Glossary

Glossary of Terms

Assertion — An prescription on unpredictable behaviour, usually of the users in the environment

Behaviour — The sequence of inputs and outputs at a component or system over time.

Behaviour Axiom — Defines the behaviour of a component or system.

Compatible — Not mutually exclusive

Compatible Properties — Two properties that are not mutually exclusive, that is, that can be simultaneously satisfied by the same behaviour of a component or system.

Compatible Property Domain — The set of possible events that are common to two properties that are compatible.

Compatible Property Specification — The conjunction of the property specifications of two individual property specifications

Completeness (of Logic) — A logic is complete if it defines all possible truths

Completeness (of Set of Properties) — The set of properties is complete if it contains all properties relevant to the desired goals (safety and progress) of a system.

Composite Structures — The different interconnections between components.

- Product Structure — There are no interconnections between the components
- Cascade Structures — The outputs of one component are sent to another component as inputs. The second component does not send (directly or indirectly) an output to the first component.
- Feedback Structure — The outputs of one component are sent to a second component as an input. The second component will send outputs to the first components as inputs.

Composite System — The interconnection of two or more individual systems or components

Component — A functional element that may be interconnected to form a composite system or may exist as an individual, unconnected element.

Composite Environment — The environment of a composite system

Composition — The process of interconnecting components and systems to create larger, more complex systems.

Component Behaviour — The sequence of inputs and outputs occurring at a component.

Component Behaviour Axiom — A rule defining how a component will respond to a given input. This response may be dependent upon the state of the component.

Correctness — Implies that a system or program exactly corresponds to the desired specifications for that system or program.

Dependable System — One that will behave in predictable manner, even in the presence of faults.

Domain-Dependent Property — A specification of desired behaviour that depends on assumptions or prescriptions about the behaviour of the unpredictable environment.

Domain-Independent Property — A specification of desired behaviour that depends only on the predictable behaviour of components in a system.

Emergent Behaviour — A behaviour that is possible for an interconnection of components that is not possible or relevant to the individual components.

Emergent Property — A property that defines the desired behaviour of the emergent behaviours of a system.

Emergent Type 1 Property	A property that is defined and relevant to some, but not all, components in a system and is relevant to the interconnection of these components.
Emergent Type 2 Property	A property that is not relevant to any component in a system but is relevant to the interconnection of these components.
Environment	The source of inputs and destination of outputs to and from a component.
Event	A model-relevant action.
Acceptable/Unacceptable	If an event will never cause a bad thing to happen it is acceptable, otherwise it is unacceptable
Harmless	A harmless event is an unacceptable event whose harmful effect has been nullified.
Input/Output Event	An event that either received by a component (input) or produced by a component (output).
Relevant /Irrelevant	A relevant event may act as a stimulus. An irrelevant event will have no direct or indirect effect.
Feedback	The output of a component that is sent to another component (from which the component receives inputs) as an input.
Filtration	The process of removing unwanted events.
Input-Output Behaviour	The predictable relationship between input and output events at a component or system.
Interconnection	A channel between components in a system.
Interconnection Rule	The definition of possible interconnections within a system.
Liveness	The requirement that "something good" will eventually happen.
Meta-Environment	The source of all inputs and destination of all outputs, including any interconnected (predictable) components and the original (unpredictable) environment of a component.
Pedantic Property	A property that is not previously known that becomes relevant and is identified for a composite system.
Process	The execution by a component of an event.
Progress Property	A property, containing the temporal operator, ™, "eventually" that defines the eventuality of an action.
Property	The formal instantiation of a policy, representing the goal or desired behaviour of a component or system.
Reflexivity (of a property)	The satisfiability of a property for a component when the component has one of its outputs fedback to itself as an input.
Risk	A measure of the possibility that an event will occur and the worth, threat and vulnerability associated with the event.
Safety	The requirement that nothing bad ever happen within a system.
Safety Critical System	A system whose failure has catastrophic consequences, for example, the controller of a nuclear power plant.
Safety Property	A property that specifies those aspects of a component's or system[s behaviour that must always be true.
Simple System	A system that is represented as a stand-alone component, without reference to the internal connections or behaviours of the system.
Secure System	A system that satisfies a given set of safety and progress properties.
Sequence	An ordered set of input and output events occurring at a component or system.
Specification	The description of a component, system, or behaviour.
Stand-Alone Component	A component that has interconnections to and from the environment and no connections to and from other components.
State	The internal variables of a component or system.
System	A stand-alone component or an interconnection of such components.
System Behaviour	The sequence of inputs and outputs occurring at a system.

Threat	The potential damage to a system caused by the occurrence of an event.
Transitivity (of a property)	The satisfiability of a property for the composition of two systems given the satisfiability of the same property for the composition of each of the systems with another system
Trust (in a user)	The belief that a user will behave in a defined or predictable manner and will not intentionally cause harmful events to be received by a system.
Verification	The process of proving that the implementation of a system or property corresponds to the specification of that system or property.
Vulnerability (of a system)	The susceptibility of a system to certain, potentially harmful events.
Required World	Those events that are allowed to occur at a system. Events that are excluded from the required world are those that are known to be harmful yet may be received by the system.

Glossary of Notation

Temporal Logic:

$\circ$	Always in the future, a unary temporal connective
$\diamond$	Eventually in the future, a unary temporal connective
$\bigcirc$	Next time in the future, a unary temporal connectives.
$\mathcal{U}$	Until, a binary temporal connective.
$\exists$	Existence, a existential quantifier.
$\forall$	Universality, a universal quantifier.

General Notation:

$\mathbf{N}$	The components in a system, where there are $\mathbf{N}$ components, numbered from 0 to **n**-1
$\mathbf{E}$	The environment of a system
Φ	The set of events defined for a system
σ	A sequence of events
p	An atomic formula
A	A formula

$\mathcal{F}ma(\Phi)$ The set of formulae derived from the set of events Φ

$(\alpha ; \beta \vee \chi)$	Component behaviour rule, input α causes outputs β or χ.

$(i \rightarrow j)$ Interconnection rule, output of **i** sent to component **j** as an input

A\B	Set exclusion: those members of set A that are not members of set B.
$\mathbb{C}$	Composite system
$\mathbb{C}(C_i)$	Component C_i within composite system $\mathbb{C}$

$\cup_{i=1}^{\mathbb{C}} \mathbf{P}(C_i)$ Set of all properties for components in $\mathbb{C}$

$\cap_{i=1}^{\mathbb{C}} \mathbf{P}(C_i)$ Set of properties common to components in $\mathbb{C}$

$\mathbf{P}(\mathbb{C}) \cup= P_k$ Set of properties $\mathbf{P}(\mathbb{C})$ union property P_k

$\mathbf{G}(\mathbb{C})$	Goals of behaviour of composite system $\mathbb{C}$ represented as a set of properties

Events:

$\Phi(in)$	The set of input events defined for a system
$\Phi(out)$	The set of output events defined for a system
$\Phi(A)$	The set of acceptable events
$\Phi(U)$	The set of unacceptable events
$\Phi(i)$ or Φ^i	The set of events relevant to component **i**
$\Phi(i,j)$ or Φ^{i+j}	The set of events relevant to components **i** and **j**
$\Phi(i,(char))$	The set of events with characteristic char relevant to component **i**

Sequences:

$\sigma(i; j)$	A sequence of events from component **i** to component **j**
$\sigma(in)$	A sequence of events consisting only of input events
$\sigma(out)$	A sequence of events consisting only of output events
$\sigma[k]$	A sequence of events occurring on the channel k, where the first ℓ channels are input channels at a component and the next m-ℓ are output channels.
$\sigma\{\mathbf{i}\}$	The sequence of states of component **i**
$\sigma\{\mathbf{i}\}(in)$	The sequence of input events received at component **i**
$\sigma\{\mathbf{i}\}(out)$	The sequence of output events produced by component **i**

Several common sequences are defined as:

$$\sigma(i; j)(in) = \langle \langle in_0^j \rangle, \langle in_1^j \rangle, \ldots \rangle$$

$$\sigma(i; j)(out) = \langle \langle out_0^i \rangle, \langle out_1^i \rangle, \ldots \rangle$$

$$\sigma(i; j)[k] = \begin{cases} \langle \langle in_0^i[k] \rangle, \langle in_1^i[k] \rangle, \ldots \rangle & \text{iff } k < \ell \\ \langle \langle out_0^i[k] \rangle, \langle out_1^i[k] \rangle, \ldots \rangle & \text{iff } k \geq \ell \end{cases}$$

$$\sigma(i; j)(in)[k] = \langle \langle in_0^i[k] \rangle, \langle in_1^i[k] \rangle, \ldots \rangle$$

$$\sigma(i; j)(out)[k] = \langle \langle out_0^i[k] \rangle, \langle out_1^i[k] \rangle, \ldots \rangle$$

$$\sigma\{\mathbf{i}\}(in)[k] = \begin{cases} \langle \langle in_0^i[k] \rangle, \langle in_1^i[k] \rangle, \ldots \rangle & \text{iff } k < \ell \\ \langle \langle null \rangle, \langle null \rangle, \ldots \rangle & \text{otherwise} \end{cases}$$

$$\sigma\{\mathbf{i}\}(out)[k] = \begin{cases} \langle \langle out_0^i[k] \rangle, \langle out_1^i[k] \rangle, \ldots \rangle & \text{iff } k \geq \ell \\ \langle \langle null \rangle, \langle null \rangle, \ldots \rangle & \text{otherwise} \end{cases}$$

$$\sigma(in)[k] = \langle \langle in_0[k] \rangle, \langle in_1[k] \rangle, \ldots \rangle \quad \text{iff } k < \ell$$

$$\sigma(out)[k] = \langle \langle out_0[k] \rangle, \langle out_1[k] \rangle, \ldots \rangle \quad \text{iff } k \geq \ell$$

The notations $\sigma(i;j)(in)$ and $\sigma(i;j)(out)$ are used indicate the relevant aspect of the sequence (the destination or source) to be considered.

$\sigma \models_j A$	The sequence σ entails the formula A at time j.
$\sigma(\mathbb{S};\mathbb{D}) \models_j p$	The sequence from source $\mathbb{S}$ to destination $\mathbb{D}$ entails the event p at time j, where A is a formulae defined as A $\in \mathcal{F}ma\{\Phi(\mathbb{S}(out),\mathbb{D}(in))\}$
$\sigma(i+j;\mathbb{E})$	The sequence of events from components **i** and **j** to the environment

Composite Systems:

$\&(C_i, C_j)$	Composition of components C_i and C_j
$\&^P(C_i, C_j)$	Product composition of components C_i and C_j
$\&^C(C_i, C_j)$	(General) Cascade composition of components C_i and C_j
$\&^{\sim C}(C_i, C_j)$	Non-general cascade composition
$\&^F(C_i, C_j)$	(General) Feedback composition of components C_i and C_j
$\&^{\sim F}(C_i, C_j)$	Non-general feedback composition
C_i^R	The reflexive composition of component C_i

Properties:

P($\&(C_i, C_j)$) The set of properties of the composite system $\&(C_i, C_j)$

Q_k	The property domain of a property P_k, $\forall x : \mathbb{N} \cdot \lambda i : \mathbb{N} \cdot \lambda A : \mathcal{F}ma\{\Phi\}$
S_k	The property specification of a property P_k, $\lambda A \in \mathcal{F}ma\{W_k\} \wedge R_k$
W_k	The required world of property P_k, $W_k \subseteq \Phi$
R_k	The specification of the behaviour
sat(C_i, P_k)	True if the behaviour of component C_i satisfies the requirements of property P_k, where $C_i \times P_k \rightarrow \{T,\perp\}$

$Q_i * Q_j$ Compatible property domain, defined as $(\forall x : \mathbb{N} \cdot \lambda_{i1}\ i : \mathbb{N}_i \cap \mathbb{N}_j \cdot \lambda_{j1} j : \mathbb{N}_i \cap \mathbb{N}_j \cdot \lambda_{i2}\ A, \lambda_{i2}\ B : \mathcal{F}ma\{\Phi_i \cap \Phi_j\})$

$S_i \wedge S_j$ Compatible property specification, defined as $(\alpha_i \in \mathcal{F}ma\{W_i\} \wedge \alpha_j \in \mathcal{F}ma\{W_j\}) \wedge (R_i \wedge R_j)$

W Required world, given as $W = (\alpha_i, \alpha_j \in \mathcal{F}ma\{W_i \cap W_j\})$

composable(P_i) Proposition defining composability of property P_i

relevant(C_i,P_i) Proposition defining relevance of property P_i to the behaviour of component C_i

Appendix A: Proof System

If a formula q is *state-valid*, that is valid in all states, then the axiom of *tautology* defines

TAU $\models_{x} q$

That is, if a formula is valid in all states, it is valid in any one state, and can be written as a tautology.

From the tautology axiom we get the *generalisation* rule, which states that for a state formula p, if p is valid at a point in the sequence then it is valid over the entire sequence. For a state formula p,

GEN $\dfrac{\models_{x} p}{\models \circ p}$

The rule of *specification* is symmetric to generalisation. Specification states that a given state formula p that is valid for the sequence is valid at any point in the sequence. For a state formula p,

SPEC $\dfrac{\models \circ q}{\models_{x} q}$

The rule of *instantiation* allows us to infer formulae from a given general formula schema by substituting formulae within the schema. For a formula schema p and a replacement α

INST $\dfrac{p}{p[\alpha]}$

The most frequently used rule is known as *modus ponens*, used to infer the validity of a formula q given the validity p and $p \Rightarrow q$. *Modus ponens* is defined as:

MP $\dfrac{(p, p \Rightarrow q)}{q}$

These axioms are defined for state formulae instances of p and q, the propositional fragment of the language. We must therefore expand our deductive system to allow for quantification, using the $\forall$-generalisation rule. For a variable u and formulae p and q(u) such that u has no free occurrences in p

$\forall$-GEN $\dfrac{p \Rightarrow q(\mathrm{u})}{p \Rightarrow \forall \mathrm{u} \cdot q(\mathrm{u})}$

The use of both temporal and quantified formulae allows for instances of the Barcan formulae,

$\forall w \cdot \circ \mathrm{B} \Rightarrow \circ \forall w \cdot \mathrm{B}$

"The Barcan formula is true when each possible world has the same domain of individuals associated with it, i.e. when the range of the quantifier $\forall w$ is independent of the world (state) in which truth is being evaluated." [Gold92]. This property is satisfied by the models given in this work.