=125);
          [Req_reject_to_withdrawOr_want] (s=79) -> (s'=126);
          [Prov_accept_to_withdrawOr_want] (s=80) -> (s'=127);
          [Prov_reject_to_withdrawOr_want] (s=80) -> (s'=128);

endmodule

rewards "prov"
s=0 : 9; s=1 : 0; s=2 : 0; s=3 : 2; s=4 : 0; s=5 : 2; s=6 : 0; s=7 : -
3; s=8 : 0; s=9 : -3; s=10 : 0; s=11 : 0;          s=12 : 0; s=13 : 0;
s=14 : 0; s=15 : 0; s=16 : 0; s=17 : 0; s=18 : 0; s=19 : 0; s=20 : 0;
s=21 : -4; s=22 : 0;          s=23 : -4; s=24 : 0; s=25 : -3; s=26 : 0;
s=27 : -3; s=28 : 0; s=29 : -3; s=30 : 0; s=31 : -4; s=32 : 0; s=33 : -
4; s=34 : 0; s=35 : -3; s=36 : 0; s=37 : -3; s=38 : 0; s=39 : -3; s=40
: 0; s=41 : 2; s=42 : 0; s=43 : 2; s=44 : 0; s=45 : 2; s=46 : 0; s=47 :
2; s=48 : 0; s=49 : 2; s=50 : 0; s=51 : 2; s=52 : 0; s=53 : 2; s=54 :
0; s=55 : 2; s=56 : 0; s=57 : 0; s=58 : 0; s=59 : 0;          s=60 : 0;
s=61 : 0; s=62 : 0; s=63 : 0; s=64 : 0; s=65 : 0; s=66 : 0; s=67 : 0;
s=68 : 0; s=69 : 0; s=70 : 0; s=71 : 0; s=72 : 0; s=73 : 0; s=74 : 0;
s=75 : 0; s=76 : 0; s=77 : 0; s=78 : 0; s=79 : 0; s=80 : 0; s=81 : -3;
s=82 : 0; s=83 : -3; s=84 : 0; s=85 : -3; s=86 : 0; s=87 : -3; s=88 :
0; s=89 : -4; s=90 : -4; s=91 : -4; s=92 : 0; s=93 : -4; s=94 : 0; s=95
: -4; s=96 : 0; s=97 : -3; s=98 : 0; s=99 : -3; s=100 : 0; s=101 : -3;
s=102 : 0; s=103 : -3; s=104 : 0; s=105 : -4; s=106 : 0; s=107 : -4;
s=108 : 0; s=109 : -4; s=110 : 0; s=111 : -4; s=112 : 0; s=113 : -4;
s=114 : 0; s=115 : -4; s=116 : 0; s=117 : -4; s=118 : 0; s=119 : -4;
s=120 : 0; s=121 : -4; s=122 : 0; s=123 : -4; s=124 : 0; s=125 : -4;
s=126 : 0;          s=127 : -4; s=128 : 0;
endrewards
```

PRISM-games SMG from Henshin LTS

```
smg

player requestor
[Req_start_to_addOr], [Req_start_to_substitute],
[Req_propose_to_withdrawOr_Not_want],
[Req_propose_to_substitute_Not_want],
[Req_propose_to_addOr_Not_want],
          [Req_accept_to_withdrawOr_want],
[Req_reject_to_withdrawOr_want], [Req_accept_to_withdrawOr_Not_want],
[Req_reject_to_withdrawOr_Not_want], [Req_accept_to_substitute_want],
[Req_reject_to_substitute_want], [Req_accept_to_substitute_Not_want],
[Req_reject_to_substitute_Not_want], [Req_accept_to_addOr_want],
[Req_reject_to_addOr_want], [Req_accept_to_addOr_Not_want],
[Req_reject_to_addOr_Not_want], [Req_propose_to_substitute_want],
[Req_propose_to_withdrawOr_want]
endplayer
player provider
[Prov_accept_to_addOr_want], [Prov_reject_to_addOr_want],
[Prov_accept_to_addOr_Not_want], [Prov_reject_to_addOr_Not_want],
[Prov_accept_to_substitute_want], [Prov_reject_to_substitute_want],
[Prov_accept_to_substitute_Not_want],
[Prov_reject_to_substitute_Not_want],
[Prov_propose_to_withdrawOr_want], [Prov_propose_to_substitute_want],
[Prov_propose_to_addOr_want], [Prov_accept_to_withdrawOr_want],
[Prov_reject_to_withdrawOr_want], [Prov_accept_to_withdrawOr_Not_want],
[Prov_reject_to_withdrawOr_Not_want],
[Prov_propose_to_substitute_Not_want],
[Prov_propose_to_withdrawOr_Not_want]
endplayer
module M

          s : [0..128] init 0;
          [Req_start_to_addOr] (s=0) -> (s'=1);
          [Req_start_to_substitute] (s=0) -> (s'=2);
          [Prov_accept_to_addOr_want] (s=1) -> (s'=3);
          [Prov_reject_to_addOr_want] (s=1) -> (s'=4);
          [Prov_accept_to_addOr_Not_want] (s=1) -> (s'=5);
          [Prov_reject_to_addOr_Not_want] (s=1) -> (s'=6);
          [Prov_accept_to_substitute_want] (s=2) -> (s'=7);
```

UNIVERSITY OF LEICESTER

PRISM-games SMG from Henshin LTS

```
            [Req_accept_to_withdrawOr_want] (s=76) -> (s'=119);
            [Req_reject_to_withdrawOr_want] (s=76) -> (s'=120);
            [Req_accept_to_withdrawOr_want] (s=77) -> (s'=121);
            [Req_reject_to_withdrawOr_want] (s=77) -> (s'=122);
            [Prov_accept_to_withdrawOr_want] (s=78) -> (s'=123);
            [Prov_reject_to_withdrawOr_want] (s=78) -> (s'=124);
            [Req_accept_to_withdrawOr_want] (s=79) -> (s'=125);
            [Req_reject_to_withdrawOr_want] (s=79) -> (s'=126);
            [Prov_accept_to_withdrawOr_want] (s=80) -> (s'=127);
            [Prov_reject_to_withdrawOr_want] (s=80) -> (s'=128);

endmodule

rewards "prov"
s=0 : 9; s=1 : 0; s=2 : 0; s=3 : 2; s=4 : 0; s=5 : 2; s=6 : 0; s=7 : -
3; s=8 : 0; s=9 : -3; s=10 : 0; s=11 : 0;          s=12 : 0; s=13 : 0;
s=14 : 0; s=15 : 0; s=16 : 0; s=17 : 0; s=18 : 0; s=19 : 0; s=20 : 0;
s=21 : -4; s=22 : 0;           s=23 : -4; s=24 : 0; s=25 : -3; s=26 : 0;
s=27 : -3; s=28 : 0; s=29 : -3; s=30 : 0; s=31 : -4; s=32 : 0; s=33 : -
4; s=34 : 0; s=35 : -3; s=36 : 0; s=37 : -3; s=38 : 0; s=39 : -3; s=40
: 0; s=41 : 2; s=42 : 0; s=43 : 2; s=44 : 0; s=45 : 2; s=46 : 0; s=47 :
2; s=48 : 0; s=49 : 2; s=50 : 0; s=51 : 2; s=52 : 0; s=53 : 2; s=54 :
0; s=55 : 2; s=56 : 0; s=57 : 0; s=58 : 0; s=59 : 0;          s=60 : 0;
s=61 : 0; s=62 : 0; s=63 : 0; s=64 : 0; s=65 : 0; s=66 : 0; s=67 : 0;
s=68 : 0; s=69 : 0; s=70 : 0; s=71 : 0; s=72 : 0; s=73 : 0; s=74 : 0;
s=75 : 0; s=76 : 0; s=77 : 0; s=78 : 0; s=79 : 0; s=80 : 0; s=81 : -3;
s=82 : 0; s=83 : -3; s=84 : 0; s=85 : -3; s=86 : 0; s=87 : -3; s=88 :
0; s=89 : -4; s=90 : -4; s=91 : -4; s=92 : 0; s=93 : -4; s=94 : 0; s=95
: -4; s=96 : 0; s=97 : -3; s=98 : 0; s=99 : -3; s=100 : 0; s=101 : -3;
s=102 : 0; s=103 : -3; s=104 : 0; s=105 : -4; s=106 : 0; s=107 : -4;
s=108 : 0; s=109 : -4; s=110 : 0; s=111 : -4; s=112 : 0; s=113 : -4;
s=114 : 0; s=115 : -4; s=116 : 0; s=117 : -4; s=118 : 0; s=119 : -4;
s=120 : 0; s=121 : -4; s=122 : 0; s=123 : -4; s=124 : 0; s=125 : -4;
s=126 : 0;          s=127 : -4; s=128 : 0;
endrewards
```

UNIVERSITY OF LEICESTER

Generating Optimal Strategies

We define rPATL reward-based property for the provider:

<<provider>> R{”prov”}max=? [F ”deadlock”]

PRISM-games generates a strategy for the provider to maximise the expected value of reward ”prov” in final states.

Seen as zero-sum game, the optimal strategy generated by PRISM-games is a Nash equilibrium.

Conclusion and Future Work

Graph Transformation Games

- Using graph transformations to model negotiation games as a state-based transformations
- Using game theory to analyse/optimise them

Future work

- other strategies: cooperation between provider and requestor, competition
- compiler approach: model transformation generating PRISM-game GSM directly
- applications: generating negotiators, feeding model with real data