Based on the given predicates for the safety case pattern and the domain information for the GPCA system, here's the instantiated hierarchical tree format of the safety case using dashes (-) to denote different levels:

```

G1: Operational safety is verified in GPCA system

C1: All specs are given by spec.allInstances()

S1: Argument over the satisfaction of all specs over GPCA system

--- (Undeveloped)

G2: All operational hazards are mitigated

S2: Argument over reliability in all suitable levels of GPCA system

--- (Undeveloped)

S3: Argument over operational hazards

C2: Operational hazards are given by operational hazard.allInstances()

G3.1: Overinfusion is mitigated

S4.1: Argument over the applied scenarios of Overinfusion

C3.1: All related scenarios are given by Overinfusion

G4.1: Overinfusion is mitigated under "Flow rate does not match the programmed rate"

S5.1: Argument over all specs related to "Flow rate does not match the programmed rate"

C4.1: All related specs are given by scenario.spec.allinstances()

G5.1: Spec is appropriate for "Flow rate does not match the programmed rate"

S6.1: Argument the appropriateness of spec over properties

C5.1: Properties are given by spec.property.allinstances()

G6.1.1: Flow rate sensor is equipped is appropriate for Flow rate sensor spec

S7.1: Argument over the source of the Flow rate sensor is equipped definition

C6.1: Source is given by property.source

G7.1: FDA standard is appropriate and trustworthy

Sn1.1: FDA standard

G8.1: Flow rate sensor is equipped definition is sufficient

Sn2.1: Flow rate sensor is equipped property

G6.1.2: Period is 15 minutes is appropriate for underinfusion alarm period spec

S7.2: Argument over the source of the Period 15 minutes definition

C6.2: Source is given by property.source

G7.2: Expertise opinion is appropriate and trustworthy

Sn1.2: Expertise opinion

G8.2: Period is 15 minutes definition is sufficient

Sn2.2: Period is 15 minutes property

G6.1.3: Flow rate is less than 90% of programmed rate setting is appropriate for Flow rate spec

S7.3: Argument over the source of the Flow rate definition

C6.3: Source is given by property.source

G7.3: Previous knowledge is appropriate and trustworthy

Sn1.3: Previous knowledge

G8.3: Flow rate is less than 90% of programmed rate setting definition is sufficient

Sn2.3: Flow rate is less than 90% of programmed rate property

G3.2: Underinfusion is mitigated

S4.2: Argument over the applied scenarios of Underinfusion

C3.2: All related scenarios are given by Underinfusion

G4.2: Underinfusion is mitigated under "Programmed rate too low"

S5.2: Argument over all specs related to "Programmed rate too low"

C4.2: All related specs are given by scenario.spec.allinstances()

G5.2: Spec is appropriate for "Programmed rate too low"

S6.2: Argument the appropriateness of spec over properties

C5.2: Properties are given by spec.property.allinstances()

(Note: Only the instantiation for 'Overinfusion' and a part of 'Underinfusion' is shown. Instantiating all the placeholders for all operation hazards, scenarios, and properties follows similar steps as demonstrated above.)

```

Feel free to reduce or expand on the "Underinfusion" and other operational hazards based on the patterns and domain information accordingly. The hierarchical structure provided demonstrates how the pattern is instantiated with specific information about properties, sources, and mitigation strategies relevant to the GPCA system.