How WEST-Validated Formulas Align with R2U2 Runtime Monitoring Specifications

1. Temporal Logic Framework Compatibility

Both WEST and R2U2 support Mission-Time Linear Temporal Logic (MLTL), which is specifically tailored for mission-critical systems. WEST validates MLTL formulas by converting them into regular expressions, ensuring they adhere to the temporal requirements before deployment. These validated formulas can then be directly utilized by R2U2, which interprets MLTL for runtime monitoring. This compatibility allows for seamless sharing of specifications between the tools.

2. Lifecycle Integration

- **Pre-Deployment Phase (WEST):** WEST focuses on formula validation, ensuring the requirements are logically sound and free of contradictions. It outputs validated regular expressions or temporal logic formulas that define expected system behaviors.
- **Deployment Phase (R2U2):** R2U2 leverages these validated formulas as monitors for runtime anomaly detection, compliance checks, and predictive diagnostics.

By ensuring the formulas used in pre-deployment are directly applicable during deployment, the integration supports a consistent verification process, reducing the risk of misinterpretation or mismatched requirements.

3. Transformation of Outputs

One technical challenge in aligning the outputs of WEST with the inputs of R2U2 is ensuring the regular expressions or logical formulas are formatted in a way that R2U2 can interpret. This requires:

- **Syntax Translation:** Developing a parser or middleware that converts WEST's outputs into R2U2-compatible specifications.
- **Validation of Monitors:** Confirming that the monitors generated from WEST's outputs align with R2U2’s runtime capabilities, such as handling probabilistic reasoning and real-time processing.

4. Advantages of Alignment

- **Traceability:** Maintains a clear link between the original requirements (formalized and validated by WEST) and runtime monitoring (executed by R2U2).
- **Efficiency:** Reduces the need for manual re-translation of requirements into runtime specifications.
- **Reliability:** Ensures consistent application of requirements throughout the software lifecycle, enhancing overall system safety.

5. Example Application
In an aerospace system, a requirement such as "the system must complete a task within 10 seconds after a signal is received" can be formalized in MLTL, validated by WEST, and monitored during operation by R2U2. Any deviations detected by R2U2 could then be linked back to the requirements validated by WEST, ensuring quick diagnosis and correction.

Challenges and Recommendations

Challenges:

1. Differences in how tools interpret temporal logic.
2. Need for a robust and standardized translation mechanism.

Recommendations:

1. Develop a middleware tool to automate the conversion of WEST outputs into R2U2 monitors.
2. Conduct iterative testing to ensure alignment of outputs and inputs between the two tools.