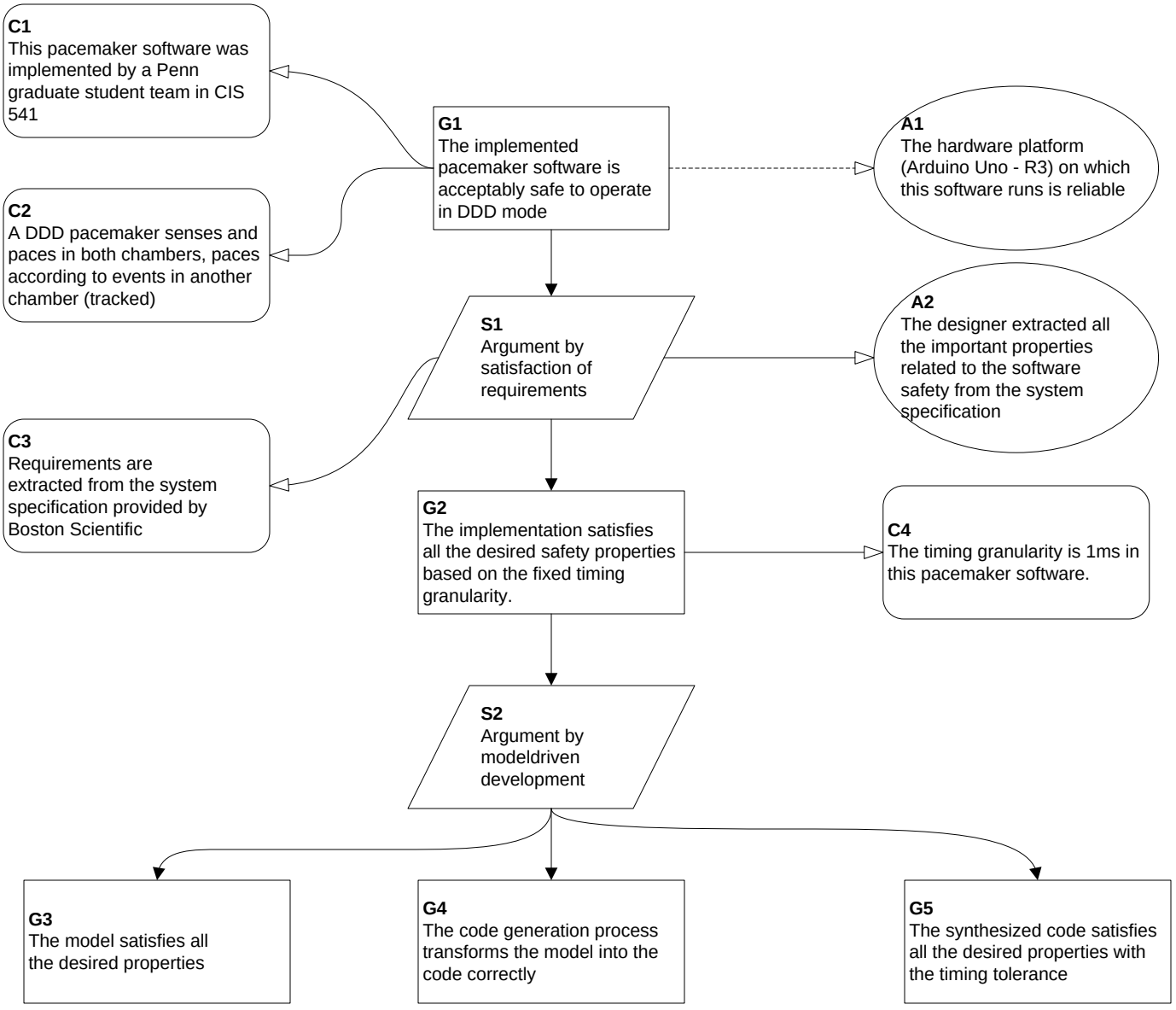
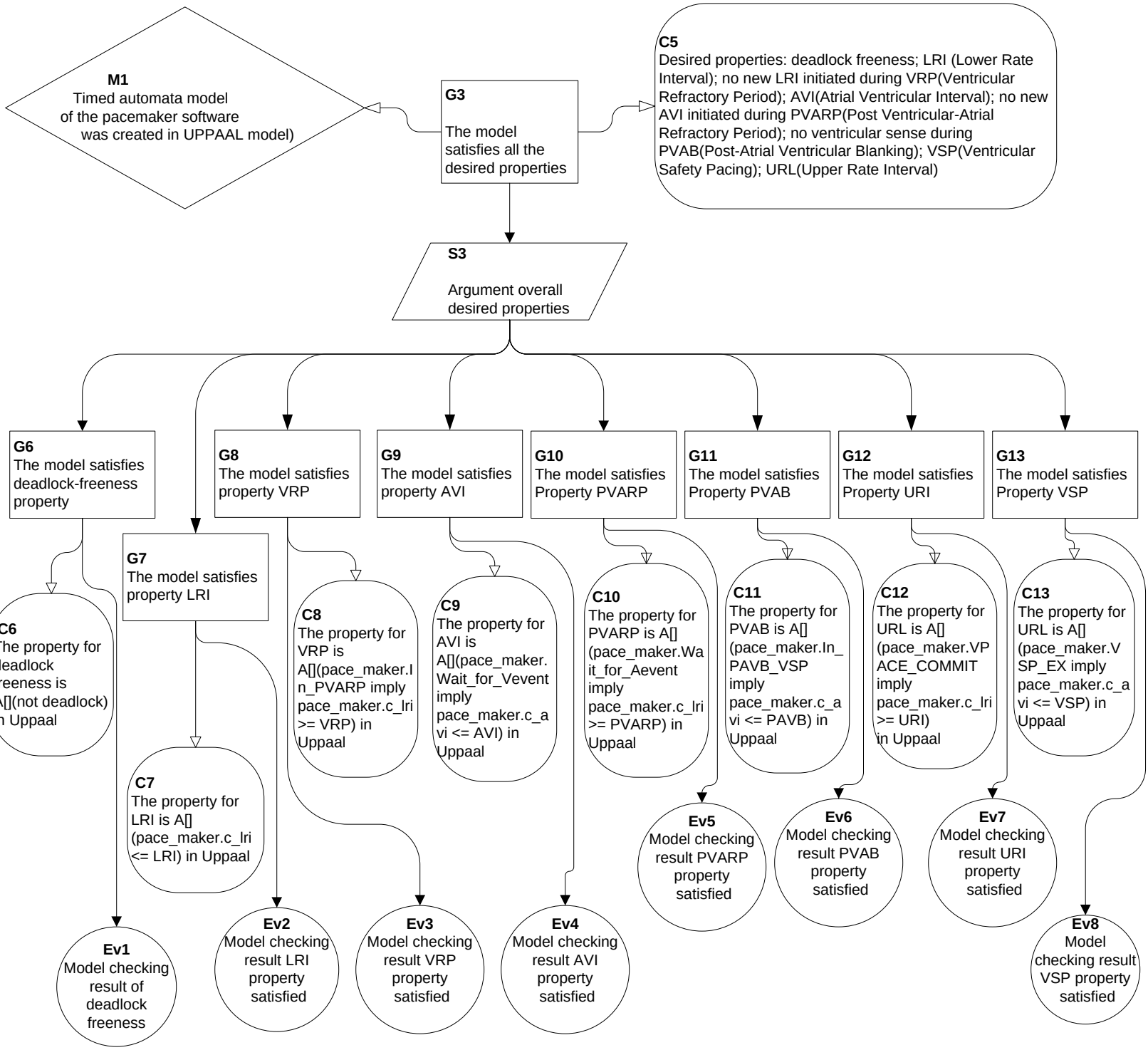
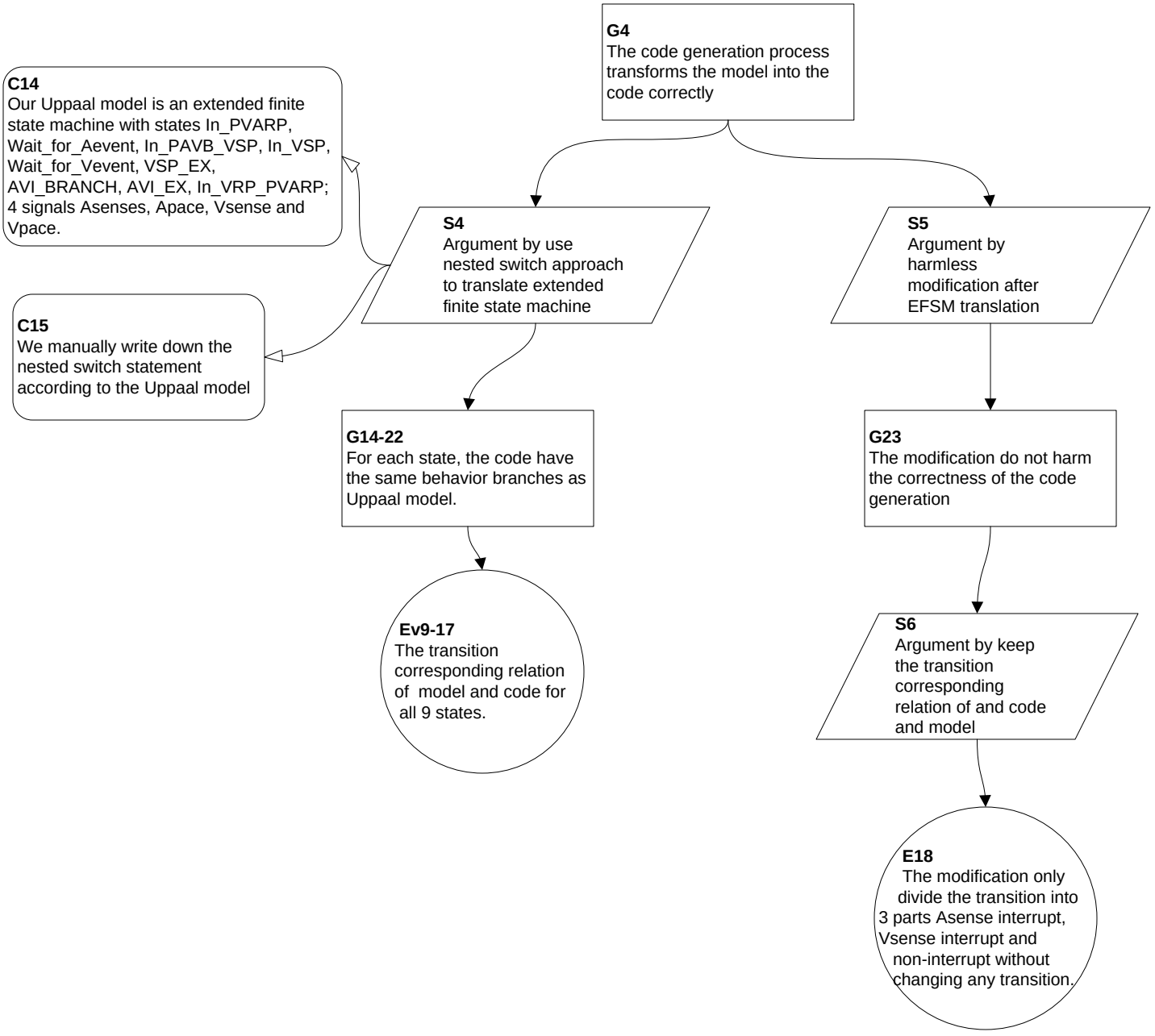
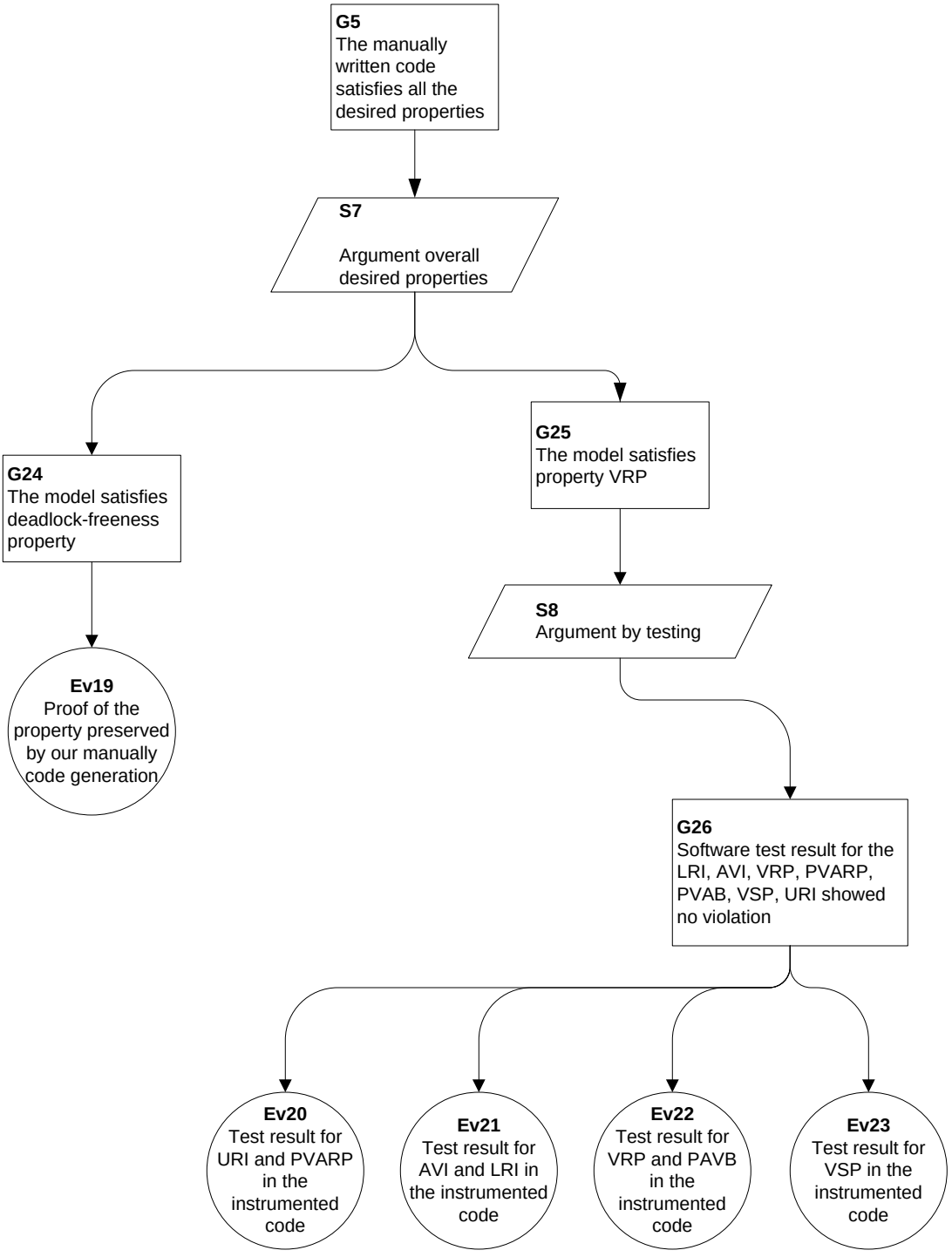

C1
This pacemaker software was implemented by a Penn graduate student team in CIS 541
C2
A DDD pacemaker senses and paces in both chambers, paces according to events in another chamber (tracked)
C3
Requirements are extracted from the system specification provided by Boston Scientific
G1
The implemented pacemaker software is acceptably safe to operate in DDD mode
A1
The hardware platform (Arduino Uno - R3) on which this software runs is reliable
S1
Argument by satisfaction of requirements
A2
The designer extracted all the important properties related to the software safety from the system specification
G2
The implementation satisfies all the desired safety properties based on the fixed timing granularity.
C4
The timing granularity is 1ms in this pacemaker software.
S2
Argument by modeldriven development
G3
The model satisfies all the desired properties
G4
The code generation process transforms the model into the code correctly
G5
The synthesized code satisfies all the desired properties with the timing tolerance


M1
Timed automata model of the pacemaker software was created in UPPAAL model)
G3
The model satisfies all the desired properties
C5
Desired properties: deadlock freeness; LRI (Lower Rate Interval); no new LRI initiated during VRP(Ventricular Refractory Period); AVI(Atrial Ventricular Interval); no new AVI initiated during PVARP(Post Ventricular-Atrial Refractory Period); no ventricular sense during PVAB(Post-Atrial Ventricular Blanking); VSP(Ventricular Safety Pacing); URL(Upper Rate Interval)
S3
Argument overall desired properties
G6
The model satisfies deadlock-freeness property
G7
The model satisfies property LRI
G8
The model satisfies property VRP
G9
The model satisfies property AVI
G10
The model satisfies Property PVARP
G11
The model satisfies Property PVAB
G12
The model satisfies Property URI
G13
The model satisfies Property VSP
C6
The property for deadlock freeness is A[](not deadlock) in Uppaal
C7
The property for LRI is A[] (pace_maker.c_lri <= LRI) in Uppaal
C8
The property for VRP is A[](pace_maker.In_PVARP imply pace_maker.c_lri >= VRP) in Uppaal
C9
The property for AVI is A[](pace_maker.Wait_for_Vevent imply pace_maker.c_avi <= AVI) in Uppaal
C10
The property for PVARP is A[] (pace_maker.Wait_for_Aevent imply pace_maker.c_lri >= PVARP) in Uppaal
C11
The property for PVAB is A[] (pace_maker.In_PAVB_VSP imply pace_maker.c_avi <= PAVB) in Uppaal
C12
The property for URL is A[] (pace_maker.VPACE_COMMIT imply pace_maker.c_lri >= URI) in Uppaal
C13
The property for URL is A[] (pace_maker.VSP_EX imply pace_maker.c_avi <= VSP) in Uppaal
Ev1
Model checking result of deadlock freeness
Ev2
Model checking result LRI property satisfied
Ev3
Model checking result VRP property satisfied
Ev4
Model checking result AVI property satisfied
Ev5
Model checking result PVARP property satisfied
Ev6
Model checking result PVAB property satisfied
Ev7
Model checking result URI property satisfied
Ev8
Model checking result VSP property satisfied


G4
The code generation process transforms the model into the code correctly
C14
Our Uppaal model is an extended finite state machine with states In_PVARP, Wait_for_Aevent, In_PAVB_VSP, In_VSP, Wait_for_Vevent, VSP_EX, AVI_BRANCH, AVI_EX, In_VRP_PVARP; 4 signals Asenses, Apace, Vsense and Vpace.
S4
Argument by use nested switch approach to translate extended finite state machine
S5
Argument by harmless modification after EFSM translation
C15
We manually write down the nested switch statement according to the Uppaal model
G14-22
For each state, the code have the same behavior branches as Uppaal model.
G23
The modification do not harm the correctness of the code generation
Ev9-17
The transition corresponding relation of model and code for all 9 states.
S6
Argument by keep the transition corresponding relation of and code and model
E18
The modification only divide the transition into 3 parts Asense interrupt, Vsense interrupt and non-interrupt without changing any transition.


G5
The manually written code satisfies all the desired properties
S7
Argument overall desired properties
G24
The model satisfies deadlock-freeness property
G25
The model satisfies property VRP
Ev19
Proof of the property preserved by our manually code generation
S8
Argument by testing
G26
Software test result for the LRI, AVI, VRP, PVARP, PVAB, VSP, URI showed no violation
Ev20
Test result for URI and PVARP in the instrumented code
Ev21
Test result for AVI and LRI in the instrumented code
Ev22
Test result for VRP and PAVB in the instrumented code
Ev23
Test result for VSP in the instrumented code