

기술

AI에게 설비를 맡기려면, 틀린 명령이 나가기 전에 멈출 수 있어야 합니다

방동걸 (주)웨이스 대표이사

2026년 9월 18일

공장에 AI를 넣는 이야기는 대개 「무엇을 더 잘 판단하는가」로 흘러갑니다. 정확도가 몇 퍼센트인지, 어떤 모델을 쓰는지가 먼저 오갑니다. 그런데 제조 AI 운영체제 VEXPLOR를 만드는 웨이스가 설비 제어까지 이야기가 닿은 자리에서 먼저 듣는 질문은 다릅니다. 「그 판단이 틀리면 무엇이 막습니까.」 정확도를 아무리 올려도 이 질문에는 답이 되지 않습니다. 99%가 맞다는 말은 100번에 한 번은 틀린 명령이 설비로 나간다는 말이기도 하기 때문입니다.

저희가 구축한 현장에서는 서로 다른 제조사의 설비를 기존 제어를 교체하지 않고 같은 시각 기준으로 맞추고, AI의 판단이 설비 제어까지 연결되는 것을 실물에서 확인했습니다. 연결이 되고 나면 질문이 바뀝니다. 연결할 수 있느냐가 아니라, 연결된 자리로 무엇을 내보낼 것인가입니다. 그래서 웨이스가 대외에 내놓는 첫 문장은 성능이 아니라 정지입니다. 「VEXPLOR는 틀린 지시가 설비로 나가기 전에 멈춥니다.」 이 글은 그 한 문장이 실제로 무엇을 요구하는지 적습니다. 왜 설비에서는 되돌리기가 성립하지 않는지, 사후에 알아차리는 것과 보내기 전에 막는 것이 어떻게 다른 일인지, 명령 하나가 안전한지 보려면 무엇이 필요한지, 그리고 저희가 여기까지 만들었고 어디부터 아직인지까지입니다.

01

소프트웨어는 되돌릴 수 있고, 설비는 되돌릴 수 없습니다

IT 쪽에서 자동화를 이야기할 때 마음이 놓이는 이유가 있습니다. 틀리면 되돌리면 되기 때문입니다. 배포가 잘못되면 이전 판으로 내리고, 데이터가 잘못 들어가면 지우고 다시 넣습니다. 틀릴 수 있다는 것을 전제로 체계가 짜여 있습니다.

설비는 그 전제가 성립하지 않습니다.

구분	소프트웨어	설비
틀린 뒤	이전 판으로 내림	이미 움직인 뒤
피해	데이터·화면	제품·설비·사람
확인까지	로그로 즉시	다음 검사까지

실린더가 한 번 내려가면 그 동작은 취소되지 않습니다. 가열로 설정 온도를 잘못 올리면 그 안에 들어 있던 제품은 이미 그 열을 받습니다. 사람이 안전문 안에 있는데 컨베이어가 움직이면 그다음은 되돌리기의 문제가 아닙니다. 틀린 판단의 대가가 시간이 지난 뒤가 아니라 그 자리에서 발생하고, 그래서 사후 대응이라는 선택지가 없습니다.

이것이 공장이 AI에게 제어를 잘 맡기지 않는 진짜 이유입니다. AI를 못 믿어서가 아니라, 틀렸을 때 회수할 방법이 없는 곳에 자동 판단을 넣는 것이 원래 어려운 일이기 때문입니다.

02

사후에 알아차리는 것과 보내기 전에 막는 것은 다른 일입니다

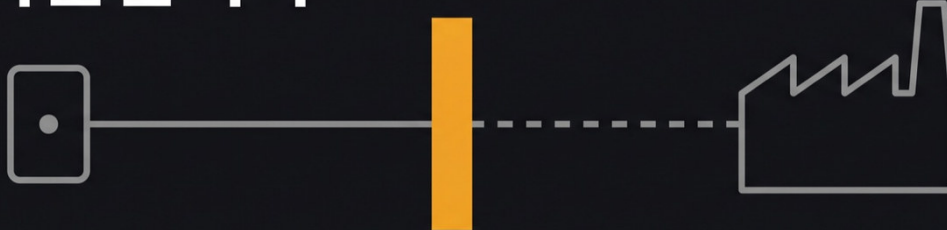
여기서 자주 섞이는 것이 있습니다. 「이상을 감지한다」와 「명령을 막는다」는 같은 말처럼 들리지만 시점이 다릅니다.

시점	무엇을 보는가	그 명령을 막는가
실행 뒤	설비가 낸 결과	못 막음
실행 중	지금 상태	이미 나간 뒤
실행 전	보낼 명령	막을 수 있음

현장에 이미 들어가 있는 감시 체계는 대부분 위의 두 줄입니다. 온도가 범위를 벗어나면 알람이 울리고, 진동이 커지면 보전 담당자에게 알림이 갑니다. 필요한 기능이고 잘 동작합니다. 다만 그 알람은 이미 일어난 일을 알려 줍니다. 사람이 운전할 때는 그것으로 충분했습니다. 사람은 버튼을 누르기 전에 스스로 한 번 더 생각하고, 이상하면 손을 멈추기 때문입니다. 그 몫이 AI로 넘어가면 그 자리가 비어 버립니다.

세 번째 줄이 비어 있다는 것이 문제의 전부입니다. AI가 내놓은 명령을 설비로 보내기 전에, 그 명령이 이 공장에서 무엇을 일으키는지 확인하는 단계가 지금은 없습니다.

막을 수 있는 시점은 하나뿐입니다



AI가 낸 명령이 설비에 닿기까지, 막을 수 있는 자리는 보내기 전 한 곳뿐입니다

03

명령 하나가 안전한지 보려면, 그 공장의 로직이 필요합니다

「위험한 명령을 막는다」를 규칙 목록으로 풀 수 있다고 생각하기 쉽습니다. 온도는 몇 도까지, 속도는 얼마까지처럼 값의 범위를 정해 두고 벗어나면 거르는 방식입니다. 이 방식은 절반만 동작합니다. 같은 값이 어떤 때는 안전하고 어떤 때는 위험하기 때문입니다.

컨베이어를 기동하라는 명령을 예로 들면 이렇습니다. 값만 보면 정상 범위입니다. 그런데 이 명령이 안전한지는 지금 안전문이 닫혀 있는지, 앞 공정에 제품이 걸려 있는지, 옆 설비가 지금 그 구간을 쓰고 있는지에 따라 갈립니다. 그 조건들은 명령 안에 없습니다. 그 공장 제어기의 프로그램 안에, 인터록이라는 형태로 들어 있습니다.

그래서 명령 하나를 판정하려면 그 공장의 프로그램이 필요합니다. 방법은 단순합니다.

- 그 공장의 제어 로직을 우리 편집기에서 다시 그린 검증용 프로그램을 검증하는 쪽에 둡니다.
- 그 프로그램의 상태를 현재 설비 상태에 맞춰 둡니다.
- AI가 보낸 명령을 실제 설비가 아니라 그 프로그램에 먼저 넣고 앞으로 몇 바퀴를 미리 실행해 봅니다.
- 그 사이에 지켜야 할 조건이 깨지면 그 명령을 설비로 보내지 않습니다.

저희는 이 단계를 「실행 전 가상 검증」이라고 부릅니다. 어려운 개념은 아닙니다. 사람이 숙련되면 머릿속으로

하는 일과 같습니다. 「지금 이걸 누르면 저게 움직이는데, 저 앞에 아직 제품이 있지」라고 한 번 그려 보는 것, 그 한 번을 기계가 대신하는 것입니다.

중요한 것은 **이 검증이 그 공장의 것이어야 한다는** 점입니다. 일반적인 안전 규칙집으로는 판정이 되지 않습니다. 인터록은 공장마다 다르고, 같은 공장에서도 설비를 바꾸면 달라집니다. 그래서 이 검증은 제품을 사 오면 바로 켜지는 것이 아니라, 그 공장의 프로그램과 함께 맞춰야 켜집니다.

04

현장은 AI의 정확도보다 「무엇이 멈추게 하는가」를 먼저 묻습니다

이 검증 단계를 실제로 공장에 들이려고 하면 질문의 종류가 바뀝니다. 저희가 현장 관점으로 설계를 검토하면서 정리한 것들은 대부분 모델 성능과 관련이 없었습니다.

- 이 장비가 공정 값을 어디에 남기는지 — 정보보안 담당자는 그 답이 없으면 반입을 승인하지 않습니다.
- 판정이 늦어지면 어떻게 되는지 — 상위 화면이 먼저 포기하고 명령을 다시 보내면, 그것은 같은 명령의 두 번째 실행입니다.
- 이 장치가 고장 나면 라인이 멈추는지 — 안전을 위해 전부 막도록 만들면, 그 장치의 고장이 곧 생산 정지가 됩니다.
- 한밤에 문제가 생기면 누가 오는지 — 첫 야간 장애에서 아무도 못 부르면 아침까지 라인이 멈춥니다.
- 막았다는 기록이 나중에 근거가 되는지 — 사고 조사에서 「우리 기록을 우리 도구로 재생해 우리가 맞다고 한다」로는 끝나지 않습니다.

다섯 줄 모두 운영의 질문입니다. AI를 공장에 넣는 일의 난이도는 판단의 정확도가 아니라 이 줄들에 답하는 데 있습니다. 그리고 이 답들은 제품을 다 만든 뒤에 붙일 수 없습니다. 어디에 무엇을 남길지, 판정이 늦으면 어느 쪽으로 기울지는 만들 때 정해지는 것이라 나중에 바꾸면 그 전에 나간 판정의 근거가 함께 흔들립니다.

05

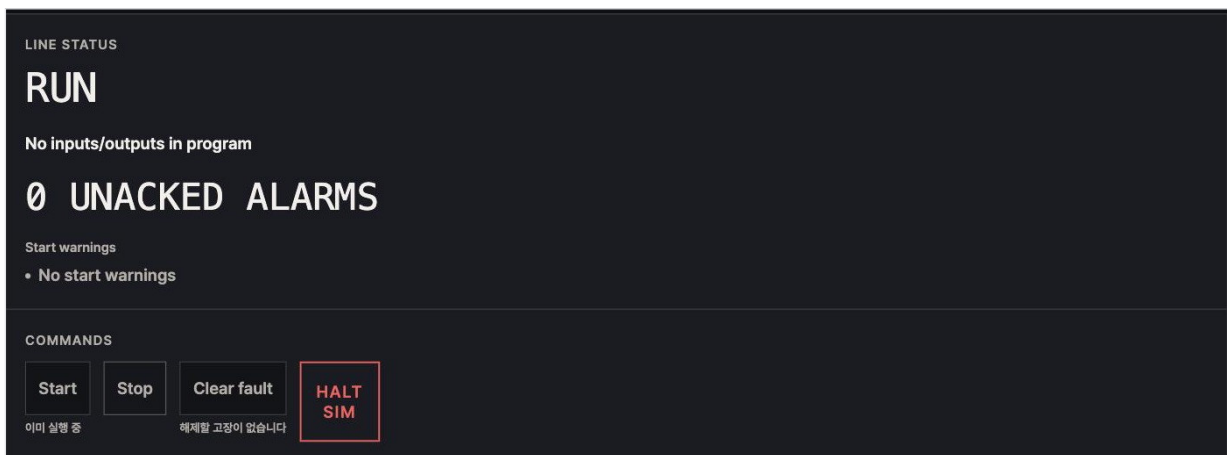
검증하는 쪽이 현장을 불편하게 만들면, 현장은 그것을 꺼 버립니다

여기에 한 가지가 더 붙습니다. 안전한 쪽으로 기울이면 기울일수록 현장에서 쓰기 어려워진다는 것입니다.

판정이 확실하지 않을 때 막는 쪽을 고르면 안전합니다. 다만 그 선택이 잦아지면 현장에서는 「이 장치 때문에 일이 안 된다」가 됩니다. 판정에 시간이 더 걸려 통신 알람이 늘어도 같은 결론에 닿습니다. 그리고 현장이 그렇게 판단하면 그 장치는 조용히 꺼집니다. 꺼진 검증은 없는 것보다 나쁩니다. 있다고 믿으면서 없기 때문입니다.

그래서 저희는 이 검증을 처음부터 자동 실행으로 켜지 않습니다. 순서가 있습니다. 먼저 AI의 판정을 숙련자의 판정과 같은 사건에 나란히 기록만 하는 기간을 둡니다. 저희 서비스 정의에서 「Shadow Mode」라고 부르는 단계로, 이때 AI는 설비에 개입하지 않습니다. 그다음에 AI가 판단을 올리고 **사람이 승인해야 실행되는 단계**이고, 자동으로 실행하는 단계는 그 뒤입니다. 앞 단계에서 쌓인 기록이 「이 조건에서는 맡겨도 된다」의 근거가 됩니다. 근거 없이 단계를 건너뛰면 현장은 그 판단을 받아들이지 않습니다.

이 검증을 만들 때 실제로 설계해야 하는 것도 「막는 기능」이 아니라 **막는 것과 흘려보내는 것 사이의 선**입니다. 얼마나 늦어도 되는지, 어떤 경우에 판정을 포기하고 보수적으로 갈지, 그때 운전원에게 무엇이 보이는지를 숫자와 화면으로 정해 두어야 합니다. 저희가 지금 시간을 가장 많이 쓰는 곳도 이 선입니다.



사내 시험 환경의 운전원 화면입니다. 버튼을 누를 수 없을 때 그 아래에 이유가 함께 나옵니다 — 「이미 실행 중」, 「해제할 고장이 없습니다」

06

지금 만들고 있는 것과 다음에 붙이는 것

저희가 이 검증에서 지금 세워 둔 것은 둘입니다. 제어 프로그램을 제어기 밖에서 같은 의미로 실행하는 부분과, 명령을 설비로 보내기 전에 붙잡아 두는 부분입니다. 사내 시험에서 이 둘이 함께 동작하는 것을 확인했습니다.

다음 순서도 정해져 있습니다.

- **판정 기록과 재계산** — 판정 입력·결과·사유를 기록으로 남기고, 그 기록만으로 같은 판정을 다시 만들어 대조합니다. 지금 붙이고 있습니다.
- **실기 제어기 대조** — 지금 값은 시뮬레이션 환경에서 잤 것이고, 다음은 실제 제어기와 붙여 같은 항목을 다시 잹니다. 숫자는 그 대조를 마친 뒤에 공개합니다.
- **운전 중 검증** — 실제 라인이 가동 중일 때 판정이 제 시간에 끝나는지, 운전원 화면에서 무엇이 보이는지를 현장에서 확인합니다.

- **고객과 함께 정하는 것** — 위에 적은 다섯 가지 운영 질문의 답은 공장마다 다릅니다. 첫 현장의 담당자와 함께 값으로 정하고, 그것을 다음 현장의 기본값으로 삼습니다.

공장이 스스로 판단하게 하려면 판단을 잘하게 만드는 것만으로는 부족합니다. 판단이 틀렸을 때 그 판단이 설비에 닿기 전에 멈추는 자리가 함께 있어야 합니다. 그 자리가 있어야 사람이 공장에 판단을 맡길 수 있습니다.

WACE(웨이스)는 공장이 스스로 판단하고 멈추지 않게 하는 제조 AI 운영체제 VEXPLOR을 만드는 회사다.



원문	blog.vexplor.com/post/stop-before-the-command-reaches-the-machine
발행	2026년 9월 18일 · 방동걸 (주)웨이스 대표이사
회사	주식회사 웨이스 (WACE Inc.) · 대표이사 방동걸 · 사업자등록번호 439-87-02852
주소	(18462) 경기 화성시 동탄구 동탄영천로 150, 에이동(A동) 1702·1706호 (영천동, 실리콘엘리동탄)
연락처	전화 070-4730-0629 · contact@wace.me
홈페이지	wace.me · linkedin.com/company/wace-inc



원문



홈페이지



링크드인