such as a chips bag is broken on the conveyor belt, an event can be generated and the software architecture can take appropriate actions, such as stopping the packaging process and alerting the operator. Overall, event-driven architecture can provide a flexible and responsive software architecture for the packaging of bags of chips, allowing for efficient and reliable packaging operations.

Note: the role of hardware is crucial when designing robotic systems, and it significantly influences the performance and capabilities of the system. The same architecture using different hardware may have different results.

VII. Limitations and Future Perspectives

A. Limitations of Current Agricultural Robotic Architectures

Software architecture still remains an emerging discipline within the agricultural application domain. It has the following limitations.

1) *Lack of implementation works:* The majority of work efforts are still in the conceptual stage because the application of software patterns in agricultural robots is still in its early stages and is underresearched. Many proposed software architecture patterns are associated with theoretical analyses and design philosophy, but there is no proof of concept. Furthermore, applying those architectural solutions to commercial robots appears to be a long way off. Even the tools and standardized ways to represent architectural patterns are lacking.
2) *Unsatisfactory performance:* The performance of architectural patterns falls short of what is expected. Many pattern proposals claim to have powerful capabilities that aim to solve a variety of well-known problems, such as context awareness and interoperability.
3) *Low context awareness:* The architecture patterns for agricultural robots have a lack of context awareness. Given the ambiguity and ever-changing nature of agricultural environments, it is necessary to be aware of the surroundings in order to support the design of architectural patterns that can make appropriate adjustments. It will take some effort to incorporate the concept of context awareness into existing architectural patterns.
4) *Lack of methods to analyze:* There is a lack of suitable methods for analyzing and predicting whether a given architecture will result in an implementation that meets the needs of agricultural robot applications.
5) *Poor understanding:* A lack of understanding of the role of software architects has resulted in a communication breakdown among stakeholders in the agriculture industry.
6) *Security:* Security is always a hot topic that draws a lot of attention and necessitates extra effort. Security, on the other hand, is a feature that is missing from the majority of current architectural patterns. Considering the agricultural robotics, the issue of security can be divided into several aspects as follows. a) Data security. b) User access restrictions, and so on.
7) *Lack of design understanding:* There is a lack of knowledge about the design process, design experience, and design evaluation. Experts who are well-versed in both software design and agricultural applications are hard to come by.
8) *Lack of suitable environment:* It is necessary to make the agricultural environment robot-friendly. That is, any robotic architecture must be more structured in order to be successfully implemented.

What happens if we do not use any architecture patterns when creating a system? Every class will eventually connect to all other classes, resulting in a "big ball of mud."

B. Potential Future Directions

The field of agricultural robotics is constantly evolving and new advancements are being made every day. In order to keep up with these advancements, the software architecture of agricultural robots must also be updated and improved. Some potential future directions for software architecture in agricultural robots are as follows.

1) *Integration of artificial intelligence (AI) and machine learning (ML) techniques:* The integration of AI and ML techniques will help the agricultural robots to learn and adapt to new situations more quickly and efficiently. This will improve the overall performance and accuracy of the robots.
2) *Cloud-based computing:* The use of cloud-based computing can help to reduce the processing load on the robots, and make it easier to share data and collaborate with other robots and systems. This can also enable real-time monitoring and control of the robots from a remote location.
3) *Edge computing:* Edge computing involves processing data on the device itself, rather than sending it to the cloud for processing. This can be useful for agricultural robots that need to make real-time decisions in the field, as it reduces the latency associated with cloud-based systems.
4) *Internet of Things (IoT):* IoT technologies are becoming increasingly common in agriculture, as they allow farmers to collect and analyze data from a wide range of sources, including sensors, drones, and robots. These data can be used to optimize farming practices, improve crop yields, and reduce waste.
5) *Increased modularity and flexibility:* Modular and flexible software architectures can allow for easier customization and adaptation of the robots to different tasks and environments. This can improve the overall efficiency and effectiveness of the robots.
6) *Interoperability:* Interoperability between different types of agricultural robots and systems can facilitate collaboration and information sharing, leading to improved performance and productivity.
7) *Security:* The security of agricultural robots and their software architecture is becoming increasingly important, as these robots may contain sensitive information and be vulnerable to cyberattacks. Future directions may include the development of more secure and robust software architectures to protect against these threats.