

Anhang

A - Quellcode des Simulationsprogramms SimSoP.pl

```
%*****  
% simsop.pl - simulation of a particular social practice  
% files: para.int - parameter file  
%       data.int - initially generated intentions  
%               and beliefs  
%       beliefs.int - protocol of belief dynamics  
%       results.int - result of simulation run(s)  
%*****  
  
%-----  
% main hook used when running this program as a stand-alone  
application  
%-----  
simsop_main_hook :-  
    start,  
    abort.  
  
%-----  
% abort hook used when running this program as a stand-alone  
application  
%-----  
simsop_abort_hook :-  
    halt(0).  
  
%*****  
%  
% predicates to launch a sequence of simulation runs  
%  
%*****  
  
%-----  
% initialize and perform a sequence of simulation runs  
%-----  
start :- nofileerrors, dynamic(equal_list/1),  
        randomize,
```

```

load_files('para.int',[all_dynamic(true)]),
check_parameter,
ini_graph_para,
begin,
retractall(fact(X)),
% close output files for graphic trace / statistics
result_file(RF),
belief_file(BF),
close(RF),
close(BF).

%-----
% perform a sequence of simulation runs
%-----
begin:-
    runs(RR), periods(TT),
    ( between(1,RR,R),
      get_data,
      mainloop(R,TT),
      fail; true ),!.

%-----
% generate a new attitude set
%-----
get_data :-
    use_old_data(X),
    X = no, !,
    gen_intentions,
    gen_belief_level(1),
    write_data, !.

%-----
% use an old attitude set
%-----
get_data :-
    use_old_data(X),
    X = yes, !,
    consult_file('beliefs.int'),
    consult_file('data.int').

```

```

%*****
%
%           generation of the agent's initial attitudes
%           - intentions and beliefs -
%
%*****

%-----
% generate the initial intentions
%-----
gen_intentions :-
    actors(AS),
    bagof(Y, fact(act(Y)), L),          % get list of action types
    length(L, N),
    ini_individual(1, AS, N, L).

%-----
% generate recursively the initial attitudes of all agents
% I - agent
% A - maximal number of agents
% N - number of defined action types
% L - list of defined action types
%-----
ini_individual(I, A, N, L) :-
    (I =< A,
     percent_int(PI),
     M is N - (N / 100 * PI),          % calculate the resting facts
     ( gen_int_facts(I, N, M, L)       % generate I's initial
                                           intentions
     ;
      I1 is I + 1,
      ini_individual(I1, A, N, L))) % initialize the next agent
    ;
    true.

%-----
% generate recursively I's initial intentions
% I - agent
% N - number of selectable action types
% M - recursion terminator, remaining facts in L
% L - list of selectable action types
%-----
gen_int_facts(I, N, M, L) :-
    N > M,
    random(N, R),

```

```

member(ACT,L,R),!,
assertz(fact(0,0,int([I,t,ACT]))),
assertz(bel([1,I,t,ACT])),
removeall(ACT,L,L1),
N1 is N - 1,
gen_int_facts(I,N1,M,L1).

```

```

%-----
% generate recursively higher order beliefs
% L - belief level to generate
%-----

```

```

gen_belief_level(L):-
  (maxlev(ML),
  L =< ML,
  L1 is L + 1,
  gen_individual(1,L),
  gen_belief_level(L1))
;
true.

```

```

%-----
% generate recursively beliefs of a certain order
% I - agent
% L - belief level to generate
%-----

```

```

gen_individual(I,L):-
  (actors(MI),
  I =< MI,
  I1 is I + 1,
  gen_beliefs(1,I,L),
  gen_individual(I1,L))
;
true.

```

```

%-----
% generate recursively agent I's higher order beliefs
% pertaining to other agents
% J - agent
% I - agent
% L - belief level to generate
%-----

```

```

gen_beliefs(J,I,L):-
  (actors(MI),
  J =< MI,
  J1 is J + 1,

```

```

get_bel_predicates(J,I,L),
gen_beliefs(J1,I,L)
;
true.

%-----
% generate recursively agent I's own higher order beliefs
% agent I believes all own predicates of lower level
% J - agent
% I - agent
% L - belief level to generate
%-----
get_bel_predicates(J,I,L):-
  J == I,
  L1 is L + 1,
  forall(bel([L,I,W,R]),
    assertz(bel([L1,I,t,bel([L,I,W,R])))).

%-----
% generate recursively agent I's higher order beliefs per-
% taining to agent J
% agent I believes that agent J believes percent_bel
% percent of own believes of lower level, too
% J - agent
% I - agent
% L - belief level to generate
%-----
get_bel_predicates(J,I,L):-
  J \= I,
  L1 is L + 1,
  bagof([W|R],bel([L,I,W,R]),R1),
  length(R1,N),
  percent_bel(PB),
  M is N - (N / 100 * PB), % calculate the resting facts
  (gen_other_beliefs(N,J,I,L1,t,M,R1)
;
  true).

```

```

%-----
% generate I's beliefs of level L pertaining to agent J's
beliefs
% N = number of entries of I's belief base of lower level
% J = other actor
% I = actor to generate
% L = level / belief order
% W = truth value
% M = maximum for resting beliefs
%-----
gen_other_beliefs(N,J,I,L,W,M,X1):-
    M < N,
    L1 is L - 1,
    random(N,R),
    member(X,X1,R),!,
    L1 is L - 1,
    append([L1,J],X,X2),
    assertz(bel([L,I,W,bel(X2)])),
    removeall(X,X1,New),
    N1 is N - 1,
    gen_other_beliefs(N1,J,I,L,W,M,New).

%-----
% start a simulation run of the social practice of the group
% life of a leisure group
% R - simulation run
% TT - points in time to process
%-----
mainloop(R,TT) :-
    ini_facts_for_run(R),!,
    ( between(1,TT,T),
      continue_practice(yes),
      write_dynamic_belief(R,T),
      write_results(R,T),
      kernel(R,T), told, fail; true ),
    do_retract,
    !.

%-----
% initialize the fact base for the simulation run
% R - simulation run
%-----
ini_facts_for_run(R):-
    asserta(continue_practice(yes)),
    findall(X,fact(0,0,X),L),

```

```

length(L,E),
( between(1,E,Z),
  member(FACT,L,Z),
  retract(fact(0,0,FACT)), asserta(fact(R,1,FACT)),
fail; true ),
  asserta(hist(actor_left([R,0,0]))).

retract_facts :- ( fact(X,Y,Z), retract(fact(X,Y,Z)), fail
;
  true ).

%-----
% perform a pattern of the social practice
%   - rule set negotiate -
% R - run
% T - point in time
%-----
kernel(R,T) :- rule_set(negotiate), actors(AS),
findall(I,between(1,AS,I),L),
  gen_actor_list(R,L), continue_practice,
  ( between(1,AS,N), check_intention(R,T,N), fail
;
  true ),
  ( between(1,AS,N), suggest_action(R,T), fail; true ),
  choose_action(R,T,ACT),
  common_action(R,T,ACT),
  ( between(1,AS,N), evaluate_action(R,T,N,ACT), fail
;
  true ),
  retract(actor_list(L1)),
  retract_pref_act,
  !.

%-----
% perform a pattern of the social practice
%   - rule set get_random -
% R - run
% T - point in time
%-----
kernel(R,T) :- rule_set(get_random), actors(AS),
findall(I,between(1,AS,I),L),
  gen_actor_list(R,L), continue_practice,
  actor_list(L1), length(L1,AC),
  ( between(1,AC,N), get_rdm_intention(R,T), fail
;

```

```

    true),
    choose_action(R,T,ACT),
    common_action(R,T,ACT),
    ( between(1,AS,N1), evaluate_action(R,T,N1,ACT), fail
;
    true),
    retract(actor_list(L2)),
    retract_pref_act,
    !.

%-----
% remove non-active agents from actor list
% R - run
% L - actor list
%-----
gen_actor_list(R,L) :-
    findall(A,hist(actor_left([R,_,A])),LX),
    length(LX,E),
    asserta(actor_list(L)),
    (between(1,E,N),
     member(A1,LX,N),
     delete_from_actor_list(A1),
     fail
;
     true).

%-----
% delete an agent from the actor-list
% A - agent
%-----
delete_from_actor_list(A) :-
    actor_list(L),
    removeall(A,L,L1), /* delete(L,A,L1) */
    retract(actor_list(L)),
    asserta(actor_list(L1)),!.

%-----
% check whether the social practice shall be continued
% A - agent
%-----
continue_practice:-
    actor_list(L),
    length(L, L1),
    min_agents(M),
    (M =< L1

```



```

;
    retract(continue_practice(yes)), !,
    asserta(continue_practice(no)), fail).

%-----
% generate the suggestions for the group action
% R - run
% T - point in time
%-----
suggest_action(R,T) :- actor_list(L),
    length(L,E), E > 0,
    random(E,R1), Y is R1, member(A,L,Y),
    activate(R,T,A), removeall(A,L,L1),
    retract(actor_list(L)), asserta(actor_list(L1)),!.

%-----
% generate agent A's suggestion for the group action
% R - run
% T - point in time
% A - agent
%-----
activate(R,T,A) :-
( check_mail(R,T,A,L),           % check the personal mailbox
  gen_common_action(R,T,A,L) ; true).

%-----
% perform random selection of the agent's proposals
%           - rule set get_random -
% R - run
% T - point in time
%-----
get_rdm_intention(R,T) :-
    actor_list(L1), length(L1,E), E > 0,
    random(E,R1), member(A,L1,R1),
    findall(int([A,B,C]), fact(R,T,int([A,B,C])),L),
    length(L,N),
    random(N,R2),
    member(int([A1,B,ACT]),L,R2),!,
    out_action(R,T,A,ACT),
    removeall(A,L1,L2),
    retract(actor_list(L1)), asserta(actor_list(L2)),!.

%-----
% generate proposals for the group action

```

```

%           - rule_set negotiate -
% R - run
% T - point in time
% A - agent
%-----
check_intention(R,T,A) :-
    findall(int([A,B,C]),fact(R,T,int([A,B,C])),L),
    ( check_int_trg(R,T,A,L),!           % check triggered intentions
      ;
      do_max_calc(R,T,A,L) ).

%-----
% search positive trigger for the current point in time
% R - run
% T - point in time
% A - agent
% L - list that contains A's potentially intended actions
%-----
check_int_trg(R,T,A,L) :-
    fact(meetings_per_year(MP)),
    MP > 0,
    M is T mod MP,
    fact(ptrg(M,ACT)),
    member(int([A,B,ACT]),L),!,
    findall(bel([1,A2,B,ACT]),
            bel([2,A,B2,bel([1,A2,B,ACT])]),L1),
    length(L1,E1),
    E1 > 1,
    retractall(mail(A,_,_)),
    forall(member(bel([1,A3,_,ACT]),L1),
           assertz(mail(A,A3,trg(ACT)))).

%-----
% search the intention that agent A believes most of the
% others will share
% R - run
% T - point in time
% A - agent
% L - list that contains A's potentially intended actions
%-----
do_max_calc(R,T,A,L) :-
    assert(tmpcnt(0)),
    length(L,E),
    ( between(1,E,Z),
      member(int([A1,B,C]),L,Z),

```

```

    findall(
      bel([1,A2,B,C]),
      bel([2,A1,B2,bel([1,A2,B,C])]),L2),
    length(L2,E2),
    tmpcnt(E1),
    E2 > E1,
    not check_neg_trg(T,C),
    retract(tmpcnt(E1)),
    assert(tmpcnt(E2)),
    retractall(mail(A,_,_)),
    forall(member(bel([1,A3,_,ACT]),L2),
      assertz(mail(A,A3,trg(ACT)))),
    fail; true
  ),
  retract(tmpcnt(X)),!.

%-----
% check negative trigger - the performance of the action is
% not possible at the current point in time
% R - run
% ACT - action
%-----
check_neg_trg(T,ACT):-
  fact(meetings_per_year(MP)),
  MP > 0,
  M is T mod MP,
  (fact(ptrg(TT,ACT)),
   TT =\= M
  );
  fact(ntrg(M,ACT)),!.

%-----
% check the mail in the personal mailboxes of the agent
% R - run
% T - point in time
% A - agent
% L - list that contains A's potentially intended actions
%-----
check_mail(R,T,A,L) :-
  % assert corresponding beliefs?
  (update_belief(yes),
   forall(mail(A1,A,trg(ACT)), expand_belief(A1,A,ACT))
  );
  true),
  findall(trg(ACT),mail(A1,A,trg(ACT)),L1),
  length(L1,E),

```

```

    ( between(1,E,Z),
      check_mail_int(R,T,A,L1,Z),
      fail
    ;
    true),
    findall(trg(ACT),mail(A1,A,trg(ACT)),L),
    retractall(mail(_,A,_)). /* clean mail of actor */

%-----
% check the mail in the personal mailboxes of the agent
% R - run
% T - point in time
% A - agent
% L - list that contains A's potentially intended actions
%-----
check_mail_int(R,T,A,L,Z) :-
    member(trg(ACT),L,Z),
    ( fact(R,T,int([A,t,ACT])),!
    ;
    retract(mail(A1,A,trg(ACT))) ).

%-----
% expand the receivers belief from received proposals
% A1 - sender
% A - receiver
% ACT - proposed action
%-----
expand_belief(A1,A,ACT) :-
    bel([2,A,t,bel([1,A1,t,ACT])])
    ;
    assertz(bel([2,A,t,bel([1,A1,t,ACT])])).

%-----
% generate the A's suggestion for the common action
% R - run
% T - point in time
% A - agent
% L - list that contains proposals in A's personal mailbox
%-----
gen_common_action(R,T,A,L) :-
    get_preferred_action(L,ACT),
    out_action(R,T,A,ACT).

%-----
% select a preferred action from a personal mailbox
% L - list that contains the proposals in a personal mailbox

```

```

% ACT - selected action
%-----
get_preferred_action(L,ACT):-
    % find maximum of proposals
    sort(L,L1,[]),
    asserta(max_elem_list(1)),
    asserta(max_list([])),
    (max_elem(trg(ACT),L1),
     (ground(ACT) ;
      default_action(ACT))

;
    % bei keinem eindeutigen Maximum Zufallsauswahl
    length(L,E),
    random(E,R1),
    member(trg(ACT),L,R1)),
    retractall(max_elem_list(X)),
    retractall(equal_list(Y)),
    retract(max_list(Y1)).

%-----
% calculate the trigger for the group action
% R - run
% T - point in time
% ACT - triggered group action
%-----
choose_action(R,T,ACT):-
    findall(ACT1, preferred_action(R,T,X,ACT1),L),
    sort(L,L1,[]),
    asserta(max_elem_list(1)),
    asserta(max_list([])),
    (max_elem(ACT,L1), ACT \== no_action
     % bei keinem eindeutigen Maximum Zufallsauswahl unter
     % den meisten Vorschlägen
; max_list(L2),
  removeall(no_action, L2, L3),
  length(L3,E),
  E > 0,
  random(E,R1),
  member(ACT,L3,R1)),
  retractall(max_elem_list(Y1)),
  retractall(equal_list(Y2)),
  retract(max_list(L2)).

%-----
% put agent A's proposal for the group action

```

```

% to the public board and make an entry to the result file
% R - run
% T - point in time
% A - agent
% ACT - action
%-----
out_action(R,T,A,ACT):-
    assertz(preferred_action(R,T,A,ACT)),
    write(preferred_action(R,T,A,ACT)),write('.'), nl.

%-----
% simulate the group action
% R - run
% T - point in time
% ACT - action
%-----
common_action(R,T,ACT):-
    write(common_action(R,T,ACT)),write('.'), nl.
%*****
%
% predicates for the success/threshold calculation
%
%*****

%-----
% calculate the success of the action
% R - run
% T - point in time
% A - agent
% ACT - performed group action
%-----
evaluate_action(R,T,A,ACT):-
    success_level(R,T,A,ACT),
    gen_next_ints(R,T,A), !.

%-----
% calculate success if the action was the agents' preferred
% action
% R - run
% T - point in time
% A - agent
% ACT - performed group action
%-----
success_level(R,T,A,ACT):-

```

```

    preferred_action(R,T,A,ACT1),
    ACT == ACT1,
    rate_s(RS), /* get success rate for suggested actions */
    random(100,R1), !,
    (R1 > RS, retract_act(A,ACT)
    ; true).

%-----
% calculate success if the action wasn't the agents'
% preferred action
% R - run
% T - point in time
% A - agent
% ACT - performed group action
%-----
success_level(R,T,A,ACT):-
    preferred_action(R,T,A,ACT2),
    ACT \== ACT2,
    /* does the agent intend the action ? */
    ( fact(R,T,int([A,Y,ACT])),
      rate_i(RI), /* get success rate */
      random(100,R1), !,
      (R1 > RI, retract_act(A,ACT) ; true)
    ;
    % get success rate for previously not intended actions
    rate_ni(RN),
    random(100,R1), !,
    (R1 =< RN, assertz(fact(X,T1,int([A,t,ACT]))),
      assertz(bel([2,A,t,bel([1,A,t,ACT]])))
    ; no_success(R,T,A,ACT),
      check_threshold(R,T,A)).

%-----
% initialize agent A's intention for the next point in time
% R - run
% T - point in time
% A - agent
%-----
gen_next_ints(R,T,A):-
    findall(Z0, fact(X0,T,int([A,Y0,Z0])),L),
    length(L,L1),
    (L1 > 0,
     T1 is T + 1,
     forall(fact(X,T,int([A,Y,Z])),
            assertz(fact(X,T1,int([A,Y,Z])))),

```

```

forall (fact (X,T,int ([A,Y,Z])),
        retract (fact (X,T,int ([A,Y,Z])))
;
assertz (hist (actor_left ([R,T,A]))) .

%-----
% retract attitudes for an unsuccessful action
% A    - agent
% ACT  - action
%-----
retract_act(A,ACT):-
    (ACT \== no,
     retract (fact (X,T,int ([A,_,ACT]))),
     retract (bel ([2,A,t,bel ([1,A,t,ACT])]))
    ; true.

%-----
% add history entry for an unsuccessful action
% R - run
% T - point in time
% A - agent
%-----
no_success (R,T,A,ACT):-
    assertz (hist (no_success ([R,T,A,ACT])),
            write (no_success (R,T,A,ACT)), write ('.'), nl.

%-----
% check threshold for an agent to leave the group
% R    - run
% T    - point in time
% A    - agent
%-----
check_threshold (R,T,A):-
    failure_rate (FR),
    findall (X,hist (no_success ([R,X,A,_])),L),
    length (L,L1),
% does the agent leave the group ?
    (L1 > T * FR,
     retract_att (R,T,A)
    ;
     check_subsequent_failures (R,T,A)).

%-----
% calculate subsequent failures
% R    - run

```



```

% T - point in time
% A - agent
%-----
check_subsequent_failures(R,T,A):-
    subsequent_failures(SF),
    (get_subsequent_failures(R,T,A,SF),
     retract_att(R,T,A)
    ;
     true).

get_subsequent_failures(R,T,A,SF):-
    SF > 0,
    hist(no_success([R,T,A,_])), !,
    T1 is T - 1,
    SF1 is SF - 1,
    get_subsequent_failures(R,T1,A,SF1).

get_subsequent_failures(R,T,A,SF):-
    SF == 0.

%-----
% retract attitudes of agent that leaves the group
% R - run
% T - point in time
% A - agent
%-----
retract_att(R,T,A):-
    % intentions of the agent
    retractall(fact(R,T1,int([A,Y,Z]))),
    % beliefs of agent A about other agents
    retractall(bel([2,A,t,bel([1,A1,t,ACT1]))),
    % beliefs of other agents about agent A
    retractall(bel([1,A,t,ACT2])),
    retractall(bel([2,A2,t,bel([1,A,t,ACT2]))),
    assertz(hist(actor_left([R,T,A]))).

%-----
% retract facts after a simulation run
%-----
do_retract:-
    retractall(continue_practice(P)),
    retractall(actor_list(AL)),
    retractall(mail(A1,A2,A3)),
    retractall(bel(B)),

```

```

    retractall (fact (D1,D2,D3)),
    retractall (preferred_action (E1,E2,E3,E4)),
    retractall (hist (X)).

retractmail:-
    forall (mail (X,Y,Z), retract (mail (X,Y,Z))).

retractint:-
    forall (int (X), retract (int (X))).

retractbel:-
    forall (bel (X), retract (bel (X))).

retractfact:-
    forall (fact (X), retract (fact (X))),
    forall (fact (X,Y,Z), retract (fact (X,Y,Z))).

retract_pref_act:-
    forall (preferred_action (X,Y,Z,A),
    retract (preferred_action (X,Y,Z,A))).

%-----
% write new intention file data.int
%-----
write_data:-
    del_file ('data.int'),
    tell ('data.int'),
    forall (fact (0,0,int (X)), out_int (0,0,X)),
    told,
    write_belief.

out_int (R,T,X):-
    write (fact (R,T,int (X))), write ('. '), nl.

%-----
% write new belief file belief.int
%-----
write_belief:-
    del_file ('beliefs.int'),
    tell ('beliefs.int'),
    forall (bel (X), out_bel (X)),
    told.

out_bel (X):-

```

```

write (bel(X)), write('.'), nl.

%-----
% write protocol of the simulation results
%-----
write_results(R,T):-
    result_file(RF),
    open(RF,append),
    bagof(X, fact(R,T,X), L),
    length(L,E),
    ( between(1,E,Z),
      member(FACT,L,Z),
      tell(RF),
      write(fact(R,T,FACT)), write('.'), nl,
      fail; true ),
    tell(RF), nl, !.

%-----
% write protocol of the belief base
%-----
write_dynamic_belief(R,T) :-
    rule_set(get_random).

write_dynamic_belief(R,T) :-
    rule_set(negotiate),
    belief_file(BF),
    open(BF,append),
    tell(BF),
    forall(bel(X), out_dyn_bel(R,T,X)),
    told.

out_dyn_bel(R,T,X):-
    write(belfact(R,T,bel(X))), write('.'), nl.

%-----
% write parameter file for the analysis tools
% - graphical trace and statistics -
%-----
ini_graph_para :-
    write_disp_file,
    para_file(PF),
    (
    del_file(PF) ; true),
    tell(PF),
    continue(C),

```

```

write(continue(C)),write('.'),nl,
runs(R),
write(runs(R)),write('.'),nl,
periods(P),
write(cycles(P)),write('.'),nl,
actors(A),
write(actors(A)),write('.'),nl,
percent_int(PI),
write(percent_int(PI)),write('.'),nl,
percent_bel(PB),
write(percent_bel(PB)),write('.'),nl,
rate_s(RS),
write(rate_s(RS)),write('.'),nl,
rate_i(RI),
write(rate_i(RI)),write('.'),nl,
rate_ni(RNI),
write(rate_ni(RNI)),write('.'),nl,
failure_rate(FR),
write(failure_rate(FR)),write('.'),nl,
subsequent_failures(SF),
write(subsequent_failures(SF)),write('.'),nl,
rule_set(RUL),
write(rule_set(RUL)),write('.'),nl,
use_old_data(UOD),
write(use_old_data(UOD)),write('.'),nl,
update_belief(UB),
write(update_belief(UB)),write('.'),nl,
forall(fact(X),out_disp_fact(X)),
told.

%-----
% write input file for the analysis tools
%   - graphical trace and statistics -
%-----
write_disp_file :-
  disp_file(DF),
  (
    del_file(DF) ; true),
  tell(DF),
  result_file(RF),
  write(result_file),write(' '),
  writeq(RF),write(' '),write('.'),nl,
  belief_file(BF),
  write(belief_file),write(' '),
  writeq(BF),write(' '),write('.'),nl,

```

```

para_file(PF),
write(para_file),write('('),
writeq(PF),write(')'),write('.'),nl,
stat_file(SF),
write(stat_file),write('('),
writeq(SF),write(')'),write('.'),nl,
told,
(
% delete old result file, if it exists
del_file(RF) ; true),
(
% delete protocol of belief base, if it exists.
del_file(BF) ; true).

out_disp_fact(X):-
write(fact(X)),write('.'),nl.

%-----
% check validity of values of input parameters from para.int
%-----
check_parameter:-
rate_s(P1),
(P1 =< 100
;
print('invalid parameter value rate_s '),!,print(P1),
nl, !, fail
),
(P1 >= 0
;
print('invalid parameter value rate_s '),print(P1),
nl, !, fail),
rate_i(P2),
(P2 =< 100
;
print('invalid parameter value rate_i '),print(P2),
nl, !, fail),
(P2 >= 0
;
print('invalid parameter value rate_i '),print(P2),
nl, !, fail),
rate_ni(P3),
(P3 =< 100
;
print('invalid parameter value rate_ni '),print(P3),
nl, !, fail),

```

```

(P3 >= 0
;
print('invalid parameter value rate_ni '),print(P3),
nl,!, fail).

%*****
%
% auxiliary predicates
%
%*****

%=====
%
% predicates for file access
%
%=====

%-----
% delete file if exists
%-----
del_file(X):-
    (exists_file(X),
    del(X))
;
    true.

%-----
% check if file exists
%-----
exists_file(X):-
    % check whether file already exists
    absolute_file_name(X,[access(read)],Y).

%-----
% load all predicates in a file as dynamic predicates
%-----
consult_file(X):-
    load_files(X,[all_dynamic(true)]).

%=====
%
% predicates for random processing
%
%=====

```

```

%-----
% initialize the random generator
%-----
randomize:-
    time(1, (X1)),
    asserta(help_rdm(X1)),
    help_rdm((X2,X)),      % take only number of milliseconds
    retract(help_rdm(X1)), % from time structure
    asserta(rdm(X)).

%-----
% calculate random number
%-----
random(Max,R):-
    Max > 0,
    retract(rdm(S)),
    R is (S mod Max) + 1,
    NewSeed is (125 * S + 1) mod 4096,
    asserta(rdm(NewSeed)),!.

%-----
% get the element, which occurs in a list most often
%-----
max_elem(X, []).

max_elem(X,L):-
    max_elem_list(N),
    member(Y,L,1),
    occurrence(Y, L, N1),
    N1 > N,
    length(L,LEN),
    retract(max_elem_list(N)),
    asserta(max_elem_list(N1)),
    retract(max_list(ML)), % Update list of equal maxima
    append([Y],[],ML1), !,
    asserta(max_list(ML1)),
    (N1 > LEN / 2, copy_term(Y,X), !
    ;
    removeall(Y, L, L2),
    (max_elem(X, L2), !
    ;
    not equal_list(N1, copy_term(Y,X))).

max_elem(X1,L):-
    max_elem_list(N),

```

```

member(X,L,1),
occurrence(X, L, N1),
N1 < N,
length(L,LEN),
(N1 > LEN / 2, !, fail ;
  removeall(X, L, L2),
  max_elem(X1, L2)).

max_elem(X1,L):-
  max_elem_list(N),
  member(X,L,1),
  occurrence(X, L, N1),
  N1 == N,
  asserta(equal_list(N1)),
  retract(max_list(ML)),
  append([X],ML,ML1), !,
  asserta(max_list(ML1)),
  length(L,LEN),
  (N1 > LEN / 2, !, fail
  ;
  removeall(X, L, L2),
  max_elem(X1, L2)),
  max_elem_list(N2),
  (N2 > N, retract(equal_list(N1)),!
  ;
  fail).

%*****
%
% various auxiliary predicates
%
%*****

between(X,Y,Z):-
  integer(Z),
  X =< Z,
  Z =< Y,!

between(_,_,Z):-
  integer(Z),!,fail.

between(X,Y1,Z):-
  var(Z),
  X1 is X + 1,
  X1 =< Y1,

```



```
    between(X1,Y1,Z1),
    Z is Z1 - 1.

between(X,Y,Y).

memnum(N,[X|_],X):-
    N = 1,!.

memnum(N,[_|R],X):-
    !,N2 is N-1,
    memnum(N2,R,X).

occurrence(X,L,N):-
    collect(X,L,L1),
    length(L1,N).

collect(X,L,[X|R1]):-
    append([X|R1],[Y|L1],L),
    X \== Y,!.

collect(X,L,[X|R1]):-
    append([X|R1],[],L),!.

quicksort([X|Xs],Ys):-
    partition(Xs,X,Littles,Bigs),
    quicksort(Littles,Ls),
    quicksort(Bigs,Bs),
    append(Ls,[X|Bs],Ys).

quicksort([],[]).

partition([X|Xs],Y,[X|Ls],Bs):-
    X =< Y,
    partition(Xs,Y,Ls,Bs).

partition([X|Xs],Y,Ls,[X|Bs]):-
    X > Y,
    partition(Xs,Y,Ls,Bs).

partition([],Y,[],[]).
```

B - Beispiel einer Parameterdatei

```

% =====
% File 'para' (parameters of social practices simulation)
% =====
rule_set(negotiate).      % negotiate / get_random
runs(50).                 % number of simulation runs
periods(20).              % number of points in time
actors(10).               % number of agents

%
=====
% par. for the generation / maintenace of the attitude set
%
=====
percent_int(60).          % percentage of shared intentions
percent_bel(50).          % percentage of shared beliefs
maxlev(1).                % belief order
use_old_data(no).         % no = generate new attitude set
                          % yes = use old attitude set
default_action(no_action).
update_belief(yes).       % update belief about other
                          % agents' wants from proposals

% =====
% parameters for the success / threshold calculation
% =====
failure_rate(0.4).        % acceptable failure rate
subsequent_failures(4).   % acceptable number of
                          % subsequent failures
min_agents(2).            % minimal number of agents
rate_s(90).                % rate if suggested
rate_i(70).                % rate if intended in principle
rate_ni(20).              % rate if previously not intended

% =====
% action types
% =====
fact(act(go_to_the_cinema)).
fact(act(skiing)).
fact(act(bicycle_tour)).
fact(act(discussion_about_medicin)).
fact(act(discussion_about_art)).
fact(act(go_to_the_theatre)).
fact(act(go_to_the_pub)).

```

```
fact(act(weiberfasching)).
fact(act(make_masks)).
fact(act(make_campfire)).
fact(act(play_poker)).
fact(act(go_to_the_disco)).
fact(act(go_to_the_fitness_studio)).
fact(act(visit_the_museum)).
fact(act(hike_in_the_mountains)).
fact(act(brunch)).
fact(act(reading_evening)).
fact(act(go_to_a_concert)).
fact(act(walpurgisnacht_party)).
fact(act(chat)).
fact(act(drum_session)).
fact(act(afro_dancing)).
fact(act(belly_dancing)).

% =====
% time slice for trigger conditions
% =====
fact(meetings_per_year(12)).

% =====
% positive intention triggers
% =====
fact(ptrg(1,make_masks)).
fact(ptrg(2,weiberfasching)).
fact(ptrg(4,walpurgisnacht_party)).

% =====
% negative intention triggers
% =====
fact(ntrg(5,skiing)).
fact(ntrg(6,skiing)).
fact(ntrg(7,skiing)).
fact(ntrg(8,skiing)).
fact(ntrg(9,skiing)).
fact(ntrg(10,skiing)).
fact(ntrg(11,skiing)).
fact(ntrg(11,bicycle_tour)).
fact(ntrg(12,bicycle_tour)).
fact(ntrg(1,bicycle_tour)).
fact(ntrg(2,bicycle_tour)).
fact(ntrg(3,bicycle_tour)).
fact(ntrg(4,bicycle_tour)).
```

```
fact(ntrg(12,make_campfire)).
fact(ntrg(1,make_campfire)).
fact(ntrg(2,make_campfire)).
fact(ntrg(3,make_campfire)).
fact(ntrg(4,make_campfire)).
fact(ntrg(11,hike_in_the_mountains)).
fact(ntrg(12,hike_in_the_mountains)).
fact(ntrg(1,hike_in_the_mountains)).
fact(ntrg(2,hike_in_the_mountains)).
fact(ntrg(3,hike_in_the_mountains)).
fact(ntrg(4,hike_in_the_mountains)).

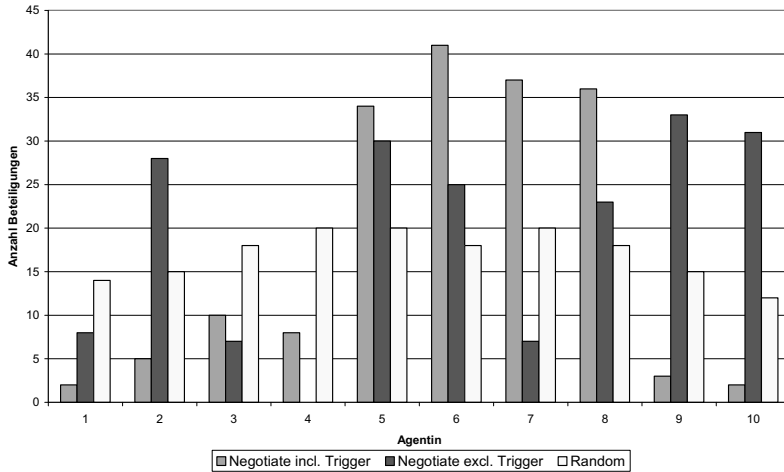
% =====
% parameters for the simulation tools
% =====
continue(no).
disp_file('d:/simsop/negotiate/dispfile').
result_file('d:/simsop/negotiate/rest/results_60_50').
belief_file('d:/simsop/negotiate/rest/belief_60_50').
para_file('d:/simsop/negotiate/rest/para_60_50').
stat_file('d:/simsop/negotiate/rest/stat_60_50').

% =====
% display colors
% =====
fact(colour([0,0,0])).
fact(colour([0,65500,0])).
fact(colour([65000,0,65000])).
fact(colour([65500,65500,0])).
fact(colour([0,30000,65500])).
fact(colour([40000,40000,40000])).
fact(colour([65000,30000,30000])).
fact(colour([20000,55000,55000])).
fact(colour([65000,65000,65000])).
fact(colour([45000,20000,45000])).
```

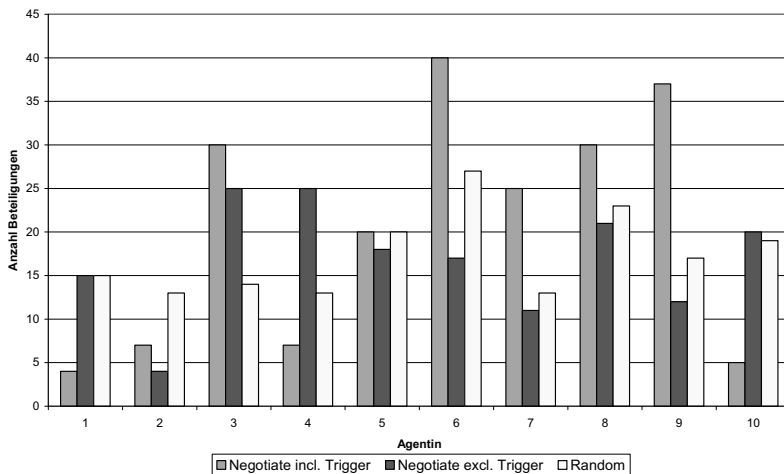
C - Muster der Gruppenentwicklung

Kernbildung

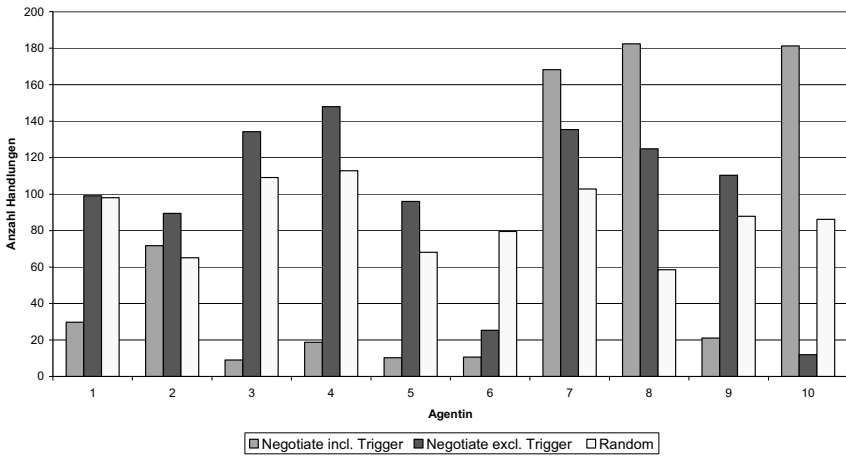
Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G3



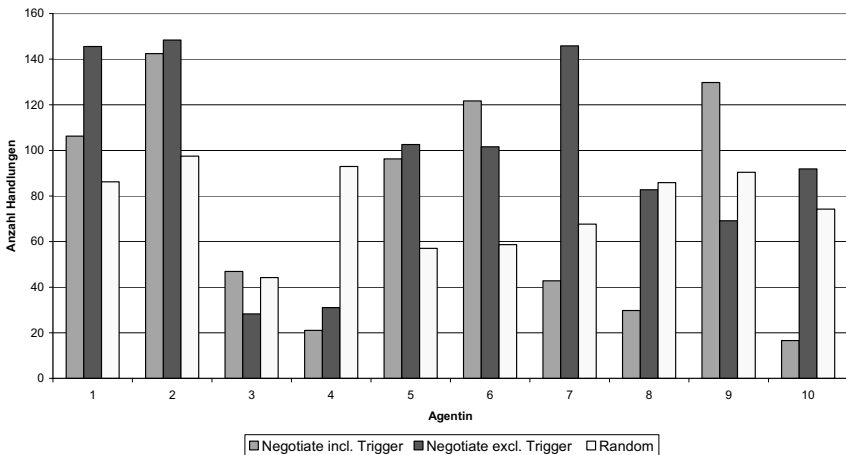
Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G7



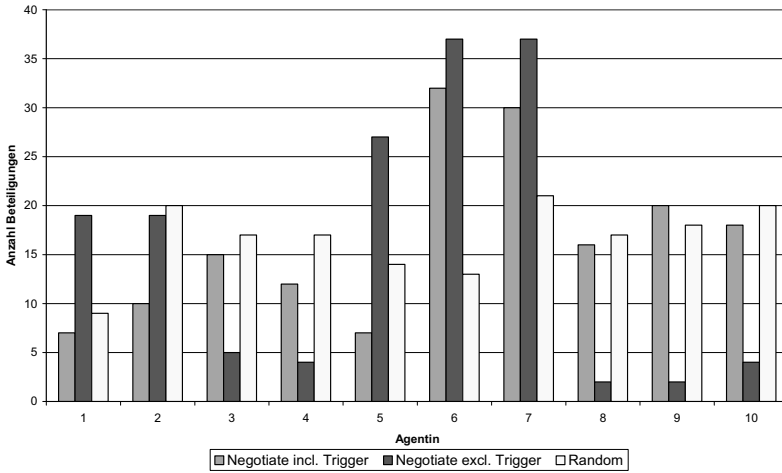
**Durchschnittliche Anzahl der Beteiligung an Gruppenhandlungen
in 50 Simulationsläufen über 200 Zeitpunkte - G18**



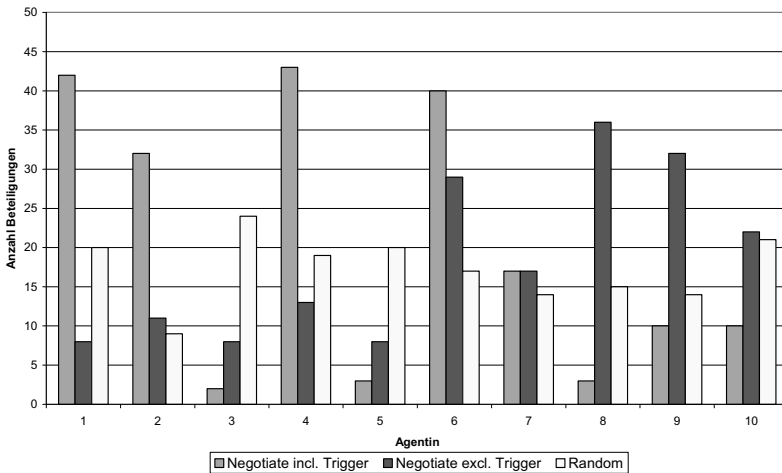
**Durchschnittliche Anzahl der Beteiligung an Gruppenhandlungen
in 50 Simulationsläufen über 200 Zeitpunkte - G11**



Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G15

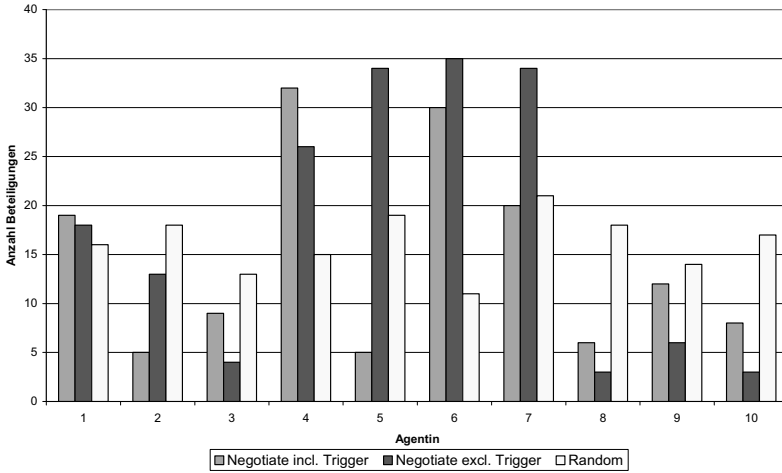


Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G19

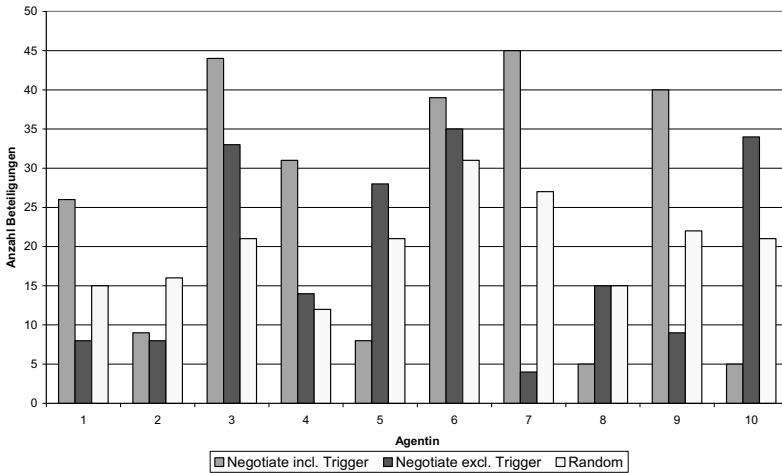


Kerne mit gleichen Agenten in verschiedenen Regelmengen

Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G16

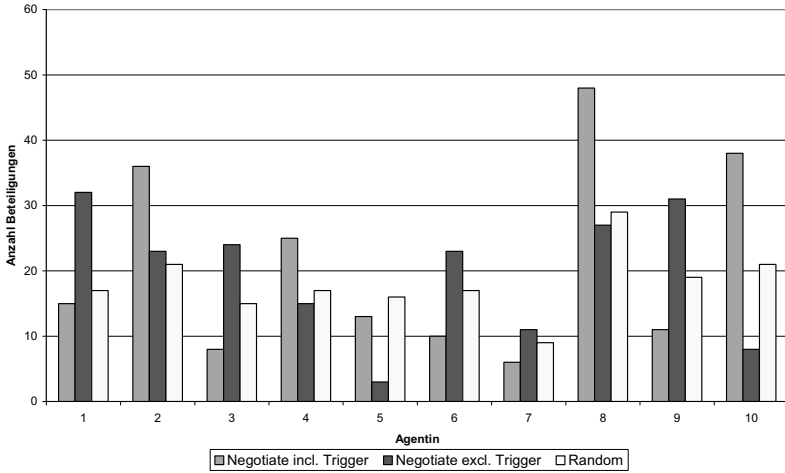


Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G10

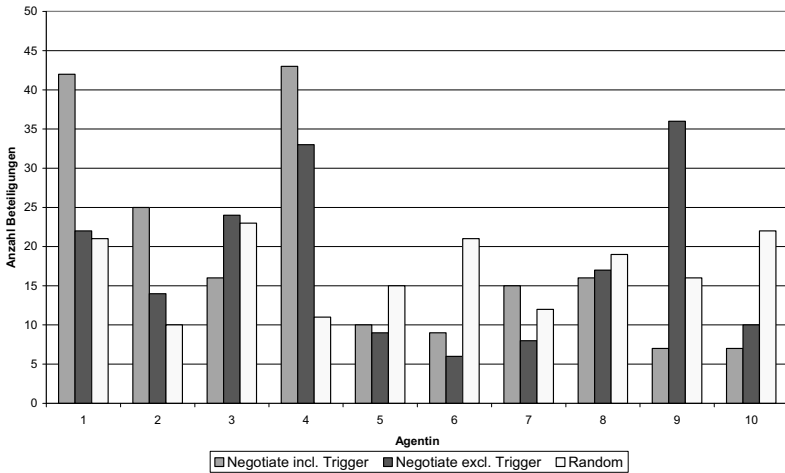


Anführerinnen

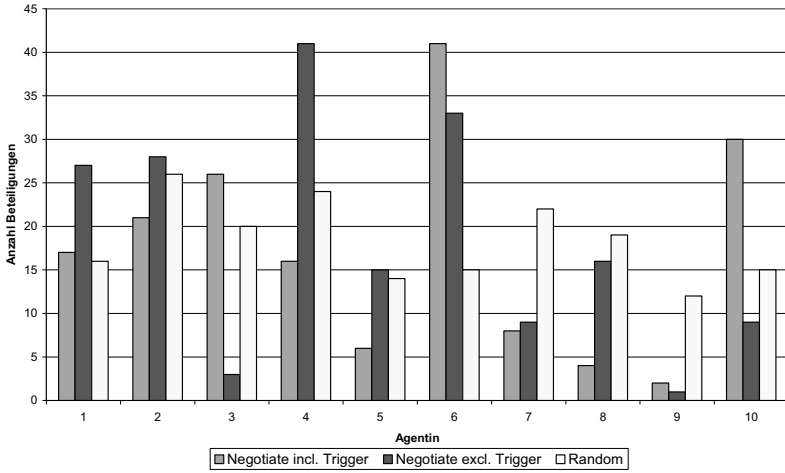
Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G20



Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G14

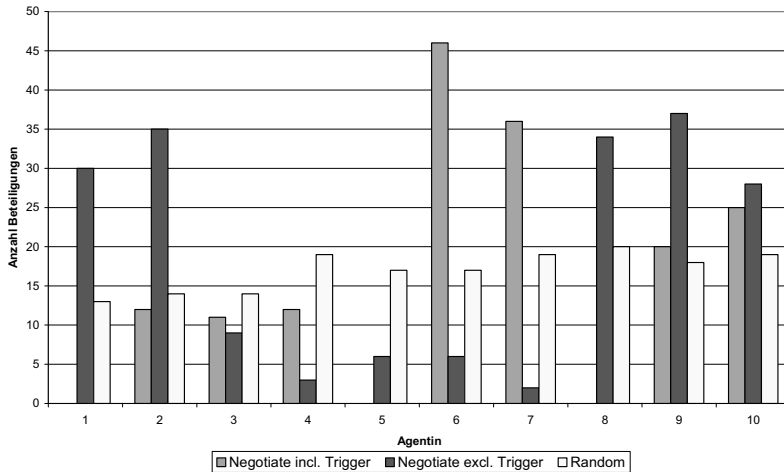


Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G5



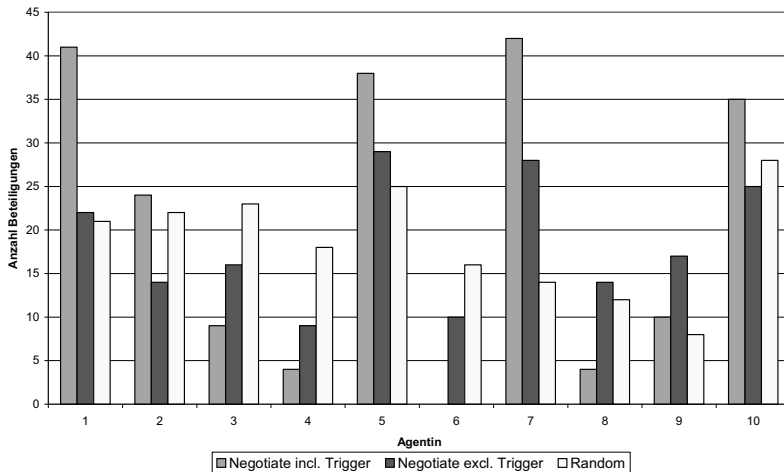
Fraktionsbildung

Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G17

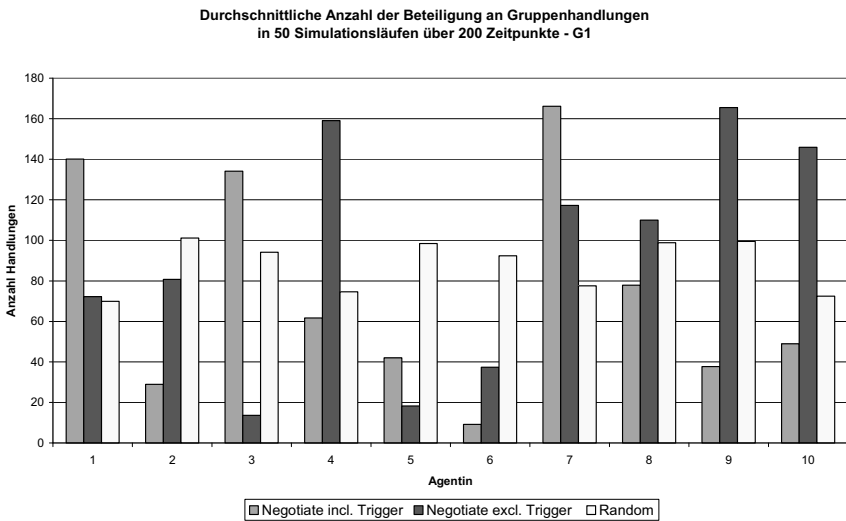
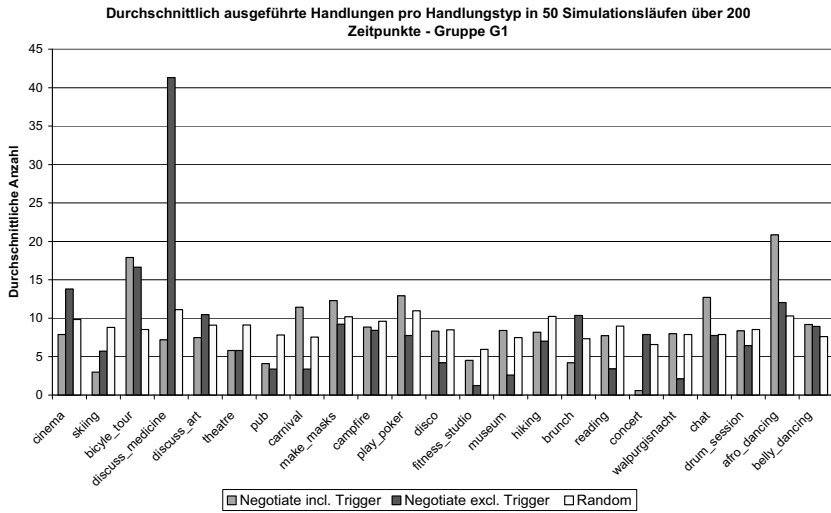


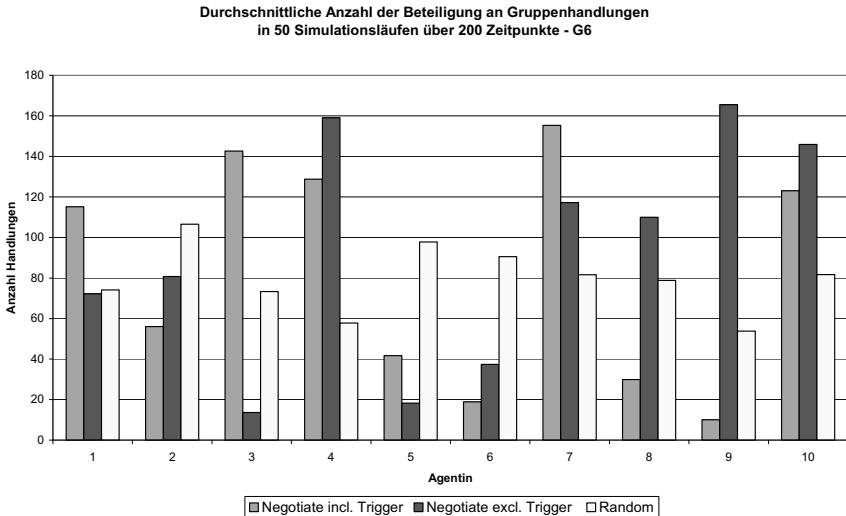
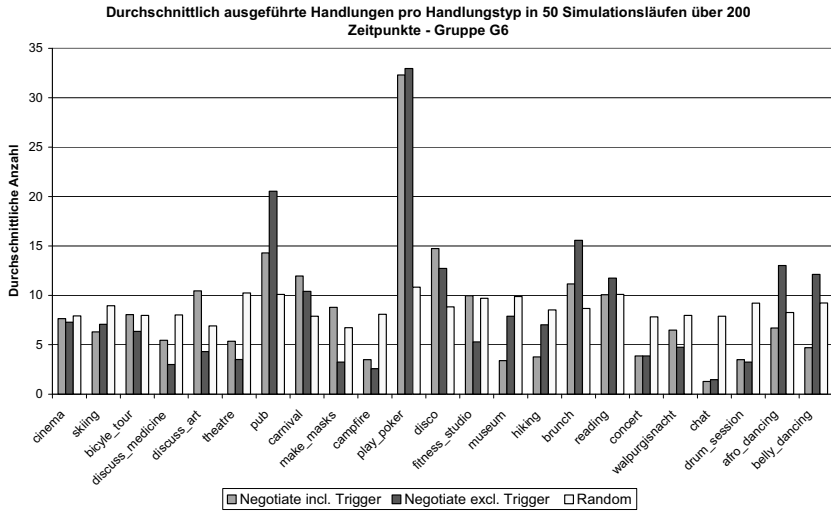
Isolation

Anzahl der Beteiligungen an Handlungen zum Zeitpunkt 200 in 50 Simulationsläufen - Gruppe G2



Spezialistinnen





Literatur

- Axelrod, Robert (1997a): *The Complexity of Cooperation*. Chichester: Princeton University Press
- Axelrod, Robert (1997b): *Advancing the Art of Simulation in the Social Sciences*. In: Conte et al (1997): 21-40.
- Balzer, Wolfgang (1993): *Soziale Institutionen*. Berlin, New York: de Gruyter
- Balzer, Wolfgang (1997): *Die Wissenschaft und ihre Methoden - Grundsätze der Wissenschaftstheorie*. Freiburg, München: Verlag Karl Alber
- Balzer, Wolfgang (2000): *SMASS: A Sequential Multi-Agent System for Social Simulation*. in Suleiman et al. (2000): 65-82.
- Balzer, Wolfgang: *Searle on Social Institutions: A Critique*. In: *dialectica* 56. 2002. 195-211.
- Balzer, Wolfgang, Karl R. Brendel und Solveig Hofmann: *Bad Arguments in the Comparison of Game Theory and Simulation in Social Studies*. In: *Journal of Artificial Societies and Social Simulation* 4 (2). 2001.
- Balzer, Wolfgang, C. Ulises Moulines und Joseph D. Sneed (1987): *An Archetectonic for Science*. Dordrecht: D. Reidel.
- Balzer, Wolfgang / Tuomela, Raimo (1997a): *A Fixed Point Approach to Collective Attitudes*. In: *Holmström-Hintikka/Tuomela* (1997): 115-142.
- Balzer, Wolfgang / Tuomela, Raimo (1997b): *The Structure and Verification of Plan-Based Joint Intentions*. In: *International Journal of Cooperative Information Systems* 6. 1997: 3-26.
- Balzer, Wolfgang / Tuomela, Raimo (2001): *Social Institutions, Norms and Practices*. In Conte/Dellorcas: (2001): *Social Order In Multiagent Systems*. Dordrecht: Kluwer Academic Publishers.
- Balzer, Wolfgang / Tuomela, Raimo (2003): *Collective Intentions and the Maintenance of Social Practices*. In: *Autonomous Agents and Multi-Agent Systems* 6 (1). 2003: 7-33.
- Benninghaus, Hans (1982): *Deskriptive Statistik*. Stuttgart: B. G. Teubner
- Benninghaus, Hans (2001): *Einführung in die sozialwissenschaftliche Datenanalyse*. München: Oldenbourg
- Bourdieu, Pierre (1980): *The Logic of Practice*. Cambridge: Cambridge University Press
- Bourdieu, Pierre (1987): *Sozialer Sinn, Kritik der theoretischen Vernunft*. Frankfurt am Main, Suhrkamp
- Bourdieu, Pierre (1997): *Selection from the Logic of Practice*. In: *Schrift* (1997): 190-230.
- Bowen, Kenneth C. (1978): *Research Games - An approach to the study of decision processes*. London: Taylor & Francis
- Brassel, Kai-H., Michael Möhring, Elke Schumacher und Klaus G. Troitzsch (1997): *Can Agents Cover All the World*. In: Conte et al. (1997): 55-72
- Bratman, Michael E. (1987): *Intention, Plans and Practical Reason*. Cambridge, Massachusetts, London, England: Harvard University Press
- Bratman, Michael E. (1993): *Shared Intention*. In: *Ethics* 104. 1993: 97-113.

- Bratley, Paul, Bennett L. Fox und Linus E.E. Schrage (1987): A Guide to Simulation, 2nd Edition. New York: Springer-Verlag
- Carnap, Rudolf (1939): Foundations of Logics and Mathematics. In: International Encyclopedia of Unified Science, vol. 1, no. 3. Chicago: The University of Chicago Press
- Carnap, Rudolf (1948): Meaning and Necessity. Chicago: The University of Chicago Press
- Casti, John L. (1997): Would-be Worlds - How Simulation is Changing the Frontiers of Science. New York, Chichester, Brisbane, Toronto, Singapore, Weinheim: John Wiley & Sons, Inc.
- Chattoe Edmund, Nicole J. Saam und Michael Möhring: Sensitivity Analysis in the Social Sciences: Problems and Prospects. In: Suleiman et al. (2000): 243-273.
- Chellas, Brian F. (1980): Modal Logic. New York, New Rochelle, Melbourne, Sydney: Cambridge University Press
- Chmura Thorsten / Pitz Thomas (2007): An Extended Reinforcement Algorithm for Estimation of Human Behaviour in Experimental Congestion Games. In: Journal of Artificial Societies and Social Simulation 10(2)1, 2007.
- Clocksins, William F. / Mellish, Christopher S. (1987): Programming in Prolog, 3rd Edition. Berlin, Heidelberg, New York: Springer-Verlag
- Cohen, Philip R., Jerry Morgan und Matha E. Pollack (Hrsg.) (1990): Intentions in Communication. Cambridge, London: The MIT Press
- Conte, Rosaria / Dellorcas, Chrysantos (Hrsg.) (2001): Social Order In Multiagent Systems. Dordrecht: Kluwer Academic Publishers.
- Conte, Rosaria, Nigel Gilbert und Jaime S. Sichman (1998): MAS and social simulation: A Suitable Commitment. In Sichman et al. (1998): 1-9
- Conte, Rosaria, Rainer Hegselmann und Pietro Terna (Hrsg.) (1997): Simulating Social Phenomena. Berlin, Heidelberg, New York: Springer-Verlag, 1997.
- Darwiche, Adnan / Pearl, Judea (1996): On the logic of iterated belief revision. Artificial Intelligence 89, Elsevier, 1996.
- Doran, Jim (2000): Questions in the Methodology of Artificial Sciences. In: Suleiman et al. (2000): 17 – 25.
- Doyle, Jon, Erik Sandewall und Pietro Torasso (Hrsg.) (1994): Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94). May 24-27, 1994. Bonn, Germany: Morgan Kaufmann
- Ellis, Brian (1979): Rational Belief Systems. Oxford, Basil Blackwell.
- Engel, Andreas / Möhring, Michael (1995): Der Beitrag der sozialwissenschaftlichen Informatik zur sozialwissenschaftlichen Modellbildung und Simulation. In: Gsänger /Klawitter (1995): 38-59
- Fagin, Ronald, Joseph Y. Halpern, Yoram Moses und Moshe Y. Vardi (1996): Reasoning about knowledge. Cambridge, Massachusetts, London, England: The MIT Press
- Fagin, Ronald / Halpern Joseph Y.: Belief, Awareness, and Limited Reasoning (1987) Artificial Intelligence 34. 1987. 39 - 76
- Forrester, Jay W. (1980): Principles of Systems 2nd preliminary edition, Cambridge, Massachusetts, London, England, The MIT Press
- Forrester, Jay W. (1971): Counterintuitive Behaviour of Social Systems. Technology Review 733. 1971, Updated March 1995. 52-68
- Gabbay, Dov. M., Christopher J. Hogger und John A. Robinson (Hrsg.) (1995): Handbook of Logic in Artificial Intelligence and Logic Programming 4. Oxford: Clarendon Press
- Gärdenfors, Peter (1988): Knowledge in Flux: Modeling the Dynamics of Epistemic States. Cambridge, Massachusetts, London, England: The MIT Press

- Gärdenfors, Peter / Rott, Hans (1995): Belief revision. In Gabbay et al. (1995)
- Georgeff, Michael, Barney Pell, Martha Pollack, Milind Tambe und Michael Wooldridge (1991): The Belief-Desire-Intention Model of Agency. In: Müller et al (1991).
- Giddens, Anthony (1984): *The Constitution of Society*. Cambridge: Polity Press.
- Gilbert, Margaret (1992): *On Social Facts*. Princeton, New Jersey: Princeton University Press
- Gilbert, Margaret (2000): *Sociality and Responsibility: New Essays in Plural Subject Theory*. Lanham: Rowman & Littlefield
- Gilbert, Nigel (1996): Holism, Individualism and Emergent Properties: An Approach from the Perspective of Simulation. In: Hegselmann et al. (1996).
- Gilbert, Nigel / Troitzsch, Klaus G. (2005): *Simulation for the Social Scientist - second edition*. Maidenhead: Open University Press.
- Gilbert, Nigel / Troitzsch, Klaus G. (1999): *Simulation for the Social Scientist*. Buckingham, Philadelphia Open University Press
- Goldspink, Chris: Methodological Implications of Complex Systems Approaches to Sociality: Simulation as a foundation for knowledge. In: *Journal of Artificial Societies and Social Simulation* 5(1)3. 2002.
- Goldszmith, Moses / Pearl, Judea: Qualitative Probabilities for Default Reasoning. In: *Artificial Intelligence* 84. 1996. 57-112.
- Gsänger, Matthias / Klawitter, Jörg (Hrsg.) (1995): *Modellbildung und Simulation in den Sozialwissenschaften*. Dettelbach: Verlag Dr. Josef H. Röhl
- Halpern, Joseph Y. (1995): Reasoning about knowledge: a survey: in Gabbay, et al. (1995): 1-34.
- Halpern, Joseph Y. / Moses, Yoram: A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence* 54 (3). 1992. 319-379.
- Hansson, Sven O. (1994): *A Textbook of Belief Dynamics*. Uppsala: Selbstverlag.
- Hartmann, Stephan (1996): The World as a Process – Simulations in the Natural and Social Sciences. In Hegselmann et al. (1996).
- Hegselmann, Rainer, Ulrich Mueller und Klaus G. Troitzsch (Hrsg.) (1996): *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*. Dordrecht: Kluwer Academic Publishers
- Hindriks, Frank (2002): Social Ontology, Collective Intentionality and Ockhamian Skepticism. In: Meggle (2002a): 125-149
- Hintikka, Jaakko (1962): *Knowledge and Belief*. Ithaca, New York: Cornell University Press
- Hörz, Herbert, Rolf Löther und Siegfried Wollgast (Hrsg.) (1978): *Philosophie und Naturwissenschaften - Wörterbuch zu den Philosophischen Fragen der Naturwissenschaften*. Berlin: Dietz Verlag
- Hofmann, Solveig (1997): *Maschinelle Erzeugung und Ableitung von Überzeugungssystemen*. München: München: Magisterarbeit an der Ludwig-Maximilians-Universität
- Hofmann, Solveig (2002): The Social Practice of a Women's Group: A First Simulation. In Meggle (2002a): 151-165.
- Hofmann, Solveig, Pekka Mäkelä, Thomas Pitz und Thorsten Chmura (2002): How Can We Avoid Traffic Jams? - Simulating the Social Practice of Joint Ride. In Urban (2002): 50-54.
- Holmström-Hintikka, Ghita / Tuomela, Raimo (Hrsg.) (1997): *Contemporary Action Theory 2*. Dordrecht: Kluwer Academic Publishers.
- Horton, Graham (Hrsg.) (2004): *Networked simulations and simulated networks (18th European simulation multiconference, Magdeburg, Germany, 13th- 16th June 2004)*. Erlangen: SCS Publishing House

- Jennings, Nicholas R., Katia Sycara und Michael Wooldridge: A Roadmap of Agent Research and Development. In: *Journal of Autonomous Agents and Multi-Agent Systems* 1 (1). 1998. 7-36.
- Katsuno, Hirofumi / Mendelzon, Albert O.: Propositional Knowledge, Base Revision and Minimal Change, *AI* 52,1991, S.264-296.
- Klein, Ernest (1967): *A Comprehensive Etymology Dictionary of the English Language*. Amsterdam, London, New York: Elsevier Publishing Compagny
- Klügl, Franziska (2001): *Multiagentensimulation: Konzepte, Werkzeuge, Anwendungen*. München: Addison-Wesley Verlag
- Klüver, Jürgen, Christina Stoica und Jörn Schmidt (2006): *Computersimulationen und soziale Einzelfallstudien. Eine Einführung in die Modellierung des Sozialen*. Herdecke – Bochum: W31 Verlag
- Lauth, Bernhard / Sareiter, Jamel (2005): *Wissenschaftliche Erkenntnis - Eine ideengeschichtliche Einführung in die Wissenschaftstheorie*. Paderborn: mentis Verlag
- Lenzen, Wolfgang (1980): *Glauben, Wissen und Wahrscheinlichkeit: Systeme der epistemischen Logik*. Wien, New York: Springer-Verlag
- Lux-Endrich, Astrid / Wachsmann, Anne (Hrsg.) (1998): *Konstruierte Wirklichkeiten - Modellbildung und Simulation in den Wissenschaften*. Tutzing Materialie 82, Evangelische Akademie Tutzing
- Mathiesen, Kay (2002): Searle, Collectiv Intentions, and Individualism. In: Meggle (2002a): 185-204.
- Meggle, Georg (Hrsg.) (2002a): *Social Facts & Collective Intentionality*. Frankfurt am Main: Dr. Hänsel-Hohenhausen AG.
- Meggle, Georg (2002b): *Gemeinsamer Glaube und Gemeinsames Wissen*, in Siebel (2002): 12-20
- Meggle, Georg (2002c): *Mutual Knowledge and Belief*. In Meggle (2002a): 205-223
- van der Meyden, Ron (1994): *Mutual Belief Revision*. In Doyle et al (1994): 595-606.
- Mittelstraß, Jürgen (Hrsg.) (1995): *Zyklusopädie Philosophie und Wissenschaftstheorie 3*. Stuttgart, Weimar: J.B.Metzler
- Mittelstraß, Jürgen (Hrsg.) (2005): *Zyklusopädie Philosophie und Wissenschaftstheorie 2*. Stuttgart, Weimar: J.B.Metzler
- Möhring, Michael / Ostermann, Raphael (1996): *MIMOSE - Eine funktionale Sprache zur Beschreibung und Simulation individuellen Verhaltens in interagierenden Populationen*. Koblenz: Druckerei der Universität Koblenz-Landau
- Mora, Michael da Costa, José Lopes, Rosa Viccari und Helder Coelho (1999): *BDI Models and Systems: Bridging the Gap*. In Müller et al (1999): 11-28.
- Müller, Jörg P., Munindar P. Singh und Anand S. Rao (1999): *Lecture notes in artificial intelligence (1555), Intelligent Agents V*. Berlin, Heidelberg, New York: Springer-Verlag
- Partee, Barbara H., Alice ter Meulen und Robert E. Wall (1993): *Mathematical Methods in Linguistics*. Dordrecht, Boston, London: Kluwer Academic Publishers
- Pitz, Thomas (2000): *Anwendung Genetischer Algorithmen auf Handlungs bäume in Multiagentensystemen zur Simulation sozialen Handelns*. Dissertation: Ludwig-Maximilians-Universität, München.
- Rao, Anand S. / Georgeff, Michael: *Decision procedures for BDI logics*. In: *Journal of Logic and Computation* 8(3). 1998. 293-344.
- Rao, Anand S. / Wooldridge, Michael (1999): *Foundations of Rational Agency*. In: *Wooldridge / Rao* (1999): 1-100.
- Reich, Wendelin: *Reasoning About Other Agents: a Plea for Logic-Based Methods*. In: *Journal of Artificial Societies and Social Simulation* 7 (4). 2004.

- Ritter, Joachim / Gründer, Karlfried (Hrsg) (1989): *Historisches Wörterbuch der Philosophie* 7, P-Q. Basel: Wissenschaftliche Buchgesellschaft Darmstadt
- Saam, Nicole (1998a): Modell - Simulation – Theorie. In: Lux-Endrich / Wachsmann (1998): 13 – 17.
- Saam, Nicole (1998b): Zur Konstruktion von Wirklichkeit – Computersimulation als Methode. In: Lux-Endrich / Wachsmann (1998): 65-79.
- Sandkühler, Hans J. (Hrsg.) (1999): *Enzyklopädie Philosophie*. Hamburg: Felix Meiner Verlag
- Sandu, Gabriel / Tuomela, Raimo: *Joint Action and Group Action Made Precise*. Synthese 105, 1996. 319-345.
- Schatzki, Theodore (1996): *Social Practices: A Wittgensteinian Approach to Human Activity and the Social*. Cambridge: Cambridge University Press
- Schelling, Thomas C.: Models of segregation. *American Economic Review* 59, 1969. 488-493
- Schelling, Thomas C.: Dynamic models of segregation. *Journal of Mathematical Sociology* 1(2). 1971. 143-186.
- Schild, Klaus (1999): On the Relationship between BDI Logic and Standard Logic of Concurrency. In Müller et al. (1999): 47-61.
- Sichman, Jaime S., Rosaria Conte and Nigel Gilbert (Hrsg.) (1998). *Lecture Notes In Artificial Intelligence: Vol. 1534: Proceedings of the First International Workshop on Multi-Agent Systems and Agent Based Simulation*. Berlin, Heidelberg, New York: Springer-Verlag
- Schoenemann, Thomas J.: The witch hunt as a culture change phenomenon. *Ethos*. 1975. 529-554.
- Schrift, Alan D. (Hrsg.) (1997): *The Logik of the Gift. Towards an Ethic of Generosity*. London, New York: Routledge
- Searle, John (1987): *Intentionalität - Eine Abhandlung zur Philosophie des Geistes*. Frankfurt am Main: Suhrkamp Verlag
- Searle, John (1990): *Collective Intentions and Actions*. In Cohen et al (1990): 401-415.
- Searle, John (1995): *The Construction of Social Reality*. London: The Penguin Press
- Siebel, Mark (Hrsg.) (2002): *Kommunikatives Verstehen..* Leipzig: Leipziger Schriften zur Philosophie 16
- Simon, Hartmut (1980): *Computer-Simulation und Modellbildung im Unterricht - Hochschuldidaktische Konzepte und Einsatzerfahrungen in den naturwissenschaftlichen Fächern Datenverarbeitung/Informatik im Bildungsbereich 3*, München, Wien: Oldenbourg Verlag
- Speck, Josef (1980): *Handbuch wissenschaftstheoretischer Begriffe*. Göttingen: Vandenhoeck & Ruprecht
- Stachowiak, Herbert (1973): *Allgemeine Modelltheorie*. Wien, New York: Springer-Verlag
- Stegmüller, Wolfgang (1973): *Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie, Bd. IV, Personelle und Statistische Wahrscheinlichkeit, Erster Halbband, Personelle Wahrscheinlichkeit und Rationale Entscheidung*. Berlin, Heidelberg, New York: Springer-Verlag
- Stegmüller, Wolfgang (1980): Hypothese. In Speck (1980): 284 – 287.
- Stegmüller, Wolfgang / Varga von Kibéd, Matthias (1984): *Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie, Bd. III, Strukturtypen der Logik*. Berlin, Heidelberg, New York: Springer-Verlag
- Sterling, Leon / Shapiro, Ehud (1994): *The Art of Prolog - Advanced Programming Techniques*, 2nd Edition, Cambridge, Massachusetts, London, England: The MIT Press.
- Suleiman, Ramzi, Klaus G Troitzsch and Nigel Gilbert (2000): *Tools and Techniques for Social Science Simulation*. Heidelberg, New York: Physica-Verlag

- Suppes, Patrick: A comparison of the meaning and uses of models in mathematics and the empirical sciences. *Synthese* 12. 1960. 287-301.
- Tarski, Alfred (1937): Einführung in die mathematische Logik und in die Methodologie der Mathematik. Wien: Springer
- Troitzsch, Klaus G. (1990): .Modellbildung und Simulation in den Sozialwissenschaften. Opladen: Westdeutscher Verlag
- Troitzsch, Klaus G. (1996): Simulation and Structuralism. In: Hegselmann et al. (1996): 183-207.
- Troitzsch, Klaus G. (1995): Mathematische Kalküle zur Bildung dynamischer Modelle in den Sozialwissenschaften. In: Gsänger /Klawitter (1995): 9-38
- Troitzsch, Klaus G. (1997): Social Science Simulation - Origins, Prospects, Purposes. In Conte et al. (1997): 41-54
- Troitzsch, Klaus G. (2004): Validating Simulation Models. In: Horton (2004): 265-270.
- Tuomela, Raimo: Group Beliefs. *Synthese* 91. 1992. 285-318.
- Tuomela, Raimo (1995): The Importance of Us. Stanford, California: Stanford University Press
- Tuomela, Raimo: Belief versus Acceptance. *Philosophical Explorations* 2. 2000. 122-137.
- Tuomela, Raimo (2003): The Philosophy of Social Practices - A Collective Acceptance View. Cambridge: Cambridge University Press
- Tuomela, Raimo / Balzer, Wolfgang: Collective Acceptance and Collective Social Notions, *Synthese* 117. 1999. 175-205.
- Tuomela, Raimo / Miller, Kaarlo: We-Intentions, *Philosophical Studies* 53. 1988. 115-137.
- Urban, Christoph (Hrsg.) (2002): 3rd Workshop on Agent-Based Simulation, April 7-9, 2002. Passau, Germany.
- Wielemaker, Jan (2006): SWI-Prolog 5.6, Reference Manual. University of Amsterdam, Dept. of Social Science Informatics
- Wielemaker, Jan / Anjewierden, Anjo (1997): Programming in XPC/Prolog. University of Amsterdam, Dept. of Social Science Informatics
- Wittenberg, Reinhard (1998): Computergestützte Datenanalyse. Stuttgart: Lucius & Lucius
- Wittgenstein, Ludwig (1958): Philosophical Investigation, 2. ed., Oxford: Blackwell
- Wooldridge, Michael (2000): Reasoning about Rational Agents. Cambridge, Massachusetts, London, England: The MIT Press.
- Wooldridge, M. / Jennings, Nicholas R.: Intelligent agents: Theory and practice. *The Knowledge Engineering Review* 10(2). 1995. 115-152.
- Wooldridge, Michael / Rao, Anand S. (Hrsg.) (1999): Foundations of Rational Agency. Dordrecht: Kluwer Academic Publishers
- Zeigler, Bernard P. (1976): Theory of Modelling and Simulation. New York: Wiley

Index

Absicht	
gemeinsame	51
geteilte	52, 72, 121
kollektive	51
vorausgehende	51
Ziel.....	52
Abstraktion	30
Actor-Liste.....	119, 121
Agenten	
Aktivierung	81
autonome	67
intelligente	66
rationale	67
Aktivierung	
aufsteigende Reihenfolge.....	120
zufällige Reihenfolge.....	119
Akzeptanz	
kollektive	44, 45
alethische Operatoren	57
All-Solution-Prädikate	86
Analyse-Tools.....	138
Anführerinnen.....	187
Anteil	
geteilte Intentionen	158
geteilte Überzeugungen	158
Anwendungen	
intendierte	32
atomarer Fakt.....	83
Ausgabedateien.....	87, 96
Dateinamen	147
Ausgabedaten.....	139
auslösende Bedingungen.....	74
Autonomie	66
Axiomenmenge	31
Backtracking	81
BDI-Logik.....	67
BDI-Struktur	67
Beteiligung	
durchschnittliche ...	173, 177, 181
Häufigkeit	172, 177, 181
Beteiligungen	
Anzahl.....	173, 177, 181
Blättermodus	
Einstellung	147
Blättern	147
Briefkasten	
persönlicher.....	123, 126
collective action	42
collective intention.....	51, 72
common belief	63
common knowledge.....	52, 63
Computermodell	30
Dateien.....	87
Dateinamen	89
Protokolldateien	96
Daten.....	31, 33
initiale	34
Datenbasis.....	84
Deliberation	68
Diagramm	
Aufbau	148
Differentialgleichungssysteme....	26
Discrete-Event-Modelle	47
dynamische Systeme.....	46
Eingabedatei.....	87
Eingabedaten.....	98

- Einstellung
 individuelle 106
 unbestimmte..... 131
- Einstellungen
 Änderung 46
 Dynamik 60
 epistemische..... 57, 59
 Generierung 93
 höherer Stufe..... 59
 initiale 93
 Menge 121, 122
- Elemente
 semantische..... 32
 syntaktische 32
- Entscheidungsfindung
 praktische..... 68
 epistemische Einstellungen..... 57
 epistemische Logik 56, 59
 epistemischer Input..... 60
 epistemischer Zustand 149
 epistemologische Theorie 60
- Erfolgsbewertung..... 118, 132
 Kriterium 164
 Steuerung..... 166
- Erfolgsrate 130
- Erfolgswahrscheinlichkeit 131
- Expansion 62
- Experiment..... 27
- Fähigkeiten
 soziale 66
- Fakt
 atomarer 83
- Fakten 33, 84
- Faktenbasis 83
 Bereinigung 95
 Manipulation..... 85
- Farbeinstellungen..... 139
- Fehler
 akzeptable 163
- Fehlerrate 165
- Fehlertoleranz 166
- Fenster..... 147
- formale Modelle..... 30
- Formalisierung 25
- Fraktionsbildung 188
- Frustrationsschwelle 166
- Funktor..... 84
- Gedankenexperiment 27, 190
- gemeinsame Absicht 51
- gemeinsame Handlung
 Auslösung 129
 Auswahl 118
 Bewertung..... 129
 Durchführung..... 118, 129
- gemeinsame Überzeugung
 Fixpunkt-Ansatz..... 64
 iterativer Ansatz..... 64
- gemeinsame Verbindlichkeit..... 54
- gemeinsames Wissen 54, 64
- Generierungsschicht..... 88
- Gesellschaften
 künstliche 67
- geteilte Absicht 52
- Glauben..... 57
- grafische Ausgabe..... 139, 140
- grafische Darstellung 148
- group intention 72
- Grund
 sozialer 42
- Gruppe 79, 142
 Entwicklung 117
 mögliche Entwicklungen 143
 Zusammenhalt..... 159
- Gruppen
 Äquivalenzklassen 142
- Gruppenentwicklung..... 143
- Gruppenhandlung
 Beteiligungen 184
 bevorzugte..... 127, 128
 Trigger 128

- Vorschlag..... 150
- Gruppenkern 185, 187
- Gruppenleben
- Erfolg..... 130
- Gruppenstärke..... 159, 164
- durchschnittliche..... 155, 165
- Stabilisierung..... 156
- Habitus..... 40
- Handlung
- bevorzugte 131, 132
- Bewertung..... 104
- Erfolg..... 75, 130
- Ergebnis..... 74
- Feedback..... 75
- gemeinsame 50
- gemeinschaftliche 42
- Häufigkeit..... 171, 174, 178
- individuelle 43, 50
- Initialzustand..... 74
- intendierte 124, 131, 132
- Modellierung..... 74
- nicht intendierte 133
- soziale..... 42
- Vorschlag..... 124, 150
- Wahlfreiheit..... 185
- Handlungen
- relative Häufigkeit 171, 175, 180
- Handlungen pro Handlungstyp . 183
- Handlungsablauf..... 153
- Handlungsabsicht..... 131
- Herausbildung..... 129
- Modellierung..... 72
- Handlungsabsichten..... 51, 106, 113
- Handlungskonzept
- aristotelisches..... 39
- Handlungsspektrum 189
- Handlungstheorie..... 50
- Handlungstypen 93, 101
- Liste..... 111
- Historie 133
- Historieneintrag 134
- Hypothese 31, 33, 34
- I-mode..... 45
- individuelles Verhalten 49
- Inferenzbasis 84
- Manipulation..... 85
- initiale Intentionen
- Generierung 100
- Initialisierungsphase 98
- Input..... 33
- epistemischer 60
- Instanz..... 84
- Institutionen
- soziale 45
- intendierte Anwendungen 32
- Intention
- Inhalt..... 105
- Prolog-Fakt 104
- intentionaler Zustand 49
- Intentionalität
- kollektive 44, 45
- Intentionen
- Generierung 100
- grafische Darstellung 148
- initiale 111
- Intentionsfakten 105
- Intersubjektivität 63
- joint Commitment 53
- joint intention..... 51
- Kernbildung..... 186, 187
- Kernsimulation..... 88
- Klauselmenge..... 81
- kollektive Absicht..... 51
- kollektive Akzeptanz 44, 45
- kollektive Intentionalität 44, 45
- kollektives Verhalten 49
- Konfigurationsparameter 90, 93
- Konstruktivismus
- sozialer..... 41
- Kontraktion 62

- künstliche Gesellschaften 67
- Laufzeitparameter 90, 92
- Logik
 epistemische..... 56, 59
- Mailbox..... 121, 124
- means-end reasoning 68
- Methoden
 wissenschaftliche 27
- Misserfolg
 Bewertung..... 132
- Misserfolge
 akzeptable 135
 Anzahl..... 134
 aufeinanderfolgende 135, 136
- Modallogik..... 56, 59
- Modell
 Entwicklung..... 30
 Konstruktion 29, 31
 Konzept..... 28
- Modellbegriff..... 28, 29, 32
- Modellbildung 29
- Modelle
 formale..... 30
- Modellklasse
 get_random 120
 negotiate..... 123
- Modellklassen 31, 33
- Modellkonzept
 strukturalistisches 32
- Modelltheorie..... 31
- mögliche Entwicklungen
 Gruppe 143
- mögliche Welten..... 27, 57
- Multiagentensystem..... 67
- mutual belief..... 63, 73
- mutual knowledge..... 63
- Nachricht 123, 124
- Nachrichtenübermittlung 65
- Name..... 83
- negativer Trigger 178
- Normen 79
- Obligation Criterion 55
- OK-Button 147
- omniscience problem 61
- Operatoren
 aletische 57
- Originalsystem 29, 30
- Output 33
- Parameter
 sensitiver 163
- Parameterinitialisierung 144
- Parameterwerte 90
- Performanz..... 82, 140
- Performativität 45
- Permission Criterion 55
- persönliche Verbindlichkeit 53
- Plan 68
- Planung 52
- Plural Subject..... 53
- Pluralistic Ignorance 110
- Poesis 39
- positiver Trigger 178
- Prädikat 84
 Definitionsbereich..... 84
 dynamisches..... 85
- praktische Entscheidungsfindung 68
- Praxis
 Änderung 46
 Aufgabe..... 138
 Aufrechterhaltung 46
 Etymologie..... 38
 Fortführung..... 135
 gesellschaftliche..... 39
- Praxisbegriff
 aristotelischer 38
 marxistischer 39
- Praxiskonzept
 marxistisches..... 40
- Praxisphilosophie..... 40
- Proaktivität..... 66

- Programmablauf 33
 Programmierung
 deklarative 82
 Programmkonstanten 90, 97
 Programmpaket..... 88
 Programmparameter 90
 Programmregeln
 Korrektheit..... 146
 Prolog..... 81
 Prolog-Shell 83
 Proposition..... 59
 Protokolldatei..... 117
 Protokolldateien..... 88, 89, 139
 Prozentsatz
 geteilte Intentionen 154
 geteilte Überzeugungen 154
 Public Mailbox..... 121, 124, 128
 Rationalitätskriterium 61
 Reaktivität..... 66
 Reduktionsprinzip..... 59
 Regelmenge 143, 144
 Regelmengen
 Vergleich 161, 167
 Regeln..... 33, 84
 Rekursion
 Terminierung 113, 114, 116
 Resolutionskalkül 81
 Revision..... 62
 Schwelle
 Ermittlung..... 133
 Schwellenbewertung..... 133, 134
 Seed 144, 146
 Semantik 32
 sensitiver Parameter..... 163
 shared intention..... 51
 shared social reason 42
 Simulation
 Anwendungsbereiche..... 21, 23
 Etymologie..... 27
 Methode 24, 27, 30
 Simulationsergebnisse
 Bewertung 141
 Sicherung 96
 Strukturierung 169
 Simulationsfolge
 ausführen..... 98
 Simulationsfolgen
 Einteilung..... 168
 Gegenüberstellung 184
 Vergleich..... 183
 Simulationslauf
 Steuerung 116
 Terminierung 138
 Simulationsprogramm
 Ausgabe 139
 Simulationsprotokoll..... 146
 SMASS 81
 soziale Handlung..... 42
 soziale Institutionen 45
 soziale Praktiken
 Einteilung..... 43
 soziale Praxis
 Aufgabe..... 117
 Beispiel 80
 Beteiligung..... 75
 Bewertung..... 75
 Muster 117
 Rahmen 76
 Schwelle..... 130
 Soziale Simulation
 Fachbereiche 24
 soziale Systeme..... 42
 Modellierung..... 26
 Sozialität 45
 Spezifikation 30, 33
 Sprache
 deklarative..... 82
 objektorientierte 82
 Stabilisierung 155
 Startbedingungen 187

- Struktur 84
- System
- reales 29
- Systeme
- dynamische 46
 - soziale 42
- Systemprädikate 85
- Teilhandlungen 49
- Term
- Argumente 84
 - zusammengesetzter 84
- Testeinstellungen
- Verifikation 145
- Theorie 31
- epistemologische 60
 - Grundbegriffe 31
 - semantische 61
 - wissenschaftliche 31
- Trigger 146, 162
- Handlung 74
 - Intention 74
 - negativer 126, 178
 - positiver 124, 126, 178
- Überzeugung
- gegenseitige 63
 - gemeinsame 63
 - Inhalt 105
 - Stufe 116
 - unbestimmte 108
- Überzeugungen
- erster Stufe 105
 - höherer Stufe 105
 - Konsistenz 58
 - widersprüchliche 110
- Überzeugungsbasen 62
- Überzeugungsbasis 105, 116
- Aktualisierung 126
 - Generierung 110, 113
 - grafische Darstellung 149
 - Protokoll 139
- Überzeugungsmenge
- Änderung 110
 - intersubjektive Konsistenz 109
 - subjektive Konsistenz 108
- Überzeugungsrevision 151
- Überzeugungssatz
- Prolog-Fakt 105
 - Struktur 105
- Überzeugungssystem 60
- Unifizierung 81
- Validierung 141
- Variable 84
- Variableninstantiierung 97
- Variablennamen 84
- Verbindlichkeit
- gemeinsame 54
 - persönliche 53
 - Urheber 54
- Verhalten
- individuelles 49
 - kollektives 49
- Verhaltensregeln 81
- Verhandlung 123
- Verifikation 141
- Verzeichnisnamen 96
- Verzeichnispfad 89, 147
- Verzeichnisse 96
- Vorschlag 123
- Auswahl 125, 127
- Vorschläge
- Bewertung 126
 - Feedback 107
 - zufällige Auswahl 121
- Wahrheit 57, 61
- Welten
- mögliche 57
- w-e-mode 45
- Wir-Einstellung 45
- Wir-Intention 73
- Wissen 57

gegenseitiges.....	63	Ziel.....	68
gemeinsames.....	54, 63, 64	Zielsystem.....	29
wissenschaftliche		Zufallsauswahl.....	86, 112
Methode.....	27	Zufallszahl.....	112, 144
Theorie.....	31	Zufallszahlengenerator.....	98, 144
Zeitabschnitt.....	47	Initialisierung.....	86, 146
Zeitpunkt.....	47	Zusammenhalt.....	161
Nachfolger.....	47	Zustand	
Zeitpunkte		epistemischer.....	60
Nachbarschaft.....	47	intentionaler.....	49
Zeitskalen.....	47	psychologischer.....	60